Cyclic-Cubes: A New Family of Interconnection Networks of Even Fixed-Degrees

Ada Wai-chee Fu
Department of Computer Science and Engineering
The Chinese University of Hong Kong, Shatin, Hong Kong
Email: adafu@cse.cuhk.edu.hk

Siu-Cheung Chau
Department of Mathematics and Computer Science,
University of Lethbridge, Lethbridge, Alberta, Canada, T1K 3M4
Email: chau@cs.uleth.ca
Fax : 403-382-7108

Abstract

We introduce a new family of interconnection networks that are Cayley graphs with fixed degrees of any even number greater than or equal to 4. We call the proposed graphs cyclic-cubes because contracting some cycles in such a graph results in a generalized hypercube. These Cayley graphs have optimal fault tolerance and logarithmic diameters. For comparable number of nodes, a cyclic-cube can have a diameter smaller than previously known fixed-degree networks. The proposed graphs can adopt an optimum routing algorithm known for one of its sub-families of Cayley graphs. We also show that a graph in the new family has a Hamiltonian cycle, and hence there is an embedding of a ring. Embedding of meshes and hypercubes are also discussed.

keywords: Cayley Graphs, generalized hypercube, fixed degree, interconnection network, fault tolerance, diameter, Hamiltonian cycle, graph embedding.

1 Introduction

Desirable properties of an interconnection network include low degree, low diameter, symmetry, low congestion, high connectivity, high fault tolerance, and efficient routing algorithms. Cayley graphs (or group graphs) [1, 2, 3] have been studied for interconnection networks because they possess many of the above properties.

However, a lot of Cayley graphs have vertex degrees that increase with the numbers of vertices in the graph logarithmically or sublogarithmically. They include the hypercubes, star graphs and pancake graphs [2], and the folded Peterson cube networks in [4]. For network construction, we may only be able to support a fixed number of I/O ports at each node, hence we are interested in fixed-degree networks.

Some fixed-degree Cayley graphs are known, e.g. n-cycles in [1], the *connected cycles* in [5], the Cube-Connected Cycles in [6], the Cayley graphs proposed in [7], and the k-ary n-cube which has been used in the design of a number of machines [8, 9, 10, 11, 12]. The k-ary n-cube graphs have even fixed degrees and the other graphs mentioned above have fixed degrees at most 4. In this paper, we propose a new family of Cayley graphs with fixed-degree of any even number greater than or equal to 4. Since for each graph in this new family, after contracting some cycles, we get a generalized hypercube, and the contracted cycles form the vertices of the hypercube, we call our proposed graphs **cyclic-cubes**.

Compared with previous fixed-degree graphs proposed for interconnection network, the new family of graphs can accommodate a much larger number of vertices for a given diameter. For comparison, the k-ary n-cube [13] has a diameter of $n\lfloor \frac{k}{2} \rfloor$, and contains k^n vertices; while G_n^k has a diameter of $\lfloor \frac{3}{2}n \rfloor$, and contains nk^n vertices. For example with diameter 6 and fixed degree of 6, the number of vertices for G_4^3 is 324, while the number of vertices for 5-ary 3-cube, with diameter of 6 and fixed degree of 6, is only 125. With diameter 9 and fixed degree of 6, the number of vertices for G_6^3 is 4374, while that of G_6^2, the graph proposed in [7], with the same diameter and fixed degree of 4, is 384, and the number of vertices for 7-ary 3-cube, with diameter of 9 and fixed degree of 6, is only 343. Some properties of G_n^k are shown in Figure 1.

The comparison of G_n^k with some other fixed degree networks such as enhanced hypercubes [14] and folded hypercubes [15] is shown in Figure 2. For the enhanced hypercube, the value of k in the table can vary from 1 to n. We can see that G_n^k is the only network in the table in which we can fix either the diameter or the connectivity and then vary the number of vertices. Therefore, for a fixed diameter, we can increase the value of k, to increase the number of vertices. For a fixed connectivity, we can increase the value of n to increase the number of vertices.

