

On The Shuffle-Exchange Permutation Network

Douglas W. Bass and I. Hal Sudborough

Computer Science Program, University of Texas at Dallas, Richardson, Texas USA

Abstract

The *shuffle-exchange permutation network* (SEP_n) is a fixed degree Cayley graph which has been proposed as a basis for massively parallel systems. We propose a routing algorithm with an upper bound of $(5/8)n^2 + O(n)$, where n is the length of the permutation. (This improves on a $(9/8)n^2$ routing algorithm described earlier [5].) Thus, the diameter of SEP_n is at most $(5/8)n^2 + O(n)$. We also show that the diameter is at least $n^2/2 - O(n)$. We demonstrate that SEP_n has a Hamilton cycle, for $n \geq 3$, left open in [5], and describe embeddings of variable-degree Cayley networks, such as bubble-sort networks [1], star networks [2] and pancake networks [4] into SEP_n . Our embeddings for these networks are substantial improvements of earlier results stated in [5].

Keywords: Cayley graphs, routing algorithms, Hamilton cycles, embeddings, shuffle-exchange permutation network.

1: Introduction

Cayley graphs [1] have proved to be a useful basis for interconnection networks for a number of reasons [3]. They are regular (unlike networks with a central hub) and vertex-symmetric (unlike two-dimensional meshes or DeBruijn networks[6]). Some examples of networks with low degree and diameter are shown in Table 1), with diameter results found in [1] and [4].

Network	Vertices	Degree	Diameter
Hypercube	2^n	n	n
Star	$n!$	$n - 1$	$\lceil 3(n - 1) / 2 \rceil$
Pancake	$n!$	$n - 1$	$\leq (5/3)n$

Table 1. A comparison of three Cayley graphs

One difficulty that hinders many proposed interconnection networks from being both large and expandable is the fact that the degree is a linear function of the dimension of the

network. For example, if the number of processors in a hypercube is doubled from 64 to 128, then an additional I/O port must be added to each processor. The shuffle-exchange permutation (SEP) network proposed by Latifi and Srimani [5] is an example of a fixed degree Cayley graph, with each node of degree 3. Specifically, SEP_n is a graph of $n!$ vertices, where each vertex is labeled with a permutation of the set $\{1, 2, \dots, n\}$. If a vertex is labeled with the permutation $\pi = (\pi_1 \pi_2 \dots \pi_n)$, there are edges from this vertex to the vertices with the labels $(\pi_2 \pi_3 \dots \pi_n \pi_1)$ (*left shuffle*), $(\pi_n \pi_1 \pi_2 \dots \pi_{n-1})$ (*right shuffle*) and $(\pi_2 \pi_1 \pi_3 \dots \pi_n)$ (*exchange of the first two elements*). Since SEP_n has $n!$ vertices, and is regular of degree 3, it has $(3/2)n!$ edges. Figure 1 is a planar depiction of SEP_4 . We refer the interested reader to [5] for additional justification, motivation and background information concerning SEP_n .

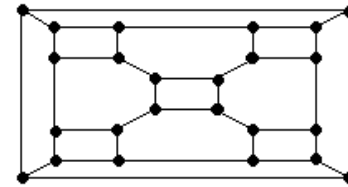


Figure 1. SEP_4

2: Routing

Routing from a vertex labeled with permutation α to a vertex labeled with permutation β is equivalent to routing from a vertex labeled with permutation $\alpha\beta^{-1}$ to the identity permutation. Therefore, a routing algorithm is equivalent to an algorithm for sorting elements in a permutation. Since left and right shuffles are among the available moves, it is useful to think of the elements $\pi_1, \pi_2, \dots, \pi_n$ of the permutation π as arranged in a circle, with, say, π_1 at the "north pole" of the circle, and the position antipodal to π_1 on this circle being the "south pole." ("Rotate" in the following means to use shuffles to get to a desired position.) We first construct a sorted block of the first $n/2$ elements. Let $N_i, 1 \leq i \leq n$, be the distance of i from the

north pole, Sum_N be the sum of N_i , for $1 \leq i \leq n/2$, S_i be the distance of i from the south pole, and Sum_S be the sum of S_i , for $1 \leq i \leq n/2$. If $\text{Sum}_S < \text{Sum}_N$, we will gather elements to the south pole, else to the north pole. We build a sorted block by moving the first $n/2$ elements to the appropriate pole. Once the first $n/2$ elements are sorted, the remaining elements are added to either the left or right end of the sorted block (as appropriate).

