A Mathematical Game and Its Applications to the Design of Interconnection Networks

Chi-Hsiang Yeh
Dept. of Electrical & Computer Engineering
Queen's University
Kingston, Ontario, K7L 3N6, Canada
yeh@ee.queensu.ca

Emmanouel A. Varvarigos*
Dept. of Electrical & Computer Engineering
University of California
Santa Barbara, CA 93106-9560, USA
manos@ece.ucsb.edu

Abstract

In this paper, we propose a mathematical game, called the ball-arrangement game (BAG). A game with a different set of rules (e.g., permissible moves) gives rise to a different network, and the algorithm that solves the game gives rise to a routing algorithm in that network. Based on the insights provided by BAG, we propose several new classes of symmetric and modular networks, called super Cayley graphs, that have optimal (intercluster) diameters and average (intercluster) distances, small (intercluster) node degrees, high bisection bandwidth, strong embedding capability, and optimal communication algorithms given their (intercluster) node degrees.

1 Introduction

In this paper, we introduce a mathematical game called the *ball-arrangement game (BAG)* and we use an interesting analogy to design several new classes of interconnection networks and their algorithms. In the ball-arrangement game, we are given k balls, each stamped with a number. Different balls may be assigned the same or different numbers. The goal of the game is to rearrange the balls so that the numbers on the balls appear in a desired order. At each step the player can take an arbitrary action from a set of d permissible moves, each being a particular permutation of the balls. The set of permissible moves remains the same throughout the game, independent of the current configuration of the balls. If the k balls have different numbers, then there are $k!$ possible configurations of the balls (i.e., states) when playing the game. If we view each of the states as a network node and a permissible move leading from one state to another as a directed link connecting the nodes corresponding to those two states, then a network with $k!$ nodes results, where each node has d outgoing links. In other words, the network can be obtained by drawing the state transition graph for the corresponding ball-arrangement game with specified movements. One can then relate playing a ball-arrangement game to routing in the corresponding network, where the initial and final states correspond to the source and destination nodes and the sequence of movements performed to solve the game corresponds to the sequence of links along the routing path. Since the in-/out-degree of the derived network is upper bounded by the number d of permissible moves and the diameter is the maximum number of steps required to solve the game, we generally prefer to select a small number of permissible moves that allow us to solve the game in a small (or optimal) number of steps for any initial and final states.

A *k-dimensional star graph*, *k-star* [1, 2], is a well-known network that has a number of desirable properties, such as degree, diameter, and average distance smaller than those of a similar-size hypercube, symmetry properties, strong embedding capability, and fault tolerance properties. A variety of efficient algorithms have been proposed for star graphs and various properties have been investigated in the literature [3, 4, 5, 10, 13, 14, 20, 21, 23, 25, 27, 28, 35]. A k-star is derived from a special case of the ball-arrangement game where each ball has a distinct number and at each step the player can interchange the leftmost ball with an arbitrary ball [1, 2]. Akers, Harel, and Krishnamurthy presented a simple and efficient algorithm to solve the game in at most $\lfloor 3(k-1)/2 \rfloor$ steps [1, 2]. Therefore, the degree of an N-node k-star is $k-1 = \Theta(\log/\log\log N)$ and the diameter is $\lfloor 3(k-1)/2 \rfloor = \Theta(\log/\log\log N)$, both of which are sublogarithmic. Moreover, it can be shown that the diameter of a star graph is optimal within a factor of $1.5 + o(1)$ † from a universal lower bound given its node degree [32]. In [2], Akers and Krishnamurthy develop a group-theoretic model, called the *Cayley graph* model, for designing and analyzing symmetric interconnection networks. A Cayley graph can be defined by a corresponding permutation group, which corresponds to a ball-arrangement game where different balls have different numbers. In [2], Akers and Krishnamurthy showed that Cayley graphs are vertex-symmetric and that most vertex-symmetric graphs can be represented as Cayley graphs; it was also shown that every vertex-symmetric graph can be represented as a *Cayley coset graph*. In [31, 37], we derived an analogous result showing that every graph corresponds to a certain ball-arrangement game. Both the Cayley graph model and the Cayley coset graph model have been used to derived a wide variety of interesting networks for parallel processing and have since received considerable attention [2, 9, 11, 12, 15, 18, 26, 32]. Many networks can be formulated by simple ball-arrangement games and that algorithms for networks derived from a similar set of moves can usually be developed in a unified manner. We have also used the underlying idea of the ball-arrangement game to derive a variety of efficient networks that have certain desirable properties [31, 32, 33, 34, 36, 37].

Although the star graph has many desirable properties, its node degree is still too large when the network size is large. The reason is that the corresponding ball-arrangement game permits $k-1$ moves for rearranging the balls so that degree $\Theta(k) =$

*Work performed while with TU Delft, Netherlands.

†The notation $f(N) = o(g(N))$ means that $\lim_{N\to\infty} f(N)/g(N) = 0$.

0190-3918/01 $10.00 © 2001 IEEE

$\Theta(\log N/\log\log N)$ is required. To reduce the node degree, we have to find a ball-arrangement game with fewer permissible moves, which, collectively, can still sort the balls to any desired order in a small number of steps. Recently, we proposed the *Balls-to-Boxes* game, which is a special case of the ball-arrangement game, where $k = nl + 1$ balls with distinct numbers are placed in l boxes, each having n balls, except for a ball left outside these boxes [32]. At each step, the player in the Balls-to-Boxes game can interchange the leftmost box with an arbitrary box or interchange the outside ball with an arbitrary ball in the leftmost box. When $l = \Theta(n)$, the number of permissible moves is $O(l) = O(\sqrt{k}) = O(\sqrt{\log N/\log\log N})$ so the degree of the resultant network, called the *macro-star network*, is considerably smaller than that of a star graph of the same size. In contrast to interchanging the leftmost ball with an arbitrary ball in the game for star graphs, we in general have to move the box containing the ball to the leftmost position before the two balls can be interchanged. For networks of practical size in the near future (e.g., $N \leq 10! \approx 3.6 \cdot 10^6$), the degree of macro-star networks is only 5 or smaller, which is even smaller than the degree of some constant-degree networks, such as 3-D meshes/tori and pyramids. We have showed that the diameter of a macro-star network is also sublogarithmic and is optimal within a factor of $1.25 + o(1)$ from a universal lower bound, given its node degree. We have also showed that no networks with similar node degree can execute multinode broadcast (MNB) and total exchange (TE) tasks [7, 29, 30] in time that is much better than macro-star networks.

Based on the insight provided by the ball-arrangement game, we obtain a number of new networks that further improve the properties of macro-star networks. In particular, some of the derived networks have diameter smaller than that of a macro-star network of the same size by a factor of 1.25. The resultant networks form a subclass of (directed) Cayley graphs and will be referred to as *super Cayley graphs*. Super Cayley graphs are symmetric and modular, and generally have node degree that is considerably smaller than that of a similar-sized star graph. They come at various sizes and degrees, which are determined by parameters l (the number of boxes) and n (the number of balls in a box). In this paper, we propose nine classes of novel super Cayley graphs which have their respective advantages. The diameters of macro-star and complete-rotation-star networks with $l = \Theta(n)$ (to be referred to as *balanced* networks) are optimal within a factor of $1.25 + o(1)$ from a universal lower bound given their node degrees; the diameters of balanced macro-rotator, complete-rotation-rotator, macro-IS, and complete-rotation-IS networks are optimal within a factor of $1 + o(1)$. The average distances of the preceding six subclasses of super Cayley graphs proposed in this paper are all optimal within a factor of $1 + o(1)$ from a universal lower bound given their node degrees.