Figure 2 also shows the parameters for the De Bruijn graph. The De Bruijn graphs have *almost* constant vertex degrees and was used for designing a 8,096 node multiprocessor at JPL for the Galileo project [16]. A De Bruijn graph has k^n vertices with diameter n and most of the vertices with degree $2k$. It can hold more vertices than the cyclic-cube for a fixed diameter. The Binary De Bruijn graph used in the Galileo project has 2^k vertices with diameter k and most of the node degrees of 4. Hence for a diameter of 10, it has 1024 vertices, compared with 896 vertices for G_7^2. However, the De Bruijn graphs do not have constant degree, most of the vertices in a Binary De Bruijn graph have degree 4 but some have only degree 2. This means that the vertex fault tolerance is minimal in the sense that the graphs cannot tolerate more than one faulty vertex.

We define *vertex (edge) connectivity* to be the smallest number of vertices (edges) whose deletion results in a disconnected subgraph. *Fault tolerant* is defined to be the maximum number j so that if any j vertices are deleted from the graph, the resulting subgraph is still connected. Hence fault tolerance is one less than the vertex connectivity. We show that a cyclic-cube has optimal fault tolerance, that is, it has a degree of d and it is still a connected graph after any $d-1$ vertices have failed. It has a diameter logarithmic in the number of vertices and the logarithmic base can be higher than that in previously known cayley graphs. Since the proposed family includes some previously known cayley graphs, it may be possible to adopt some previously known results. We show that some known routing algorithms can easily be generalized and be used with the new family. Furthermore, we show that each cyclic-cube has a Hamiltonian cycle, meaning that a ring can be embedded. Embedding of meshes and hypercubes are also considered.

The paper is organized as follows: in Section 2, we give the definition and some examples of the proposed graph family. Section 3 derives the optimal fault tolerance property. Section 4 is about the diameter and the routing algorithms. The graph embedding properties are discussed in Section 5. Section 6 is a conclusion.

2 Cayley Graphs of Even Degrees

In this section, we define the family of Cayley graphs that we propose. First we give the definition of a Cayley graph.

Definition 1 *[17] Let FG be a finite group and let $\Omega \subset FG$ satisfy $I \notin \Omega$, where I denotes the identity element of FG, and $h \in \Omega$ implies $h^{-1} \in \Omega$. The* **Cayley graph** *$G(FG, \Omega)$ is the graph whose vertices are the elements of FG and for each $g \in FG$, an edge joining g to $g \cdot h$ for all $h \in \Omega$, where $\cdot$ is the group operation. Ω is called the* **generator** *set.*

k	n	# vertices (nk^n)	# edges (nk^{n+1})	diameter ($\lfloor \frac{3n}{2} \rfloor$)	connectivity ($2k$)
2	3	24	48	4	4
2	4	64	128	6	4
2	5	160	320	7	4
2	6	384	768	9	4
2	7	896	1792	10	4
2	8	2048	4096	12	4
3	3	81	243	4	6
3	4	324	1296	6	6
3	5	1215	3645	7	6
3	6	4374	13122	9	6
4	3	192	768	4	8
4	4	1024	4096	6	8
5	3	375	1875	4	10
5	4	2500	12500	6	10
6	3	648	3888	4	12
6	4	5184	31104	6	12
7	3	1029	7203	4	14
8	3	1536	12288	4	16

Figure 1: Some properties of G_n^k

	# vertices	# edges	diameter	connectivity (degree)
n-dimensional binary cube	2^n	n2^{n-1}	n	n
enhanced hypercube n-cube	2^n	$(n+1)2^{n-1}$	$k + \lceil \frac{n-k}{2} \rceil$	$n+1$
Folded n-cube	2^n	$(n+1)2^{n-1}$	$\lceil \frac{n}{2} \rceil$	$n+1$
k-ary n-cube	k^n	nk^n	$n\lfloor \frac{k}{2} \rfloor$	$2n$
De-bruijn graph	k^n	$\approx k^{n+1}$	n	mostly $2k$
G_n^k	nk^n	nk^{n+1}	$\lfloor \frac{3n}{2} \rfloor$	$2k$

Figure 2: Comparison with some other networks

To introduce our family of Cayley graph, we first define a finite group with the generator set. We have n **symbols** $t_1, t_2, ..., t_n$, and k **ranks** $1, 2, ..., k$. We give our symbols an ordering of $t_1 > t_2 > t_3 > ... > t_n$. A rank is also assigned to each of the symbols and each symbol becomes a **ranked symbol**. (We use t_j^i to indicate the result of assigning rank i to symbol t_j.) The resulting ordered set of n ranked symbols is called a **ranking**. If $n \leq 26$, we use English alphabets as symbols; thus for $n = 3, k = 3$, we have $t_1^1 = a^1, t_1^2 = a^2, t_1^3 = a^3$, $t_2^1 = b^1, t_2^2 = b^2, t_2^3 = b^3$, $t_3^1 = c^1, t_3^2 = c^2, t_3^3 = c^3$.