Algorithm ROUTE(π)

```

begin
  Compute  $N_i, S_i$  for  $1 \leq i \leq n$ , and  $\text{Sum}_N$  and  $\text{Sum}_S$ 
  if  $\text{Sum}_S < \text{Sum}_N$  then pole ← south else pole ← north
  for each item in  $\{1, 2, \dots, n/2\}$  do
    Move  $i$  to the pole, keeping the center of the
      sorted block at the pole.
  Move to one of the ends of the sorted block.
  for  $i = (n/2) + 1$  to  $n$  do
    Add an element to an end of the sorted block.
    Rotate as necessary.
end

```

Theorem 1 Algorithm ROUTE sorts the elements of a permutation of length n in no more than $(5/8)n^2 + O(n)$ moves.

Sketch of Proof: (Figures in bold are components of the final total.) It takes $2k - 1$ steps to move a position 1 element k positions clockwise, and $2k - 1$ steps to move a position 2 element k positions counterclockwise. Assume we gather elements to the north pole. If we start at the north pole, and wish to move i to the north pole, then it may take N_i shuffles to rotate, and then $2N_i - 1$ moves to move i to the north pole. Therefore, it takes $3N_i - 1$ moves to move i to the north pole. Let $M = \min(\text{Sum}_N, \text{Sum}_S)$. We observe that since $S_i = N_i - n/2 - 1$ for $1 \leq i \leq n/2$, $\text{Sum}_S = \text{Sum}_N - n^2/4 - n/2$. Therefore $M = \min(\text{Sum}_N, \text{Sum}_N - n^2/4)$. It follows that M has a maximum value when $\text{Sum}_N = \text{Sum}_S$. This occurs when the first $n/2$ elements are arranged about the equator, i.e., halfway

between poles. When this is the case, $M = 2 \sum_{i=n/8}^{3n/8} i = n^2/8$

+ $O(n)$. Therefore, it takes at most $(3/8)n^2 + O(n)$ moves to move the first $n/2$ elements to the north pole. Rotating to one of the ends of the sorted block takes $O(n)$ moves. Now consider the number of moves needed to put the last $n/2$ elements in place. Suppose there are i elements in the sorted block, where $n/2 \leq i \leq n$. The worst case is when an element to be added has to be moved $n - i$ positions. Whichever end of the sorted block we are at, in such a case, the element to be added to the other end is one

rotation away, then $2(n - i) - 1$ moves to put it to the proper spot. Therefore, to put the remaining $n/2$ elements into the sorted block, the number of moves required is at

worst $\sum_{i=n/2}^n 2(n - i) = n^2/4 + O(n)$. Once the sorted block is

complete, we may need to rotate, but the total number of moves required is no greater than $(5/8)n^2 + O(n)$. □

Consider the permutation $\{3\ 7\ 5\ 1\ 10\ 6\ 9\ 4\ 2\ 8\}$. 3 is at the north pole, and 6 is at the south pole. The N_i , for $i = 1$ to 5, are 3, 2, 0, 3 and 2 which makes $\text{Sum}_N = 10$. The S_i are 2, 3, 5, 2 and 3, therefore $\text{Sum}_S = 15$. Therefore, we will gather the elements to the north pole.

```

3 7 5 1 10 6 9 4 2 8  $\xrightarrow{(8)}$  3 4 7 5 1 10 6 9 2 8
3 4 7 5 1 10 6 9 2 8  $\xrightarrow{(10)}$  1 3 4 7 5 10 6 9 2 8
1 3 4 7 5 10 6 9 2 8  $\xrightarrow{(5)}$  1 2 3 4 7 5 10 6 9 8
1 2 3 4 7 5 10 6 9 8  $\xrightarrow{(5)}$  5 7 10 6 9 8 1 2 3 4
5 7 10 6 9 8 1 2 3 4  $\xrightarrow{(5)}$  6 7 10 9 8 1 2 3 4 5
6 7 10 9 8 1 2 3 4 5  $\xrightarrow{(5)}$  8 10 1 2 3 4 5 6 7 9
8 10 1 2 3 4 5 6 7 9  $\xrightarrow{(2)}$  8 9 10 1 2 3 4 5 6 7
8 9 10 1 2 3 4 5 6 7  $\xrightarrow{(3)}$  1 2 3 4 5 6 7 8 9 10

```

2.1: A lower bound for the diameter of SEP_n

It is well known that any sorting algorithm that works by exchanging adjacent elements requires $\Omega(n^2)$ steps (see, for example, [8]). Any sorting algorithm using the moves of SEP_n is subject to this same lower bound.