The intercluster degrees of super Cayley graphs are small, leading to high link bandwidth when implemented as *multiple chip-multiprocessors (MCMP)* [36], where a chip contains multiple processors on it and multiple such chips are interconnected together to build a parallel system. Also, the intercluster diameters and average intercluster distances of suitably constructed super Cayley graphs are asymptotically optimal. Moreover, the bisection bandwidths of many super Cayley graphs are higher than those of hypercubes and k-ary n-cubes. These characteristics indicate that super Cayley graphs can achieve high performance for communication-intensive tasks such as total exchange and random routing when they are implemented as MCMPs (see [36] for more details).

2 The ball-arrangement game (BAG) and associated algorithms

As described in the introduction there is an interesting relationship between network topologies and the ball-arrangement game and between algorithms that solve the game and algorithms that perform routing in the corresponding networks. In this section, we start by describing a particular instance of the game, called the *Balls-to-Boxes game*, and then generalize it to obtain the *ball-arrangement game (BAG)* with boxes and distinct balls. We also propose several rules and algorithms to solve the game.

2.1 The balls-to-boxes game and algorithm

Balls-to-Boxes Game [32]:

We are given l boxes, each of which is assigned a distinct color in $\{1,2,...,l\}$, and $k = nl + 1$ balls. $k-1$ of the balls are partitioned into l groups of size n, each of which is assigned a distinct color in $\{1,2,...,l\}$, while the remaining ball is assigned color 0 and does not belong to any group. Initially, $k-1$ of the balls are mixed together in the l boxes, so that each box contains n balls (of different colors, in general), and one ball is left outside the boxes. At each step the player can take one of the following actions:

(1) exchange the outside ball with one of the balls in the leftmost box, or

(2) exchange the leftmost box with any of the other boxes.

The goal of the game is to rearrange the balls and the boxes so that each ball ends up in a specific position in the box that has the same color, except for the ball with color 0 that ends up outside the boxes. Also, the boxes should be sorted so that the box of color i, $i \in \{1,2,...,l\}$, appears in the i^{th} position from the left.

Note that there are $N = (nl+1)!$ distinct configurations (i.e., placements of balls to boxes), and $n + l - 1$ possible moves from one configuration to another. At any time in the game, the ball that is currently outside all boxes will be called the *outside ball*, while the box currently at position 1 will be called the *leftmost box*. A ball that is currently in a box of color different than its own color, or a ball that is at a wrong position in a box of the same color (in a Balls-to-Boxes game) will be referred to as a *dirty ball*. A box that contains at least one dirty ball will be referred to as a *dirty box*. A ball or box that is not dirty will be called *clean*. It is easy to verify that the following algorithm solves the Balls-to-Boxes game.

Balls-to-Boxes Algorithm

- Phase 1
  - Case 1.1: If the outside ball has color 0:
    - 1.1.1: If all boxes are clean, go to Phase 2; If the leftmost box is clean, exchange it with a dirty box and go to Step 1.1.2.

* 1.1.2: Exchange the outside ball (which has color 0) with any dirty ball in the leftmost box and go to Step 1.2.1.

- Case 1.2: If the outside ball has color $c \neq 0$:
  * 1.2.1: If the color of the leftmost box is different than c, then swap the leftmost box with the box of color c and go to Step 1.2.2.
  * 1.2.2: Put the outside ball at its correct position in the leftmost box (which has color c), take the ball occupying that position outside, and go to Phase 1.

- Phase 2 Now all boxes are clean (they contain balls of the correct color, placed at their correct positions), but they may not be in the correct order. To put them in the correct order so that box of color i is placed at position i, the following algorithm is run:
  - 2.1: If all boxes appear in the correct order, then stop; otherwise, go to Step 2.2.
  - 2.2: If the leftmost box has color 1 then exchange it with any box that is not at its correct position and go to Step 2.3.
  - 2.3: Exchange the leftmost box with the box at the i^{th} position and go to Step 2.1.

Phase 1 of the algorithm requires at most $\lfloor 2.5nl \rfloor + l - 1$ steps, while Phase 2 requires $\lfloor 1.5(l-1) \rfloor$ steps. More details concerning the Balls-to-Boxes algorithm can be found in [32].

2.2 Description of the ball-arrangement game

Observe that Phase 1 of the Balls-to-Boxes algorithm results in an incorrect order for the boxes, so that additional movements are required in Phase 2 to rearrange the boxes, increasing the time required to complete the game. A question arises: *Is there any rule that can keep the order of boxes unchanged (relative to each other) when we play the game?* In that case Phase 2 can be eliminated or simplified.

One way to achieve that is to modify the Balls-to-Boxes algorithm so that a box is moved back to its original position after it is interchanged with the leftmost box (Phase 1.2.1 of the Balls-to-Boxes algorithm). Then, we can skip Phase 2 completely, assuming that the boxes are in correct order at the beginning of the algorithm. This modification, however, usually results in longer execution time since it leads to a reduction of only $O(l)$ steps while requiring about $k \approx nl$ additional steps.

Consider rotating the boxes to any position, instead of swapping them, in a single movement. In this way, the order of boxes is preserved (cyclicly), and Phase 2 (i.e., reordering the boxes) can be completed in at most one rotation step. Then we can show that the balls and boxes can be sorted in about $1.5l$ fewer steps in the worst case. Note that Phases 1.1.1 and 1.2.1 of the Balls-to-Boxes algorithm, which exchange the leftmost box and a box i, should now be replaced by "rotating the boxes until box i appears at the leftmost position."

Note that when playing this game, the colors of boxes have to be in ascending order cyclicly at any given time. Therefore, if the boxes appear in incorrect order at the beginning, this game cannot be solved by rotating the boxes along. Motivated by this, we propose the *ball-arrangement game (BAG)*, which permit more flexibility in possible movements than the Balls-to-Boxes game, in order to investigate other interesting ways to sort balls and boxes. A special case of the ball-arrangement game that involves l boxes, each having n balls, and a ball outside boxes, is described as follows.

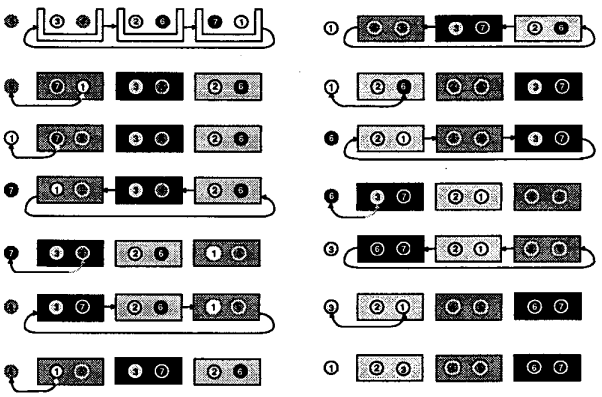

Figure 1: A ball-arrangement game where boxes are moved by rotations. The initial state (i.e., source node) is 1 and the final state (i.e., destination node) is $I = 1234567$. The boxes do not have colors at the beginning, and are assigned colors 2, 3, and 1 for performing routing. Note that ball 1 (i.e., symbol 1) appears at the leftmost position several times.