We define a finite group FG such that each element in the group is represented by a circular permutation of a ranking in lexicographic order. For example for $n = 3, k = 3$, $t_2^1 t_3^3 t_1^2$ is an element, but $t_1^1 t_3^2 t_2^3$ is not an element in FG. For n distinct ranked symbols, there are exactly n different circular permutations of the ranked symbols in lexicographic order. Since each symbol has k possible ranks, we have a total of nk^n elements in FG.

The set of generators is given by $\Omega = \{t_2^1 t_3^1 ... t_n^1 t_1^1,\ t_2^1 t_3^1 ... t_n^1 t_1^2,\ t_2^1 t_3^1 ... t_n^1 t_1^3,\ ...,\ t_2^1 t_3^1 ... t_n^1 t_1^k,\ t_n^1 t_1^1 ... t_{n-1}^1,$ $t_n^2 t_1^1 ... t_{n-1}^1,\ ...,\ t_n^k t_1^1 ... t_{n-1}^1\}$. The group operation is defined by:

$$\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n} \cdot t_2^1 t_3^1 ... t_n^1 t_1^i = \alpha_2^{j_2} \alpha_3^{j_3} ... \alpha_n^{j_n} \alpha_1^{j_1 + i - 1}$$
$$\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n} \cdot t_n^i t_1^1 ... t_{n-1}^1 = \alpha_n^{j_n - (i-1)} \alpha_1^{j_1} ... \alpha_{n-1}^{j_{n-1}}$$

(We assume throughout the paper that the superscript (rank) of a symbol is taken modulo k.) The Cayley graph we introduce is $G(FG, \Omega)$, we also refer to such a graph by G_n^k. As described in the previous section G_n^k is also called a *cyclic-cube*. The identity element in FG is $t_1^1 t_2^1 ... t_n^1$. It is also called the **identity vertex** of G_n^k. We may also define the edges of G_n^k by the following **generator functions** in the graph:

$$\begin{aligned}
g(\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n}) &= \alpha_2^{j_2} \alpha_3^{j_3} ... \alpha_n^{j_n} \alpha_1^{j_1} \\
g^{-1}(\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n}) &= \alpha_n^{j_n} \alpha_1^{j_1} ... \alpha_{n-1}^{j_{n-1}} \\
f^i(\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n}) &= \alpha_2^{j_2} \alpha_3^{j_3} ... \alpha_n^{j_n} \alpha_1^{j_1 + i} \\
f^{-i}(\alpha_1^{j_1} \alpha_2^{j_2} ... \alpha_n^{j_n}) &= \alpha_n^{j_n - i + k} \alpha_1^{j_1} ... \alpha_{n-1}^{j_{n-1}}
\end{aligned}$$

$$\begin{aligned}
\text{where} \qquad i &= 1, 2, ..., k-1 \\
\text{and} \qquad \alpha_j^0 &= \alpha_j^k \quad \text{for } 1 \leq j \leq n.
\end{aligned}$$

In G_n^k, there is an edge between u and each of $g(u), g^{-1}(u),\ f^1(u), ..., f^{k-1}(u),\ f^{-1}, ..., f^{-(k-1)}$, for each vertex u. It is easy to see that Ω corresponds to the functions of $\{g, f^1, ..., f^{k-1}, g^{-1}, f^{-1}, ..., f^{-(k-1)}\}$.