Theorem 2 For $n > 1$, the diameter of SEP_n is at least $(n^2/2) - O(n)$.

Sketch of Proof: Let $d(i, j)$ be the circular distance between i and j in π , a permutation of length n . Let $S = \sum_{i=1}^{n-1} d(i, i+1)$. S can be, as large as $(n^2/2) + O(n)$.

Consider the permutation $\pi = \{1\ 3\ 5\ \dots\ n-1\ 2\ 4\ \dots\ n\}$. The value of S is $(n - 1)((n/2) - 1) + (n/2) = (n^2/2) + O(n)$. When π is the identity, the value of S is $n - 1$. So, sorting must reduce S in this case from $(n^2/2)$ to $O(n)$. The left and right shuffle operations do not change the value of S at all, and the exchange operation changes the value of S by at most 2. And, after making an exchange, a shuffle must be performed, otherwise the exchange is lost. So, in any two moves, S can be at most reduced by 2. Hence, it takes at least $(n^2/2) - O(n)$ moves to sort in the worst case. □

3: The Hamilton Cycle for SEP_n

Every vertex of SEP_n belongs to a simple cycle of n vertices. These cycles correspond, for example, to n successive left shuffles. If a simple cycle touches each of these cycles in one and only one edge, then that cycle could be expanded into a Hamilton cycle. The dotted lines in Figure 2 signify an edge that is present in SEP_4 , but not being used for the Hamilton cycle. In the figure, the edges (D, E), (H, I), (L, M), (P, Q), (T, U) and (X, A) represent exchanges. All the other edges represent shuffles. There is such a Hamilton cycle for SEP_4 with the property that the element 4 is never in the first or second position when an exchange is made. That is, 4 never gets exchanged.

The above cycle of 24 vertices in SEP_4 can be modified into a cycle of 30 vertices in SEP_5 by the following steps:

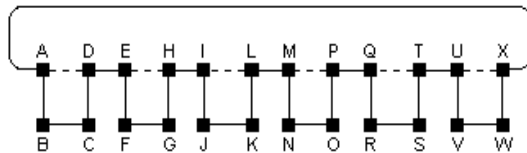


Figure 2. A Hamilton cycle for SEP_4

- 1) Replace all occurrences of the element 4 in each vertex with 4 5
- 2) Replace a right shuffle with 5 on the right end by two right shuffles
- 3) Replace a left shuffle with 4 on the left end by two left shuffles

Since this cycle is a modified copy of SEP_4 , and it was generated by replacing 4 with 4 5, this cycle will be called the **{5}-copy**. By a similar procedure, the {1}-copy, {2}-copy and {3}-copy can also be created. Because of the method of constructing these cycles in SEP_5 , each vertex of SEP_5 is in one and only one of these cycles. To find out which cycle a given vertex is in, simply look at whatever element is to the right of the element 4. These cycles can be linked together to form a Hamilton cycle for SEP_5 in the manner described by Figure 3.

Notice that instead of jumping from the {5}-copy to the {1}-copy, we could have chosen to simply traverse the {5}-copy. In the same way, instead of jumping from the {1}-copy to the {2}-copy, we could have chosen to traverse all of the {5}-copy and {1}-copy. If one copy is traversed, we have the option of traversing none, some or all of the remaining copies. Another property of the copies in SEP_5 is that each copy can directly access each of the other copies.

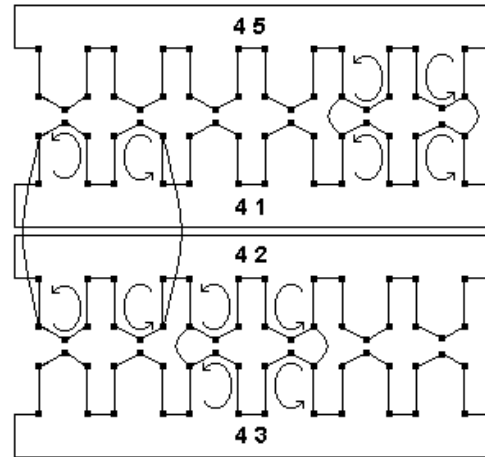


Figure 3. A Hamilton cycle for SEP_5

A π -copy in SEP_n is a modified copy of the cycle in Figure 2, where

- 1) All occurrences of the element 4 in each vertex are replaced with 4 π , where π is a permutation of $n - 4$ of the elements of $\{1, 2, 3, 5, \dots, n\}$
- 2) A right shuffle with π_{n-4} on the right end is replaced by $n - 3$ right shuffles
- 3) A left shuffle with 4 on the left end is replaced by $n - 3$ left shuffles.