BAG with l Boxes and $nl+1$ Distinct Balls:

We are given l boxes and $k = nl + 1$ balls, one of which has color 0 and n of which has color i for all $i = 1,2, ..., l$. These boxes do **not** *have color (at the beginning). Initially, $k-1$ of the balls are mixed together in the l boxes, so that each box contains n balls (of different colors, in general), and one ball is left outside the boxes. At each step the player can take one of the following two types of actions:*

- *(1) rearrange the order of the leftmost $n+1$ balls (i.e., the outside ball and the balls in the leftmost box), or*
- *(2) rearrange the order of boxes.*

The goal of the game is to rearrange the balls and the boxes so that balls with the same color ends up in the same box, with proper order. Also, these boxes should be sorted so that the balls of color i, $i \in \{1,2, ..., l\}$, appears in the i^{th} box from the left.

Note that the boxes do not have colors at the beginning and we have the freedom in assigning colors to the boxes so as to facilitate the use of algorithms similar to the Balls-to-Boxes algorithm to solve a game. For example, we can assign box i (of the initial configuration) with color $(i + b - 1 \bmod l) + 1$ for some integer b, then the "rotation of boxes" can completely replace Phase 2 of the Balls-to-Boxes algorithm. Figure 1 illustrates an example with such permissible moves.

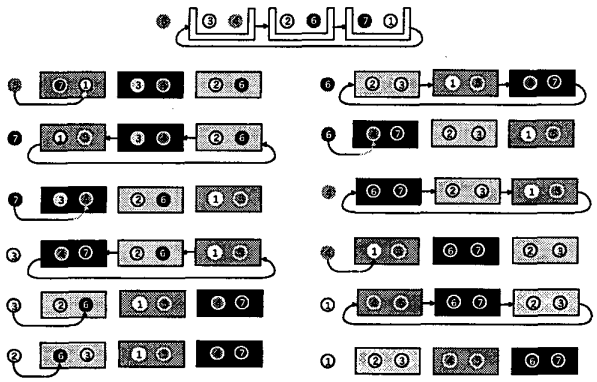

Figure 2: A ball-arrangement game where balls are moved by insertions. The initial state (i.e., source node) is 5342671 and the final state (i.e., destination node) is $I = 1234567$. The assignment of colors to boxes is the same as that in Fig. 1.

2.3 A BAG where balls are moved by insertion

After playing the Balls-to-Boxes game and the preceding ball-arrangement game with rotation several times, we find that the ball with color 0 frequently becomes the outside ball (Case 1.1) for some initial configurations of balls, in which case we have to waste our time in exchanging it with a dirty ball. This situation may happen up to about $k/2$ times in the worst case and cannot be eliminated by any algorithm. Figure 1 gives such an example for a ball-arrangement game where boxes are moved by rotation. In a Balls-to-Boxes game with the same initial and final configurations, the ball of color 0 will become the outside ball for the same number of times.

In what follows, we introduce a different set of permissible moves that can significantly reduce the occurrence of this undesirable situation: Instead of interchanging the outside ball with a ball in the leftmost box, we now let the player "insert" the outside ball to any position of the leftmost box, which also moves all the balls that are not on the righthand side of that position to the left by one position, and take the original leftmost ball in the box outside the box.

To derive an algorithm that solves this ball-arrangement game, let us first consider a special case where there is only one box ($l = 1$), in order to simplify the problem. At Step 1, we compare the outside ball with the rightmost a balls, where $a \in \{1,2,...,n\}$ is the largest number such that balls $n-a+1, n-a+2, ..., n$ are in ascending order. We then insert the current outside ball to an appropriate position (i.e., among the rightmost $a+1$ positions) so that the $a+1$ rightmost balls are in ascending order. Similarly, at Step t, $t = 2,...,k-a$, we compare the outside ball with the rightmost $t+a-1$ balls, and insert it in an appropriate position so that upon the completion of Step t, the rightmost $t+a$ balls are in ascending order. These rightmost $t+a$ balls that are in ascending order will be referred to as *clean* balls (for this particular game), and the remaining will be referred to *dirty* balls. Therefore, at most $k-1$ steps are required to solve this ball-arrangement game with one box,l since there is at least one clean ball at the beginning of the game ($a \geq 1$).

Let us now consider the general case of this ball-arrangement game, where there are l boxes. To solve the game, we have to repeatedly "insert" the outside ball into an appropriate position within the box that has the same color i. The first step, of course, is to "bring" the box with same color i to the leftmost position, by either swapping or rotating the boxes (Phase 1.2.1), or any other permissible actions of the second type. Let c_i, $i \in \{1,2,...,l\}$, be the number of rightmost balls that have color i and are placed in ascending order within the box of color i; that is, c_i is the number of *clean balls* in the box of color i. Then, Phase 1.2.2 of the Balls-to-Boxes game is replaced by a phase where we "insert" the outside ball into an appropriate position among the rightmost $c_i + 1$ positions in the box of color i.

When the outside ball has color 0, we still have to get it out of the way by placing it at the $(c_i+1)^{th}$ rightmost position in a dirty box of color i. This, however, can only happen at most l times (right after a box becomes clean), for a total of $2l-1$ steps. As a comparison, the previous rules [32] may happen up to about $k/2 \approx nl/2$ times, for a total of $k \approx nl$ steps. Thus, we have improved the execution time by about $k/2 - 2l$ steps in the worst case by using this set of permissible movements. Figure 2 illustrates a ball-arrangement game where balls are moved by insertions. Figure 3 illustrates another way to solve the game in Fig. 2, which uses a different assignment of colors and considerably reduces the number of steps required.

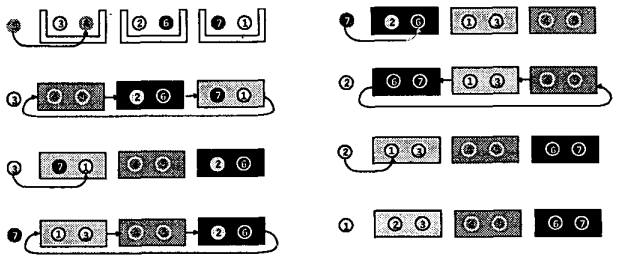

Figure 3: A ball-arrangement game where balls are moved by insertions. The initial state (i.e., source node) is 5342671 and the final state (i.e., destination node) is $I = 1234567$. The assignment of colors to boxes is different from those in Fig. 1 and Fig. 2, which leads to considerable reduction in the number of steps compared to that required in Fig. 2.

3 Super Cayley graphs: BAGs with l boxes and $nl+1$ distinct balls

In this section, we formally define several communication-efficient networks that correspond to games with different sets of permissible movements. To provide some intuition and better visualize the network topologies, we first relate the ball-arrangement game to network topologies and the algorithm that solves a game to the routing algorithm in the corresponding network.