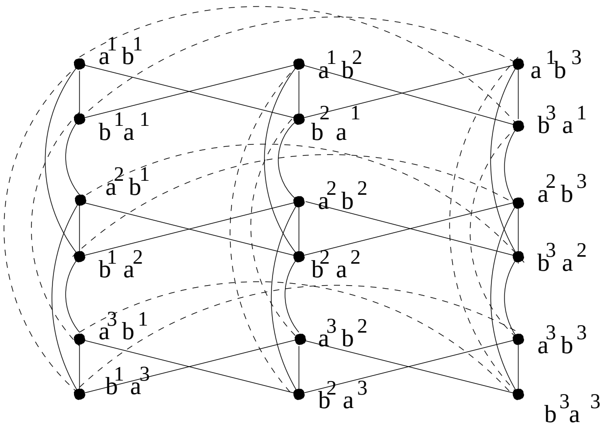

Figure 3: Graph G_2^3

Ω is closed under inverse; in particular, g^{-1} is the inverse of g, f^{-i} is the inverse of f^i. That is, if $g(u) = v$, then $g^{-1}(v) = u$, if $f^i(u) = v$, then $f^{-i}(v) = u$. Therefore the edges in G_n^k are bidirectional.

Theorem 1 *For any $n, n \geq 3$, for any $k, k \geq 2$, the graph G_n^k*

1. *is a symmetric (undirected) regular graph of degree $2k$;*
2. *has $n(k^n)$ vertices; and*
3. *has $n(k^{n+1})$ edges.*

Example: For $n = 2, k = 3$, the vertex set of G_2^3 is given by

$\{a^1b^1, a^1b^2, a^1b^3, a^2b^1, a^2b^2, a^2b^3, a^3b^1, a^3b^2, a^3b^3, b^1a^1, b^1a^2, b^1a^3, b^2a^1, b^2a^2, b^2a^3, b^3a^1, b^3a^2, b^3a^3\}$.

G_2^3 and G_3^2 are shown in Figures 3 and 4, respectively.

3 Fault Tolerance of G_n^k

The **fault tolerance** of a graph is the maximum number of vertices which can become faulty without disconnecting the functional vertices of the graph. The **vertex connectivity** of a graph is the size of the smallest set of vertices whose deletion results in a disconnected subgraph or a single vertex. Hence the fault tolerance of a graph is one less than its connectivity. A graph is said to be **regular of degree** d if every vertex has degree d, it is also said to have a **fixed-degree** d. The maximum fault tolerance that we can expect in a regular graph of fixed-degree d is $d - 1$. A regular graph of degree d with fault tolerance $d - 1$ is called an **optimal fault**

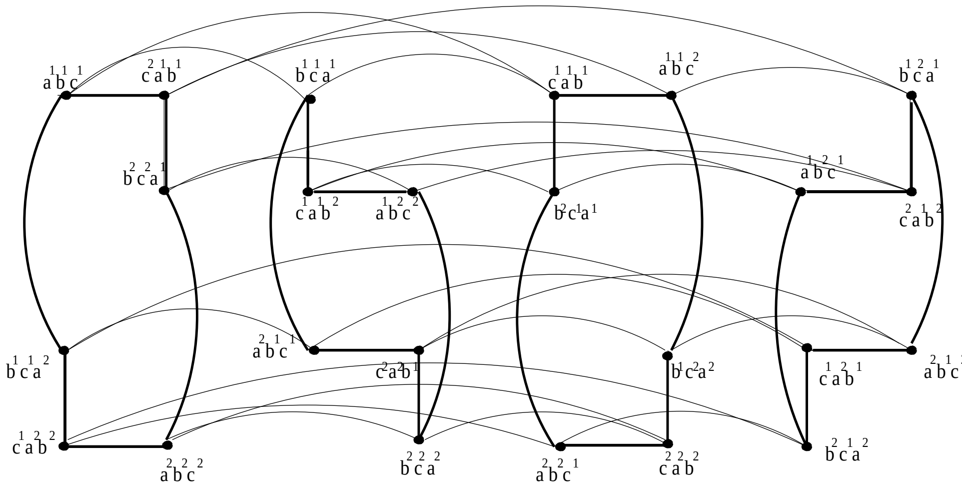

Figure 4: Graph G_3^2

tolerant graph. In this section we shall show that G_n^k is optimal fault tolerant. We shall need the following definitions.