For example, there are 120 π -copies in SEP_7 , and every vertex of SEP_7 is in one and only one π -copy. These 120 π -copies can be grouped into 30 groups of 4 π -copies, where each group is a modified copy of SEP_5 . Since each π -copy can access each of the other three members of its group, if a simple cycle can be constructed which accesses at least one π -copy of each group, then that cycle can be expanded to create a Hamilton cycle for SEP_7 . Figure 4 describes a simple cycle which touches 6 π -copies in one and only one edge. The arrows represent exchanges. This cycle will be called a **hexagon**. A hexagon has the effect of permuting the three numbers following 4. By using a hexagon, 6 of the 120 cycles of SEP_7 , and 6 of the 30 groups of 4 cycles have been accessed. By using the method shown in Figure 3, the {6 7 5}-copy can directly access the {6 7 1}-copy. Because of this, 6 more cycles and 4 more groups can be accessed by accessing the hexagon of which the {6 7 1}-copy is a member. When this process is repeated four more times, 30 of the cycles and 24 of the groups have been accessed. The remaining 6

	4->4567	4->4576	4->4756	
	1234567	1234576	1234756	
	7123456	6123457	6123475	
	6712345	→ 7612345	5612347	→ 6512347
6571234	→ 5671234	5761234	→ 7561234	
	4567123	4576123	4756123	
	3456712	3457612	3475612	
	2345671	2345761	2347561	
	4->4765	4->4675	4->4657	
	1234765	1234675	1234657	
	5123476	5123467	7123465	
5612347	→ 6512347	7512346	→ 5712346	
	7651234	→ 6751234	6571234	→ 5671234
	4765123	4675123	4657123	
	3476512	3467512	3465712	
	2347651	2346751	2346571	

Figure 4. A cycle which touches 6 π -copies in one and only one edge

groups are accessed in a similar fashion. Figure 5 is a picture of the resulting cycle, which can be expanded to create the Hamilton cycle for SEP_7 by accessing the cycles which haven't been touched yet.

$SEP_{n,n-1,k}$ is a subgraph of SEP_n , where k is an element of $\{1,2,3,5, \dots, n\}$, and a vertex v in SEP_n is also in $SEP_{n,n-1,k}$ if and only if v is a vertex in a π -copy, where the first element of π is k . If two vertices u and v are in $SEP_{n,n-1,k}$ and there is an edge between u and v in SEP_n , then there is an edge between u and v in $SEP_{n,n-1,k}$. The subgraph $SEP_{n,n-1,k}$ is a modified copy of $SEP_{(n-1)}$ inside SEP_n , with $n^*(n-2)!$ vertices. Using the cycle we have constructed for SEP_7 , we will construct cycles for $SEP_{8,7,1}$, $SEP_{8,7,2}$, ..., $SEP_{8,7,8}$. We will then link those cycles to construct a Hamilton cycle for SEP_8 . The same procedure will be followed for $SEP_9, SEP_{10}, \dots, SEP_n$.

The **sibling** of $SEP_{n,n-1,k}$, or **sibling($SEP_{n,n-1,k}$)**, is $SEP_{n,n-1,m}$, where $m = (k+j) \bmod n$, where j is the smallest positive number such that $(k+j) \bmod n$ is an element of $\{1, 2, 3, 5, \dots, n\}$. The cycle for sibling($SEP_{n,n-1,i}$) will be linked with the cycle for $SEP_{n,n-1,i}$ to form the Hamilton cycle for SEP_n . $SEP_{n,n-1,k}$ has one and only one sibling. For example, sibling($SEP_{8,7,1}$) = $SEP_{8,7,2}$ and the sibling of $SEP_{10,9,10}$ is $SEP_{10,9,1}$. Let sibling²($SEP_{n,n-1,k}$) have its expected definition.

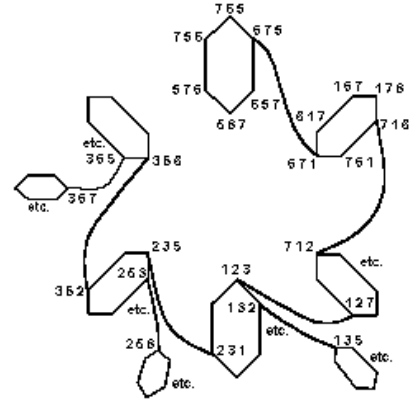


Figure 5. A framework for the Hamilton cycle for SEP_7

Let the π of a π -copy of SEP_n be $\{\pi_{n-4}, \pi_{n-5}, \dots, \pi_1\}$. The **k-hexagon**, for $7 \leq k \leq n-4$, touches the following π -copies in one and only one edge, in the following order.

- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_1\}$ -copy
- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_{k-4} \pi_{k-6} \pi_{k-5} \dots \pi_1\}$ -copy
- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_{k-6} \pi_{k-4} \pi_{k-5} \dots \pi_1\}$ -copy
- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_{k-6} \pi_{k-5} \pi_{k-4} \dots \pi_1\}$ -copy
- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_{k-5} \pi_{k-6} \pi_{k-4} \dots \pi_1\}$ -copy
- the $\{\pi_{n-4} \pi_{n-5} \dots \pi_{k-5} \pi_{k-4} \pi_{k-6} \dots \pi_1\}$ -copy

The k -hexagon will be used to link cycles for $SEP_{k,k-1,m}$, where $m = 1, 2, 3, 5, \dots, k$, creating a cycle for SEP_k . For example, by using the 8-hexagon, the $\{1 2 3 5\}$ -copy in SEP_8 can access the $\{1 3 2 5\}$, $\{3 1 2 5\}$, $\{3 2 1 5\}$, $\{2 3 1 5\}$ and $\{2 1 3 5\}$ -copies.

Lemma 1: If a Hamilton cycle can be constructed for SEP_j which connects π -copies by hexagons, and by the method used to construct SEP_5 , then a cycle can be constructed for $SEP_{j+1,j,k}$, $k = 1, 2, 3, 5, \dots, j+1$, which connects π -copies by hexagons, and by the method used to construct SEP_5 .

In constructing the Hamilton cycle for SEP_n , we assume we have constructed $n-1$ copies of the cycle for $SEP_{n,n-1,k}$, $k = 1, 2, 3, 5, \dots, n$.

Lemma 2: There exist two π -copies in $SEP_{n,n-1,k}$, two π -copies in sibling($SEP_{n,n-1,k}$) and two π -copies in sibling²($SEP_{n,n-1,k}$) such that a n -hexagon touches each of these π -copies in one and only one edge.

We can always construct a simple cycle which touches parts of $SEP_{n,n-1,k}$, its sibling and its sibling's sibling.

Lemma 3: If cycles for $SEP_{n,n-1,k}$, $\text{sibling}(SEP_{n,n-1,k})$ and $\text{sibling}^2(SEP_{n,n-1,k})$ can be constructed where π -copies are connected by hexagons and the method used to construct SEP_5 , then a cycle can be constructed which accesses all of $SEP_{n,n-1,k}$, all of $\text{sibling}(SEP_{n,n-1,k})$ and all of two π -copies in $\text{sibling}^2(SEP_{n,n-1,k})$, which connects π -copies by hexagons, and by the method used to construct SEP_5 .

We can take the cycle for $SEP_{n,n-1,k}$, the cycle for $\text{sibling}(SEP_{n,n-1,k})$, and two π -copies of $\text{sibling}^2(SEP_{n,n-1,k})$, and merge them into one cycle. (See Figure 6)

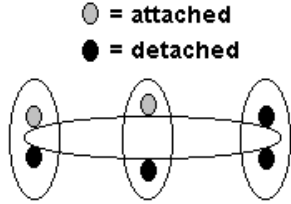


Figure 6. Merging cycles for two copies of SEP_{n-1} within SEP_n into one cycle

Lemma 4: If cycles can be created for $SEP_{n,n-1,1}$, $SEP_{n,n-1,2}$, $SEP_{n,n-1,3}$, $SEP_{n,n-1,5}$, ... $SEP_{n,n-1,n}$ can be constructed where π -copies are connected by hexagons, and by the method used to construct SEP_5 , then a cycle can be constructed which includes all of $SEP_{n,n-1,1}$, $SEP_{n,n-1,2}$, $SEP_{n,n-1,3}$, $SEP_{n,n-1,5}$, ... $SEP_{n,n-1,n}$, which connects π -copies by hexagons, and by the method used to construct SEP_5 .