For a ball-arrangement game with k balls, there are $k!$ distinct configurations (i.e., states), each of which can be visualized as a node of a network. Given a set of actions for moving the boxes and balls, we can visualize a movement between two configurations as a link connecting those two corresponding nodes. That is, the network can be obtained by drawing the state transition graph for the ball-arrangement game.

If d possible actions are allowed in a ball-arrangement game,

then each node in the derived network has d outgoing links connecting it to d other nodes in the network. Sending a packet from node $X^{(0)}$ to node $X^{(1)}$ through link i corresponds to moving the boxes or balls according to action i so that the configuration is changed from $X^{(0)}$ to $X^{(1)}$. Therefore, we can relate routing in the network to sorting boxes and balls in the corresponding game, where the source and destination nodes corresponds to the initial and final configurations, the routing path consists of the links corresponding to the actions taken to solve the game, and the diameter is the maximum number of steps required to solve the game for any initial and final configurations.

In what follows, we formally define the graphs that correspond to the games described in the previous section.

3.1 Macro-star networks

The MS(l,n) network corresponds to the Balls-to-Boxes game with l boxes, each having n balls, and an outside ball. A node in the MS(l,n) network can be represented as a permutation of $k = nl+1$ distinct symbols, where each symbol can be viewed as a ball in the Balls-to-Boxes game. More precisely, a permutation of k distinct symbols in the set $\{1,2,...,k\}$ is represented by $U = u_{1:k} = u_1u_2\cdots u_k$, where $u_i \in \{1,2,...,k\}$ and $u_i \neq u_j$ for $i \neq j$, $1 \leq i,j \leq k$. On the set of all possible permutations of k symbols, we introduce two types of operators (generators). The first type of generators corresponds to the actions that interchange the outside ball with a ball in the leftmost box; while the second type corresponds to the actions that interchange the leftmost box with another box in the Balls-to-Boxes game.

Definition 3.1 (Transposition T_i and Swap Generators T_i, $S_{i,n}$) : Given a permutation $U = u_{1:k}$, we define the *dimension-i transposition generator* T_i, $i = 2,3,...,k$, as the operator that interchanges symbol u_i with symbol u_1 in $u_{1:k}$. We also define the *level-i swap generator* $S_{i,n}$ as the operator that interchanges the sequence of symbols $u_{(i-1)n+2:in+1}$ with the sequence of symbols $u_{2:n+1}$ in $u_{1:k}$, where $2 \leq i \leq l$ and $k = nl+1$.

In what follows, we will use S_i instead of $S_{i,n}$, suppressing the dependence on n, unless explicitly stated otherwise. More details concerning the preceding generators and the definition, structure, example, and properties of macro-star networks can be found in [31, 32].

3.2 Super Cayley graphs

In this subsection we present the definition of super Cayley graphs, which are derived from the ball-arrangement game.

Each node of a super Cayley graph is represented as a permutation of k distinct symbols, where k is the number of balls in the ball-arrangement game it is derived from. We define the i^{th} *super-symbol* of node label U as the n-long sequence of symbols at positions $(i-1)n+2,(i-1)n+3,...,in+1$ in the permutation label of node U. On the set of all possible permutations of k symbols, we introduce two classes of operators:

- *nucleus generators,* which permute in some way the leftmost $n+1$ symbols (i.e., the leftmost symbol and the leftmost super-symbol, corresponding to the outside ball and the balls within the leftmost box) in the ball-arrangement game.
- *super generators,* which permute super-symbols without changing the contents of each of these super-symbols (i.e., corresponding to moving boxes in the ball-arrangement game).

For example, transposition generators T_i are nucleus generators and swap generators $S_{i,n}$ are super generators.

A *super Cayley graph* is a (directed) Cayley graph [2, 9, 12] defined by nucleus generators and super generators. For example, macro-star networks MS(l,n) are Cayley graphs defined by n transposition nucleus generators T_i, $i = 2,3,...,n+1$, and $l-1$ swap generators $S_{i,n}$, $i = 2,3,...,l$. A super Cayley graph that is defined with l super-symbols is called an l-level super Cayley graph.

According to the preceding definition, node U of a super Cayley graph is connected to node V by a directed link if and only if the permutation label of node V can be obtained from that of node U either by permuting the leftmost $n+1$ symbols of U using one of the nucleus generators in its definition, or by permuting super-symbols of U using one of the super generators in its definition. Links corresponding to the former are called *nucleus links*; while links corresponding to the latter are called *inter-cluster links*. Clearly, a super Cayley graph is a directed Cayley graph [2], whose in-/out-degree is equal to the number of generators in its definition. Since any directed Cayley graph is vertex-symmetric and regular [2, 9, 12], super Cayley graphs are vertex-symmetric and regular. Note that in some Cayley graphs, such as macro-star networks, each directed link has a corresponding directed link that has the same ending nodes and opposite direction. These graphs can be viewed as undirected Cayley graphs [2], by merging each pair of such directed links.

3.3 Definitions of several generators and super Cayley graphs

In what follows we formally define several other super Cayley graphs, which corresponds to the ball-arrangement game that uses different moves. Before doing so, we introduce some operators, which will be useful in defining these networks. These generators corresponds to the actions that insert the outside ball into the leftmost box in the ball-arrangement game.

Definition 3.2 (Insertion Generator I_i): Given a permutation $U = u_{1:k}$, we define the *dimension-i insertion generator* I_i, $i = 2,3,...,k$, as the operator that cyclicly shift the leftmost i symbols $u_{1:i}$ to the left by one position.

In other words, for $i = 2,3,...,k$,

$$I_i(U) = u_{2:i}u_1u_{i+1:k}.$$

It can be viewed as inserting the outside ball to the $(i-1)^{th}$ position of the leftmost box (i.e., the i^{th} position from the left).

The following generators are the *inverse* of the corresponding insertion generators.

Definition 3.3 (Selection Generator I_i^{-1}): Given a permutation $U = u_{1:k}$, we define the *dimension-i selection generator* I_i^{-1}, $i = 2,3,...,k$, as the operator that cyclicly shift the leftmost i symbols $u_{1:i}$ to the right by one position.

In other words, for $i = 2,3,...,k$,

$$I_i^{-1}(U) = u_iu_{1:i-1}u_{i+1:k}.$$

It can be viewed as selecting the i^{th} ball from the leftmost box.

The last type of generators corresponds to the actions that cyclicly shift all the boxes.

Definition 3.4 (Rotation Generator R_n^i) : Given a permutation $U = u_{1:k}$, we define the *rotation generator* R_n^i as the operator that cyclicly shift the rightmost $k-1$ symbols $u_{2:k}$ to the right by ni positions.

Therefore, for $i = 2,3,...,l$, we have

$$R_n^i(u_{1:k}) = u_1 u_{k-in+1:k} u_{1:k-in}$$

In what follows, we will use R^i instead of R_n^i, suppressing the dependence on n, unless explicitly stated otherwise. We may also use R instead of R^1. We can see that

$$R^i = R^{i \bmod l} = \underbrace{RR\cdots R}_{i \bmod l}, \text{ and } R^i R^{-i}(U) = U.$$

We are now ready to define various interesting super Cayley graphs as directed or undirected Cayley graphs, Insertion, selection, and transposition generators will be used as nucleus generators, while swap and rotation generators will be used as super generators in these super Cayley graphs.