Definition 2 *An edge induced by the symmetric functions g or g^{-1} is called a g-***edge***. If $\{u, v\}$ is a g-edge, then u is a g-***neighbour*** of v and vice versa. Any cycle in G_n^k consisting of only g-edges is called a g-***cycle***. An edge induced by the symmetric functions f^i or f^{-i}, for $1 \leq i \leq k-1$, is called an f-***edge***.*

Definition 3 *For each g-cycle in G_n^k, the unique vertex with symbol starting with t_1^j, for some j, $1 \leq j \leq k$, is called the* **leader** *vertex (or simply the leader).*

Example: In Figure 2, $\{a^1b^1c^1, b^1c^1a^1, c^1a^1b^1, a^1b^1c^1\}$ is a g-cycle. $a^1b^1c^1$ is the leader vertex in this g-cycle.

Definition 4 *Two vertices u and v in G_n^k are g-***adjacent*** if $g(u) = v$ or $g(v) = u$. Two vertices u and v in G_n^k are f-***adjacent*** if $f(u) = v$ or $f(v) = u$. Two g-cycles are f-adjacent if a vertex in one cycle is f-adjacent to a vertex in the other cycle. A vertex u is f-adjacent to a g-cycle if it is f-adjacent to a vertex in the g-cycle.*

Example: In Figure 2, $a^1b^1c^1$ is g-adjacent to $b^1c^1a^1$ and f-adjacent to $b^1c^1a^2$. The g-cycle $\{a^1b^1c^1, b^1c^1a^1, c^1a^1b^1, a^1b^1c^1\}$ is f-adjacent to the g-cycle $\{a^2b^1c^1, b^1c^1a^2, c^1a^2b^1, a^2b^1c^1\}$.

Lemma 1 *The $n(k^n)$ vertices of G_n^k are partitioned into k^n vertex disjoint g-cycles of length n.*

Proof: For any vertex $u = \alpha_1^{j_1}\alpha_2^{j_2}...\alpha_n^{j_n}$, a g-cycle is obtained by starting with u and applying the function g repeatedly. This can be seen as rotating the ranked symbols of u. Hence each g-cycle has length n. Since $g(u) = v \iff g^{-1}(v) = u$, the graph is bidirectional, and we can describe all the edges in the cycles with only the g functions. If the cycles are not vertex disjoint it means that there is at least one vertex that appears in at least 2 g-cycles. Therefore, there exists a vertex u and 2 different vertices v and w such that $g(v) = u$ and $g(w) = u$. We note that $g(w) = g(v) \iff w = v$, which is a contradiction. Hence, the g-cycles must be vertex disjoint, and the $n(k^n)$ vertices of G_n^k are partitioned into k^n vertex disjoint g-cycles of length n. □

Example: In Figure 2, there are 8 g-cycles, the leader vertices of these cycles are $a^1b^1c^1$, $a^1b^1c^2$, $a^1b^2c^1$, $a^1b^2c^2$, $a^2b^1c^1$, $a^2b^1c^2$, $a^2b^2c^1$, $a^2b^2c^2$.

The proofs of the following three lemmas are given in the Appendix. Lemma 2 says that a g-cycle with a leader vertex $u = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$, is f-adjacent to any g-cycle having a leader vertex in which one and only one symbol has a rank different from that of u.

Lemma 2 *A g-cycle C_1 that contains vertex $u = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$, in G_n^k is f-adjacent to the g-cycle C_2 that contains $v = \alpha_1^{p_1}...\alpha_{j-1}^{p_{j-1}}\alpha_j^{q_j}\alpha_{j+1}^{p_{j+1}}...\alpha_n^{p_n}$, where for some j, $1 \leq j \leq n$, $q_j \neq p_j$ The g-cycles such as C_2 are all the g-cycles f-adjacent to C_1.*

Example: In Figure 2, the g-cycle that contains vertex $a^1b^1c^1$ is f-adjacent to the 3 g-cycles $a^2b^1c^1, a^1b^2c^1, a^1b^1c^2$.