We can take cycles for the $n-1$ copies of SEP_{n-1} within SEP_n , and merge them into one cycle. (See Figure 7)

Algorithm SEP-HAMILTON-CYCLE(n)

```

begin
  for i = 8 to n do
    begin
      for j = 1, 2, 3, 5, ..., i do
        begin
          Construct  $SEP_{i,i-1,j}$  from  $SEP_{i-1}$ 
          Using the connections in the
          cycle for  $SEP_{i-1}$ , construct
          a cycle for  $SEP_{i,i-1,j}$ 
        end
      for j = 1, 2, 3, 5, ... i - 1 do
        Use the i-hexagon to link the cycle
        for  $SEP_{i,i-1,j}$  with the cycles
        for its sibling and its sibling's
        sibling, adjusting the links as
        described in Lemma 3.
      end
    end
  end
end

```

Theorem 3 Algorithm SEP-HAMILTON-CYCLE produces a Hamilton cycle for SEP_n , for all $n > 7$.

Proof: The algorithm starts with SEP_7 , which we know has a Hamilton cycle which connects π -copies by hexagons and by the method used for constructing the cycle for SEP_5 and iteratively builds up to cycles for larger and larger values of n , by making 7 copies of the cycle for SEP_7 , 8 copies of the cycle for SEP_8 , ... $n-1$ copies of the cycle for SEP_{n-1} . The first loop is possible by Lemma 1, and the second loop is possible by Lemmas 2, 3 and 4. For each value of i , a hexagon is used that was not used for any previous values of i . This is how we know the cycle for SEP_i won't interfere with the cycles for smaller values of i .

□

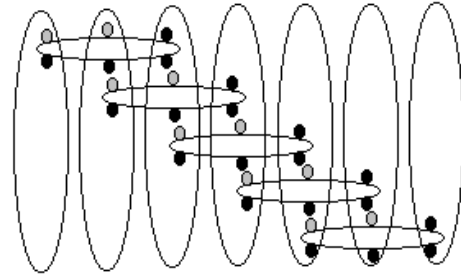


Figure 7. Merging cycles for $n-1$ copies of SEP_{n-1} within SEP_n into one cycle

4. Network embeddings

Since the efficiency of many parallel algorithms is dependent on the topology of the interconnection network, it is useful to determine a rough upper bound on the number of SEP moves required to simulate moves in other networks. Each of the three Cayley networks we consider is defined on S_n , the group of permutations on n , and was previously considered in [5]. In each case, we improve on the earlier results, which were all based on embeddings given by the identity map.

- The **Star** network of dimension n : S_n [2]. If a vertex is labeled with the permutation $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, then there are edges from this vertex to vertices with the labels $(\pi_k, \pi_2, \dots, \pi_{k-1}, \pi_1, \pi_{k+1}, \dots, \pi_n)$, for $2 \leq k \leq n$ (swapping the first and the k^{th} element)
- The **Bubble-Sort** network of dimension n : B_n [1]. If a vertex is labeled with the permutation $\pi = (\pi_1, \pi_2, \dots, \pi_n)$, then there are edges from this vertex to vertices with the labels $(\pi_1, \pi_2, \dots, \pi_{k+1}, \pi_k, \dots, \pi_n)$, for $1 \leq k < n$ (swapping two adjacent elements)

- The **Pancake** network of dimension n : P_n [4]. If a vertex is labeled with the permutation $\pi = (\pi_1, \pi_2 \dots \pi_n)$, then there are edges from this vertex to vertices with the labels $(\pi_k \pi_{k-1} \dots \pi_2 \pi_1 \pi_{k+1} \dots \pi_n)$, for $2 \leq k \leq n$ (*flipping the leftmost k elements, i. e., a prefix reversal of size k*)

For the Star network, we use the fact that the Star network is bipartite, with a bipartition based on classifying permutations by whether they can be expressed as an even or odd number of transpositions.

Let the embedding M_1 from Star nodes to SEP nodes be as follows:

- If π is an even permutation, then $M_1(\pi) = \pi$.
- If π is odd, $M_1(\pi) = (\pi_1 \pi_{n/2+1} \dots \pi_n \pi_2 \dots \pi_{n/2})$

Theorem 4. M_1 embeds S_n into SEP_n with dilation $n - 1$.

Sketch of Proof: It is sufficient to consider the distance between the image of an arbitrary odd permutation and the image of any of its neighbors. Suppose $\pi = (\pi_1, \pi_2 \dots \pi_n)$ is an odd permutation. Then it maps by M_1 to $\pi' = (\pi_1 \pi_{n/2+1} \dots \pi_n \pi_2 \pi_3 \dots \pi_{n/2})$. Any neighbor of π must be an even permutation and hence maps by M_1 to itself. Suppose we consider the image of π after the transposition $(1, k)$. For a dilation bound, we want to measure the distance from $(\pi_k \pi_2 \dots \pi_{k-1} \pi_1 \pi_{k+1} \dots \pi_n)$ to π' .