3.3.1 Star-Based Super Cayley Graphs

In this subsection, we define several super Cayley graphs that use the n-star as their basic building module. These networks are symmetric, undirected, and can efficiently embed and emulate star graphs and a variety of other important graphs, such as trees, meshes, hyprcubes, transposition networks, and bubble-sort networks.

Rotation-star (RS) networks and complete-rotation-star (complete-RS) networks are super Cayley graphs derived by the ball-arrangement game where boxes are moved by rotation and balls are moved by transposition.

Definition 3.5 (Rotation-Star Networks, RS(l,n))
An l-level rotation-star network based on an $(n+1)$-star graph is an undirected Cayley graph defined by

- transposition nucleus generators $T_2, T_3, ..., T_{n+1}$ and
- a pair of rotation super generators $R^1, R^{l-1} = R^{-1}$.

Rotation-star networks have constant degree when $n = O(1)$. An RS$(l,1)$ network can be viewed as an efficient degree-3 variant of the $(l+1)$-star graph [33].

Definition 3.6 (Complete-RS(l,n))
An l-level complete-rotation-star network based on an $(n+1)$-star graph is an undirected Cayley graph defined by

- transposition nucleus generators $T_2, T_3, ..., T_{n+1}$ and
- the complete set of rotation super generators $R^1, R^2, ..., R^{l-1}$.

3.3.2 Rotator-Based Super Cayley Graphs

In this subsection, we define several super Cayley graphs that use the n-dimensional rotator graphs [9] as their basic building module. The diameters of these networks are in general smaller than super Cayley graphs based on star graphs. In particular, the diameters of macro-rotator and complete-rotation-rotator networks are optimal within a factor of $1 + o(1)$ given their node degree. They are, however, directed and cannot emulate star graphs as efficiently as star-based super Cayley graphs.

Macro-rotator (MR) networks are super Cayley graphs derived by the ball-arrangement game where boxes are moved by transposition and balls are moved by insertion.

Definition 3.7 (Macro-Rotator Networks, MR(l,n)) : An l-level macro-rotator network based on an $(n+1)$-rotator graph is a directed Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$ and
- swap super generators $S_2, S_3, ..., S_l$.

Rotation-rotator (RR) networks and complete-rotation-rotator (complete-RR) networks are super Cayley graphs derived by the ball-arrangement game where boxes are moved by rotation and balls are moved by insertion.

Definition 3.8 (Rotation-Rotator Networks, RR(l,n)) : An l-level rotation-rotator network based on an $(n+1)$-rotator graph is a directed Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$ and
- a single rotation super generators R^1.

Definition 3.9 (Complete-RR(l,n)) : An l-level complete-rotation-rotator network based on an $(n+1)$-rotator graph is a directed Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$ and
- the complete set of rotation super generators $R^1, R^2, ..., R^{l-1}$.

3.3.3 IS-Based Super Cayley Graphs

In this subsection, we define several super Cayley graphs that use the insertion-selection (IS) network as their basic building module. The diameters of these networks are in general smaller than super Cayley graphs based on star graphs. In particular, the diameters of IS, macro-IS, and complete-rotation-IS networks are optimal within a factor of $1 + o(1)$ given their node degree. They are symmetric, undirected, and can efficiently embed and emulate star graphs and a variety of important graphs, such as trees, meshes, hyprcubes, transposition networks, and bubble-sort networks. Their degrees, however, are somewhat larger than those of star-based and rotator-based networks.

The insertion-selection (IS) network is defined as an undirected Cayley graph derived by the ball-arrangement game with one box and an outside ball, where balls are moved by insertion and selection. Insertion-selection networks can embed star graphs of the same size with congestion 1 and dilation 2, and emulate star graphs of the same size with a slowdown factor of at most 2 under any communication model.

Definition 3.10 (Insertion-Selection (IS) Networks)
A k-dimensional insertion-selection, k-IS, is an undirected Cayley graph defined by

- insertion generators $I_2, I_3, ..., I_k$ and
- selection generators $I_3^{-1}, I_4^{-1}, ..., I_k^{-1}$

Macro-IS (MIS) networks, rotation-IS (RIS) networks, and complete-rotation-IS (complete-RIS) networks are super Cayley graphs derived by the ball-arrangement game with l boxes, where balls in the leftmost box are moved by insertion and selection. The can be viewed as the undirected versions of rotator-based super Cayley graphs.

Definition 3.11 (Macro-IS Networks, MIS(l,n)): An l-level MIS network based on an $(n+1)$-IS network is an undirected Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$,
- selection nucleus generators $I_3^{-1}, I_4^{-1}, ..., I_{n+1}^{-1}$, and
- swap super generators $S_2, S_3, ..., S_l$.

For $n = 1$, the macro-star MS$(l,1)$, macro-rotator RS$(l,1)$, and macro-IS MIS$(l,1)$ are all identical to an $(l+1)$-star graph.

Definition 3.12 (Rotation-IS Networks, RIS(l,n))
An l-level RIS network based on an $(n+1)$-IS network is an undirected Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$,
- selection nucleus generators $I_3^{-1}, I_4^{-1}, ..., I_{n+1}^{-1}$, and
- a pair of rotation super generators R^1, R^{-1}.

Definition 3.13 (Complete-RIS(l,n)): An l-level complete-RIS network based on an $(n+1)$-IS network is an undirected Cayley graph defined by

- insertion nucleus generators $I_2, I_3, ..., I_{n+1}$,
- selection nucleus generators $I_3^{-1}, I_4^{-1}, ..., I_{n+1}^{-1}$, and
- the complete set of rotation super generators $R^1, R^2, ..., R^{l-1}$.

3.3.4 Other Super Cayley Graphs

Note that we can use a subset of rotation generators $R^1, R^2, ..., R^{l-1}$ to generate networks whose cost and performance fall between those of rotation-star networks (or RR and RIS networks) and complete-rotation-star networks (or complete-RR and complete-RIS networks). Moreover, if we remove all the links corresponding to nucleus generators, a rotation-star network (or RR or RIS network) is partitioned into $k!/l$ disjoint l-node rings; if we remove all the links corresponding to nucleus generators, a complete-rotation-star network (or complete-RR or complete-RIS network) is partitioned into $k!/l$ disjoint l-node complete graphs. We can replace each of these disjoint graphs with any connected small graph in order to obtain networks with desired cost and performance. We have used similar strategies in [31] for the design of cyclic networks.