Lemma 3 *Each g-cycle in G_n^k is f-adjacent to $(k-1)n$ different g-cycles.*

Example: In Figure 2, $k = 2, n = 3$, $(k-1)n = 3$, each g-cycle is f-adjacent to 3 other g-cycles. In Figure 1, $k = 3, n = 2$, $(k-1)n = 4$, each g-cycle (which is a single edge in this case) is f-adjacent to 4 other g-cycles, e.g. g-cycle $\{a^1b^1, b^1a^1, a^1b^1\}$ is f-adjacent to g-cycles $\{a^1b^2, b^2a^1, a^1b^2\}$, $\{a^2b^1, b^1a^2, a^2b^1\}$, $\{a^1b^3, b^3a^1, a^1b^3\}$ and $\{a^3b^1, b^1a^3, a^3b^1\}$.

Lemma 4 *Each vertex u in a g-cycle in G_n^k is f-adjacent to $2(k-1)$ different g-cycles.*

Note that although each vertex in a g-cycle C_1 is f-adjacent to $2(k-1)$ different other g-cycles and there are n vertices in C_1, C_1 is f-adjacent to only $n(k-1)$ different other g-cycles, by Lemma

3. This is because the g-cycles that are f-adjacent to two vertices in C_1 may not be distinct. In particular, we have the following remark:

Remark 1 *If $u = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ is in C_1, and $v = \alpha_2^{p_2}...\alpha_n^{p_n}\alpha_1^{(p_1+m)}$, then u is f-adjacent to a cycle C_2 that contains v. Also, $u' = \alpha_2^{p_2}...\alpha_n^{p_n}\alpha_1^{p_1}$, which is in C_1, is f-adjacent to $v' = \alpha_1^{(p_1+m)}\alpha_2^{p_2}...\alpha_n^{p_n}$ $= f^{-(k-m)}(u')$, which is in C_2. Hence C_1 is f-adjacent to two vertices in C_2.*

Example: In Figure 2, the g-cycle $\{a^1b^1c^1, b^1c^1a^1, c^1a^1b^1, a^1b^1c^1\}$ is f-adjacent to both vertex $c^1a^1b^2$ and vertex $b^2a^1b^1$ in the g-cycle $\{c^1a^1b^2, b^2c^1a^1, a^1b^2c^1, c^1a^1b^2\}$.

Lemma 5 *If $n \geq 3$ then $2k$ distinct other g-cycles are f-adjacent to a vertex u or its 2 g-neighbours in G_n^k.*

Proof: Let $u = \alpha_1^{j_1}\alpha_2^{j_2}...\alpha_n^{j_n}$ be in g-cycle C. Its 2 g-neighbours are $u_1 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{j_1}$ and $u_2 = \alpha_n^{j_n}\alpha_1^{j_1}...\alpha_{n-1}^{j_{n-1}}$. Vertex u is f-adjacent to
$v_1 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+1)}$, $v_2 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+2)}$, ..., $v_{k-1} = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+k-1)}$,
$v_{-1} = \alpha_n^{(j_n-1)}\alpha_1^{j_1}\alpha_2^{j_2}...\alpha_{n-1}^{j_{n-1}}$, $v_{-2} = \alpha_n^{(j_n-2)}\alpha_1^{j_1}\alpha_2^{j_2}...\alpha_{n-1}^{j_{n-1}}$, ..., $v_{-(k-1)} = \alpha_n^{(j_n-k+1)}\alpha_1^{j_1}\alpha_2^{j_2}...\ \alpha_{n-1}^{j_{n-1}}$.
Vertex u_1 is f-adjacent to $v_k = \alpha_3^{j_3}...\alpha_n^{j_n}\alpha_1^{j_1}\alpha_2^{(j_2+1)}$ and u_2 is f-adjacent to $v_{-k} = \alpha_{n-1}^{(j_{n-1}-k+1)}\alpha_n^{j_n}\alpha_1^{j_1}$ ... $\alpha_{n-2}^{j^{n-2}}$. It is easy to check that for $n \geq 3$, the g-cycles that contain $v_1, v_2, ..., v_k, v_{-1}, ..., v_{-k}$ are distinct. $\square$

Definition 5 *For a given G_n^k, we construct a reduced graph SG_n^k as follows: contract each g-cycle C into a single vertex x and label this vertex by extracting the ordered ranks of the ranked symbols in the label of the leader vertex in the g-cycle. We say that C* **corresponds** *to x and vice versa. An edge in SG_n^k exists between 2 vertices if and only if there is one or more edges between the 2 corresponding g-cycles in G_n^k.*