Case 1: $k = 2$. Then a path from $(\pi_2 \pi_1 \pi_3 \dots \pi_n)$ to π' is described by: rotate π_1 to position 1, then move π_1 $(n/2) - 2$ positions right. The total is $n - 3$ moves.

Case 2: $2 < k \leq (n/2)$. Find the distance from $(\pi_k \pi_2 \dots \pi_{k-1} \pi_1 \pi_{k+1} \dots \pi_n)$ to π' . Move π_k $k - 2$ positions right. Rotate so that π_1 is in position 1. Move π_1 $(n/2) - k$ positions right. The total is $n - 3$ moves.

Case 3: $k = (n/2) + 1$. Find the distance from $(\pi_{(n/2)+1} \pi_2 \dots \pi_{n/2} \pi_1 \pi_{(n/2)+2} \dots \pi_n)$ to π' . Move $\pi_{(n/2)+1}$ $(n/2)$ positions right, for a total of $n - 1$ moves.

Case 4: $(n/2) + 1 < k < n$. Find the distance from $(\pi_k \pi_2 \dots \pi_{k-1} \pi_1 \pi_{k+1} \dots \pi_n)$ to π' . Rotate so that π_k is in position 2. Move π_k $n - k$ positions left. Rotate so that π_1 is in position 2, then move π_1 $k - (n/2) - 1$ positions left. The total is $n - 1$ moves.

Case 5: $k = n$. Find the distance from $(\pi_n \pi_2 \dots \pi_{n-1} \pi_1)$ to π' . Rotate π_1 into position 2. Move π_1 $(n/2) - 1$ positions left. The total is $n - 1$ moves. \square

Theorem 4 represents a notable improvement over a so-called optimum result in [5], where the dilation was $2(n - 1)$.

The bubble sort network is also bipartite, with the same bipartition as the Star network. Let the mapping M_2 be as follows:

- If π is an even permutation, then $M_2(\pi) = \pi$.
- If π is odd, $M_2(\pi) = (\pi_{(n/2)+1} \dots \pi_n \pi_1 \pi_2 \dots \pi_{n/2})$

Theorem 5 M_2 embeds B_n into SEP_n with dilation $(n/2) + 1$.

Sketch of Proof: Suppose $\pi = (\pi_1, \pi_2 \dots \pi_n)$ is odd. Then it maps to $\pi' = (\pi_{(n/2)+1} \dots \pi_n \pi_1 \pi_2 \dots \pi_{n/2})$. Any permutation that is a neighbor of π must be even and hence maps to itself by M_2 . Consider the Bubble-Sort move swapping π_{k+1} with π_k . We measure the distance between their images, i.e. from $(\pi_1 \pi_2 \dots \pi_{k+1} \pi_k \dots \pi_n)$ to π' . One can easily find a routing with at most $(n/2) + 1$ moves, as one needs to rotate to the desired exchange, and then rotate to the middle position. This takes at most $n/2$ moves as one rotates to the middle. The remainder of the proof is very similar to the proof of Theorem 4. \square

Theorem 5 is also a significant improvement over a so-called optimum result in [5], where the dilation was given to be $n + 1$.

Unlike the Star and Bubble-Sort networks, the pancake network is not bipartite. Morales and Sudborough [7] demonstrated that P_n can be embedded in S_{2n} with dilation 1, therefore P_n can be embedded in SEP_{2n} with dilation $2n - 1$. However, we wish to embed P_n in SEP_n . Our embedding M_3 is as follows: Take the last $n/2$ elements, reverse their order, and perfectly shuffle them with the first $n/2$ elements, yielding $\pi' = (\pi_1 \pi_n \pi_2 \pi_{n-1} \dots \pi_{n/2} \pi_{(n/2)+1})$.