We can also derive recursive versions of super Cayley graphs. For example, we can replace each of the nucleus $(n+1)$-stars of an MS(l,n) network (or the nucleus $(n+1)$-IS network of an RIS(l,n)) with a small MS(l_1,n_1) network (or an RIS(l_1,n_1) network) with $l_1 n_1 = n$. The resultant network is called a recursive MS(l,l_1,n_1) network (or a recursive RIS(l,l_1,n_1) network, respectively) and has node degree

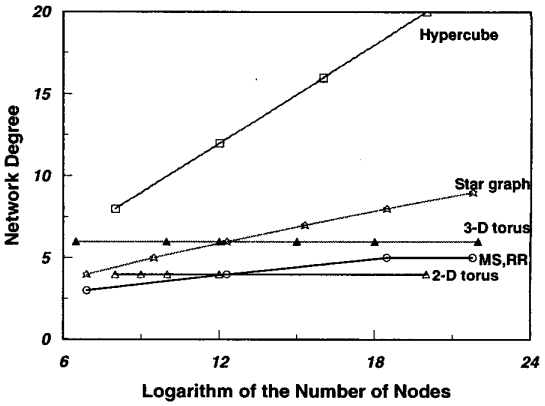

Figure 4: Comparison of the node degrees of various interconnection networks. The parameters of the MS and RR networks are $(2,2),(2,3),(2,4),(3,3)$.

smaller than that of an MS(l,n) network (or larger than that of an RIS(l,n) network, respectively). We can also combine different types of super generators to design such recursive Cayley graphs. We have used similar strategies in [31, 34, 37] for the design of recursive hierarchical swapped networks and recursive cyclic networks. The details for these networks are omitted in this paper.

4 Topological properties

In this section, we derive some basic properties of super Cayley graphs and compare them with those of other popular topologies.

4.1 Degree, diameter, and comparisons

The diameter of a super Cayley graph is equal to the best possible time required to solve the corresponding game for any initial configuration of the balls.

Theorem 4.1 *The diameter of a complete-rotation-star network, complete-RS(l,n), is at most equal to*

$$\lceil 2.5k \rceil + l - 4 = \Theta\left(\frac{\log N}{\log\log N}\right),$$

where $k = nl + 1$ and $N = k!$ is the number of nodes.

Theorem 4.2 *The diameter of a macro-star network, MS(l,n), is at most equal to*

$$\lceil 2.5k \rceil + 2l - 6 = \Theta\left(\frac{\log N}{\log\log N}\right).$$

Theorem 4.3 *The diameter of a complete-rotation-rotator network (complete-RR(l,n)) or a complete-RIS(l,n) network is at most equal to*

$$2k + 2l - 3 = \Theta\left(\frac{\log N}{\log\log N}\right).$$

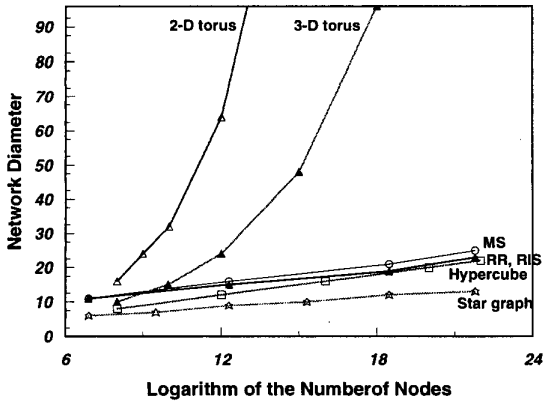

Figure 5: Comparison of the diameters of various interconnection networks. The parameters of the MS, RR, and RIS networks are $(2,2),(2,3),(2,4),(3,3)$.

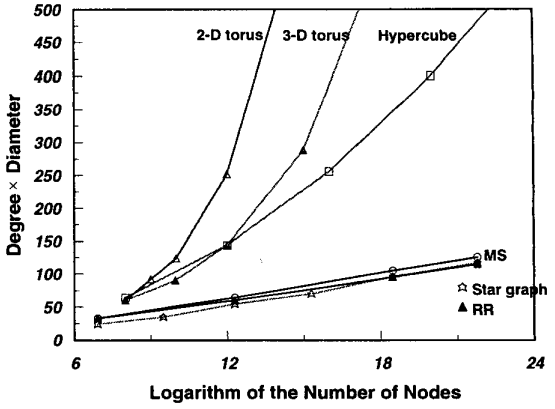

Figure 6: Comparison of the product of degree and diameter for various interconnection networks The parameters of the MS and RR networks are $(2,2),(2,3),(2,4),(3,3)$.

The diameter of a macro-rotator network $MR(l,n)$ or macro-insertion-selection network $MIS(l,n)$ is at most equal to

$$2k+3l-5=\Theta\left(\frac{\log N}{\log\log N}\right).$$

The number l of hierarchical levels in N-node super Cayley graphs discussed so far in the paper is

$$l=\Theta\left(\frac{\log N}{n\log\log N}\right). \quad (1)$$

The node degree of these super Cayley graphs is minimized when $l=\Theta(n)=\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$, leading to the following lemma.

Theorem 4.4 *The node degree of an N-node $MS(l,n)$, $MR(l,n)$, $MIS(l,n)$, complete-$RS(l,n)$, complete-$RR(l,n)$, or complete-$RIS(l,n)$ network is minimized and is equal to $\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ if and only if $l=\Theta(n)$.*

Since $O(n+l)=O\left(\frac{\log N}{\log\log N}\right)$ for any positive integers n and l, the node degree of the above super Cayley graphs can take values in the range from $\Omega\left(\sqrt{\frac{\log N}{\log\log N}}\right)$ to $O\left(\frac{\log N}{\log\log N}\right)$, depending on the particular choice of the parameters n and l. The degrees of rotation-star, rotation-rotator, and rotation-IS networks are constant when $n=O(1)$, and can take values in the range from $O(1)$ to $O\left(\frac{\log N}{\log\log N}\right)$, depending on the particular choice of the parameters n and l.

4.2 Optimal diameter and average distance

In [32], we defined the *universal lower bound* $D_L(N,d)$ on the diameter of a static undirected interconnection network that has N nodes and degree $d \geq 3$ as

$$D_L(N,d) \stackrel{\text{def}}{=} \log_{d-1} N + \log_{d-1}(1-\frac{2}{d}). \quad (2)$$

Also, for a given graph G, we defined the asymptotic diameter to lower-bound ratio

$$a(G)=\lim_{N\to\infty}\frac{D(G)}{D_L(N,d)}.$$

Theorem 4.5 *Any MS, MR, MIS, complete-RS, complete-RR, or complete-RIS network has asymptotically optimal diameter.*

Theorem 4.6 *The asymptotic diameter to lower-bound ratio of an $MR(l,n)$, $MIS(l,n)$, complete-$RR(l,n)$, or complete-$RIS(l,n)$ network is $a=1$, when $l=\Theta(n)$ [that is, when these super Cayley graphs are balanced].*

For $l=\Theta(n)$, an N-node MS, MR, MIS, complete-RS, complete-RR, or complete-RIS network has node degree

$$d=\Theta\left(\sqrt{\frac{\log N}{\log\log N}}\right)$$

and diameter

$$\frac{2.5\log_2 N}{\log_2\log_2 N}+o\left(\frac{\log N}{\log\log N}\right) \text{ or } \frac{2\log_2 N}{\log_2\log_2 N}+o\left(\frac{\log N}{\log\log N}\right)$$

both of which are sub-logarithmic. The asymptotic diameter to lower-bound ratios for several balanced super Cayley graphs and several interconnection networks of interest are summarized in Table 1.

Theorem 4.7 *The asymptotic average distance to lower-bound ratio of an $MS(l,n)$, $MR(l,n)$, $MIS(l,n)$, complete-$RS(l,n)$, complete-$RR(l,n)$, or complete-$RIS(l,n)$ network is $a_A=1$, when $l=\Theta(n)$ [that is, when these super Cayley graphs are balanced].*

Figures 4 and 5 show the node degrees and diameters of various network topologies as a function of the network size. For network sizes that are expected to be practical in the near future

(e.g., $N \leq 10! \approx 3.6 \cdot 10^6$), degree at most equal to 5 is sufficient for star-based and rotator-based networks, which is smaller than constant-degree networks such as 3-D meshes/tori, pyramids, and multigrids; degree at most equal to 7 is sufficient for IS-based networks. Super Cayley graphs are also competitive in terms of the diameter $\times$ degree cost measure as shown in Fig. 6.

Although diameter and average distance may be less important for networks that use wormhole routing under light traffic, they are crucial to network performance under heavy load when the networks are pin-limited rather than area limited, a common condition in parallel architectures with multiple packaging levels [6]. In fact, the maximum throughput of a network is inversely proportional to these parameters for any switching technology under the constraint of constant pin-outs. More details can be found in [32].

4.3 Intercluster diameter, average intercluster distance, and bisection bandwidth

A parallel computer is typically built from several chips on a board, multiple boards in a cabinet, and several such cabinets interconnected together. Modules at each level of the packaging hierarchy have their respective characteristics in terms of the number of pins, maximum area/volume, minimum wire width, and the number of wires per link [6]. In this subsection, we consider the case where several nodes (processors, routers, and associated memory banks) of a network are implemented on a single chip (see [17] for an example), or more generally, a single module (e.g., chip, board, wafer, or multi-chip module (MCM)), and several chips are used to build the parallel architecture or a higher-level module. We refer to the former type of architectures that are built of chip-multiprocessors as *multiple chip-multiprocessors (MCMP)* [36].

We define the *intercluster degree* of a network as the maximum of the average-per-node intercluster links over all clusters. We also define the *intercluster distance* between a pair of nodes as the minimum number of intercluster transmissions required for routing between them, the *intercluster diameter* of a network as the maximum of the intercluster distance between any pair of nodes, and the *average intercluster distance* of a network as the average of the intercluster distances between all pairs of nodes. More details concerning the terminology and the properties of MCMPs can be found in [36]. In what follows, we investigate on these parameters for super Cayley graphs.

The intercluster degree of a network has a bearing on its implementation cost and communication performance. If a chip contains multiple nodes, a node has off-chip bandwidth w, and the intercluster degree is d_i, then an off-chip link will have bandwidth w/d_i. The intercluster degree of a super Cayley graph is equal to the number of super generators in its definition, and is usually a small number, leading to high bandwidth for its links.

The proofs for the following theorems and corollaries are similar to those in [31, 32, 37] and are omitted in this paper.

Theorem 4.8 *Assuming that a cluster is composed of exactly one nucleus, the intercluster diameter (or average intercluster distance) of an MS, MR, MIS, complete-RS, complete-RR, or complete-RIS is asymptotically optimal within a factor of* $1 + o(1)$ *from the corresponding lower bound given its intercluster degree if* $M = \log^{o(1)} N$, *where M is the number of nodes in a nucleus and N is the network size. The intercluster diameter (or average intercluster distance) of an MS, MR, MIS, complete-RS, complete-RR, or complete-RIS is asymptotically optimal within a constant factor from the corresponding lower bound given its intercluster degree if* $M = \log^{O(1)} N$.

To obtain networks with optimal intercluster diameters and optimal average intercluster distances when the clusters are larger, we have to resort to another class of networks derived from BAG, called *super-index-permutation graphs* [31, 36, 37]. The major difference between super Cayley graphs and super-index-permutation graphs is that some of the balls for a super-index-permutation graph are assigned with the same numbers. More details can be found in [31, 34, 36, 37].

A set B_i of links is a bisection of a network if the removal of B_i partitions the network into two parts that differ in size by at most one node. Let $B_{i,1}, B_{i,2}, B_{i,3}, \ldots, B_{i,L_i}$ denote the bandwidths of links in partition B_i. Then the *bisection bandwidth* B_B of a network is given by

$$B_B = \min_{\text{all } i}\left(\sum_{j=1}^{L_i} B_{i,j}\right)$$

for all possible bisections i of the network. Bisection bandwidth of a network is usually the limiting factor on the performance of communication-intensive tasks, such as total exchange and random routing (see [36]). In what follows we derive lower bounds on the bisection bandwidths of super Cayley graphs, assuming that each chip holds a single nucleus and on-chip links are made wide enough so that the bisection bandwidth is derived without removing any on-chip links.

Theorem 4.9 *The bisection bandwidth* B_B *of an MS, MR, MIS, RS, RR, RIS, complete-RS, complete-RR, or complete-RIS is lower bounded by*

$$B_B \geq \frac{wN}{4D_{I,ave}},$$

where w is the average aggregate off-chip bandwidth of a node and $D_{I,ave}$ *is the average intercluster distance assuming that each chip holds a single nucleus.*

Since the average intercluster distance of an N-node MS, MR, MIS, complete-RS, complete-RR, or complete-RIS is $\Theta\left(\frac{\log N}{\log\log N}\right)$, the lower bounds on their bisection bandwidths are larger than the bisection bandwidths of hypercubes, CCCs, and k-ary n-cubes (see [36] for more details).

5 Conclusions

Desirable properties in interconnection networks for parallel systems include small average distance, high bisection bandwidth, symmetry, modularity, ease of mapping efficient algorithms onto them, and reasonable implementation cost. The hypercube and star graph meet most of these requirements, but their node degrees and intercluster degrees are prohibitively large for networks of medium to large size. The super Cayley graphs proposed in this paper form a new class of interconnection networks for the modular construction of parallel computers. Super Cayley graphs have several desirable algorithmic and topological properties, such as high bisection bandwidth when implemented as MCMPs, while using nodes of small degree and intercluster degree. We showed that suitably constructed super Cayley graphs have asymptotically optimal diameter, average

distance, intercluster diameter, and average intercluster distance within factors of $1 + o(1)$.

Constant-dilation embeddings of a variety of important topologies, such as trees, meshes, hypercubes, star graphs, bubble-sort graphs, and transition networks, are available for some of these super Cayley graphs. Also, there exist efficient algorithms for super Cayley graphs to emulate star graphs, leading to asymptotically optimal algorithms to execute MNB and TE tasks in super Cayley graphs under both the single-port and the all-port communication models. In all routing and parallel algorithms we have developed thus far, the expected traffic is balanced on all links of suitably constructed super Cayley graphs. More details will be reported in the near future.

References

[1] Akers, S.B., D. Harel, and B. Krishnamurthy, "The star graph: an attractive alternative to the n-cube," *Proc. Int'l Conf. Parallel Processing,* 1987, pp. 393-400.

[2] Akers, S.B. and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Trans. Comput.,* Vol. 38, Apr. 1989, pp. 555-565.

[3] Akl, S.G., K. Qiu, and I. Stojmenovic, "Fundamental algorithms for the star and pancake interconnection networks with applications to computational geometry," *Networks,* Vol. 23, Jul. 1993, pp. 215-225.

[4] Al-Ayyoub, A.-E. and K. Day, "Matrix decomposition on the star graph," *IEEE Trans. Parallel and Distributed Systems,* Vol. 8, no. 8, Aug. 1997, pp. 803-812.