Suppose a move in P_n flips all the elements, going from $(\pi_1 \pi_2 \dots \pi_n)$ to $(\pi_n \pi_{n-1} \dots \pi_2 \pi_1)$. To represent that move in SEP_n using M_3 , we must route from $(\pi_1 \pi_n \pi_2 \pi_{n-1} \dots \pi_{n/2} \pi_{(n/2)+1})$ to $(\pi_n \pi_1 \pi_{n-1} \pi_2 \dots \pi_{(n/2)+1} \pi_{n/2})$. This can be done in $(3/2)n$ moves by repeatedly doing an exchange followed by two rotations. Similarly, a move in P_n where all but one of the elements are flipped can be represented in SEP_n using M_3 in $5n - 10$ moves. That is, we must route from $(\pi_1 \pi_n \pi_2 \pi_{n-1} \dots \pi_{n/2} \pi_{(n/2)+1})$ to $(\pi_{n-1} \pi_n \pi_{n-2} \pi_1 \pi_{n-3} \pi_2 \dots \pi_{n/2} \pi_{(n/2)-1})$. This can be done by exchanging appropriate paired items, such as (π_{n-1}, π_1) , (π_{n-2}, π_2) , ..., where the paired items are at distance 3 from each other in π . So, the exchange takes $5n - 10$ moves.

Theorem 6 M_3 embeds P_n into SEP_n with dilation $(3/8)n^2$.

Sketch of Proof: By examining the distance of the elements to be exchanged, one deduces that the prefix reversal of size $(3/4)n$ creates the maximum distance between images, because the individual elements to be exchanged are then the furthest distance from each other. We measure the distance from $(\pi_1 \pi_n \pi_2 \pi_{n-1} \dots \pi_{n/2} \pi_{(n/2)+1})$ to $(\pi_{(3/4)n} \pi_n \pi_{((3/4)n-1)} \pi_{n-1} \dots \pi_{((3/4)n+1)} \pi_{n/2} \pi_1 \pi_{(n/2)-1} \pi_2 \dots \pi_{n/4})$. We observe that the relative order of π_1 through $\pi_{n/4}$ does not have to be changed, and that the same is true for $\pi_{(n/2)+1}$ through $\pi_{(3/4)n}$, and $\pi_{((3/4)n+1)}$ through π_n . Only $\pi_{(n/4)+1}$ through $\pi_{n/2}$ has to have its order reversed. Our strategy is 1) to leave π_1 through $\pi_{n/4}$ in their current position, 2) move $\pi_{(n/4)+1}$ through $\pi_{(3/8)n}$ to the left, 3) move $\pi_{((3/8)n+1)}$ through $\pi_{n/2}$ to the right, 4) leave $\pi_{(3/4)n}$ through $\pi_{(n/2)+1}$ in their current position, and 5) move $\pi_{((3/4)n+1)}$ through π_n to the right. It will cost approximately $n^2/4$ moves to put $\pi_{((3/4)n+1)}$ through π_n in position, and approximately $n^2/8$ moves to put $\pi_{(n/4)+1}$ through $\pi_{n/2}$ in position, for a total of $(3/8)n^2$ moves.

Theorem 6 represents an improvement over another result in [5], advertised as optimal, namely an embedding with dilation $(3/2)n^2 - (7/2)n + 2$. In fact, such is almost three times the diameter of SEP_n , hence, random placements of pancake network nodes is equally good.

5. Future Research

Exact values for the diameter of SEP_n are not yet known. None of our embeddings are provably optimum; good lower bounds are desired. It is also not yet clear how to minimize the congestion of these embeddings.

References

- [1] S. B. Akers and B. Krishnamurthy. A group-theoretic model for symmetric interconnection networks. *IEEE Transactions on Computers*, 38(4):555-566, April 1989.
- [2] S. B. Akers and B. Krishnamurthy. The star graph: an attractive alternative to n-cube. In *Proceedings of International Conference on Parallel Processing (ICPP-87)*, pages 393-400, St. Charles, Illinois, August 1987.
- [3] F. Annexstein, M. Baumslag and A. L. Rosenberg. Group action graphs and parallel architectures. *SIAM Journal on Computing*, 19(3):544-569, June 1990.
- [4] W. H. Gates and C. H. Papadimitriou. Bounds for sorting by prefix reversal. *Discrete Math.*, vol. 27, pages 47-57, 1979.

[5] S. Latifi and P. K. Srimani. A new fixed degree regular network for parallel processing. In *Proceedings of the Eighth IEEE Symposium on Parallel and Distributed Processing*, pages 152-159, Los Alamitos, California, October 1996.

[6] F. T. Leighton. *Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes*. Morgan Kaufmann Publishers, San Mateo, California, 1992.

[7] L. Morales and I. H. Sudborough. Comparing star and pancake networks. In *Proceedings of the Eighth IEEE Symposium on Parallel and Distributed Processing*, pages 152-159, Los Alamitos, California, October 1996.

[8] M. A. Weiss. *Data Structures and Algorithm Analysis*. Benjamin/Cummings, Redwood City, California, 1995.