[5] Bagherzadeh, N., M. Dowd, and S. Latifi, "A well-behaved enumeration of star graphs," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 6, no. 5, May 1995, pp. 531-535.

[6] Basak, D. and D.K. Panda, "Designing clustered multiprocessor systems under packaging and technological advancements," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 7, no. 9, Sep. 1996, pp. 962-978.

[7] Bertsekas, D.P. and J. Tsitsiklis, *Parallel and Distributed Computation: Numerical Methods,* Athena Scientific, 1997.

[8] Bouabdallah, A., M.C. Heydemann, J. Opatrny, and D. Sotteau, "Embedding complete binary trees into star networks," *Proc. Int'l Symp. Mathematical Foundations of Computer Science,* 1994, pp. 266-275.

[9] Corbett, P.F., "Rotator graphs: an efficient topology for point-to-point multiprocessor networks," *IEEE Trans. Parallel Distrib. Sys.,* vol. 3, no. 5, pp. 622-626, Sep. 1992.

[10] De Azevedo, M.M., "Star-based interconnection networks and fault-tolerant clock synchronization for large multicomputers," Ph.D. dissertation, Dept. Electrical & Comp. Eng., Univ. of California, Irvine, 1997.

[11] Draper, R.N., "Ascend/descend algorithms on Cayley graphs," *Int'l Phoenix Conf. Computers and Communications,* 1995, pp. 249-255.

[12] Faber, V., J.W. Moore, and W.Y.C. Chen, "Cycle prefix digraphs for symmetric interconnection networks," *Networks,* Oct. 1993, vol. 23, no. 7, pp. 641-649.

[13] Fragopoulou, P. and S.G. Akl, "Optimal communication algorithms on star graphs using spanning tree constructions," *J. Parallel Distrib. Computing.,* Vol. 24, 1995, pp. 55-71.

[14] Fujita, S., "A fault-tolerant broadcast scheme in the star graph under the single-port, half-duplex communication model," *IEEE Trans. Computers,* Vol. 48, no. 10, Oct. 1999, pp. 1123-1126.

[15] Huang, J.-P., S. Lakshmivarahan, and S.K. Dhall, "Analysis of interconnection networks based on simple Cayley coset graphs," *Proc. IEEE Symp. Parallel and Distributed Processing,* 1993, pp. 150-157.

[16] Jwo J.S., S. Lakshmivarahan, and S.K. Dhall, "Embedding of cycles and grids in star graphs," *Proc. IEEE Symp. Parallel and Distributed Processing,* 1990, pp. 540-547.

[17] Kogge, P.M., "EXECUBE - a new architecture for scalable MPPs," *Proc. Int'l Conf. Parallel Processing,* Vol. I, 1994, pp. 77-84.

[18] Lakshmivarahan, S., J.-S. Jwo, and S.K. Dhall, "Symmetry in interconnection networks based on Cayley graphs of permutation groups: a survey," *Parallel Computing,* Vol. 19, no. 4, Apr. 1993, pp. 361-407.

[19] Latifi, S. and P.K. Srimani, "Transposition networks as a class of fault-tolerant robust networks," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 45, no. 2, Feb. 1996, pp. 230-238.

[20] Lo, N.W., B.S. Carlson, D.L. Tao, "Fault tolerant algorithms for broadcasting on the star graph network," *IEEE Trans. Computers,* Vol. 46, no. 12, Dec. 1997, pp. 1357-1362.

[21] Menn, A. and Somani A.K., "An efficient sorting algorithm for the star graph interconnection networks," *Proc. Int'l Conf. Parallel Processing,* Vol. III, 1990, pp. 1-8.

[22] Miller, Z., D. Pritikin, and I.H. Sudborough, "Bounded dilation maps of hypercubes into Cayley graphs on the symmetric group," *Math. Sys. Theory,* Vol. 29, no. 6, Springer-Verlag, Nov.-Dec., 1996, pp.551-572.

[23] Mišić, J. and Z. Jovanović, "Communication aspects of the star graph interconnection network," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 5, no. 7, Jul. 1994, pp. 678-687.

[24] Öhring, S., F. Sarkar, S.K. Das, and D.H. Hohndel, "Cayley graph connected cycles: a new class of fixed-degree interconnection networks," *Proc. Int'l Conf. Systems Sciences,* 1995, pp. 479-487.

[25] Saikia, D.K., R. Badrinath, and R.K. Sen, "Embedding torus on the star graph," *IEEE Trans. Parallel and Distributed Systems,* Vol. 9, no. 7, Jul. 1998, pp. 650-663

[26] Scherson, I.D., and A.S. Youssef, *Interconnection Networks for High-Performance Parallel Computers,* IEEE Computer Society Press, 1994.

[27] Sheu, J.-P., C.-T. Wu, and T.-S. Chen, "An optimal broadcasting Algorithm without message redundancy in star graphs," *IEEE Trans. Parallel Distrib. Sys.,* Vol. 6, no. 6, Jun. 1995, pp. 653-658.

[28] Tseng, Y.-C., S.-H. Chang, and J.-P. Sheu, "Fault-tolerant ring embedding in a star graph with both link and node failures," *IEEE Trans. Parallel and Distributed Systems,* Vol. 8, no. 12, Dec. 1997, pp. 1185-1195.

[29] Varvarigos, E.A., "Static and dynamic communication in parallel computing," Ph.D. dissertation, Dept. Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1992.

[30] Varvarigos, E.A. and D.P. Bertsekas, "Multinode broadcast in hypercubes and rings with randomly distributed length of packets," *IEEE Trans. Parallel Distrib. Sys.,* vol. 4, no. 2, Feb. 1993, pp. 144-154.

[31] Yeh, C.-H., "Efficient low-degree interconnection networks for parallel processing: topologies, algorithms, VLSI layouts, and fault tolerance," Ph.D. dissertation, Dept. Electrical & Computer Engineering, Univ. of California, Santa Barbara, Mar. 1998.

[32] Yeh, C.-H. and E.A. Varvarigos, "Macro-star networks: efficient low-degree alternatives to star graphs," *IEEE Trans. Parallel Distrib. Sys.,* vol. 9, no. 10, Oct. 1998, pp. 987-1003.

[33] Yeh, C.-H. and E.A. Varvarigos, "Parallel algorithms on the rotation-exchange network – a trivalent variant of the star graph," *Proc. Symp. Frontiers of Massively Parallel Computation,* Feb. 1999, pp. 302-309.

[34] Yeh, C.-H. and B. Parhami, "The index-permutation graph model for hierarchical interconnection networks," *Proc. Int'l Conf. Parallel Processing,* Sep. 1999, pp. 48-55.

[35] Yeh, C.-H. and B. Parhami, "On the VLSI area and bisection width of star graphs and hierarchical cubic networks," *Proc. Int'l Parallel and Distributed Processing Symp.,* 2001.

[36] Yeh, C.-H. and B. Parhami, "Parallel algorithms for index-permutation graphs – An extension of Cayley graphs for multiple chip-multiprocessors (MCMP)," *Proc. Int'l Conf. Parallel Processing,* 2001, to appear.

[37] Yeh, C.-H. and B. Parhami, "A unified model for hierarchical networks based on an extension of Cayley graphs," *IEEE Trans. Parallel Distrib. Sys.,* to appear.