From Lemma 1, after contracting all the g-cycles, no vertex in G_n^k will remain. From Lemma 3, each g-cycle is f-adjacent to $(k-1)n$ distinct other g-cycles, hence each vertex in SG_n^k is adjacent to $(k-1)n$ distinct other vertices. From Lemma 2, SG_n^k is a generalized hypercube [18] of k^n vertices, which is a product graph [19] of n complete graphs of k vertices each. Since in G_n^k, each vertex of this hypercube is replaced by a g-cycle, we therefore call G_n^k a *cyclic-cube*. Hence we have the following remark.

Remark 2 *Each vertex in SG_n^k has a label of length n and has a degree of $(k-1)n$. Thus, SG_n^k is a $(k-1)n$-regular undirected graph.*

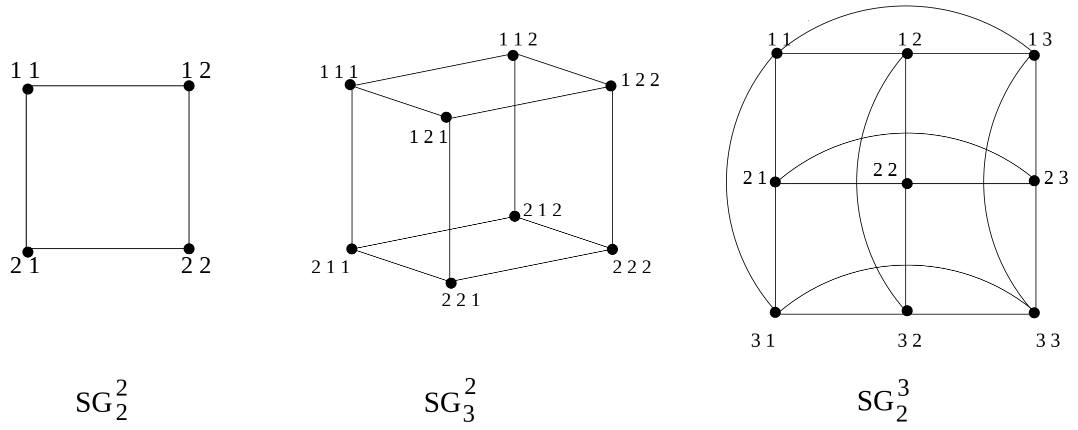

Figure 5: Some SG_n^k graphs

Figure 3 shows some SG_n^k graphs. If A and B are two n-digit radix k vectors, the **Hamming distance** between A and B is given by the number of positions in which A and B differ. From Lemma 2, the Hamming distance between two vertices in SG_n^k is 1 if and only if the two vertices are adjacent in SG_n^k.

Theorem 2 *For $n \geq 2, k \geq 2$, the vertex connectivity of SG_n^k is $(k-1)n$.*

Proof: Consider two arbitrary vertices $u = u_0u_1...u_{n-1}$ and $v = v_0v_1...v_{n-1}$ in SG_n^k. We construct a set of $(k-1)n$ vertex disjoint paths between u and v as follows: Let the set of ranked symbols in u that are different from v be $D = \{u_{i_1}, u_{i_2}, ...u_{i_d}\}$, where $0 \leq d < n$.

We shall proceed in the following steps:

1. The first vertex disjoint path is the path that passes through vertices $z_1, ..., z_d$ in SG_n^k, where $z_1 = u$, and z_m is obtained by modifying the i_m-th symbol of the label of z_{m-1} changing it from rank u_{i_m} to v_{i_m}.
2. The next $(d-1)$ paths are obtained by repeating the ranking modification in Step (1) except the order of modification is rotated. (ie. the second path is in the order of $i_2, i_3, ...i_d, i_1$, the third path is in the order of $i_3, i_4, ...i_d, i_1, i_2$, etc.)
3. The next set of $(k-2)$ paths are obtained by first modifying the rank of u_{i_1} to be a rank that is not v_{i_1}, there are $(k-2)$ choices, then repeat the sequence of rank modification as in Step (1). Finally the rank of the (i_1)-th symbol is changed to that of v_{i_1}.

The next set of $(d-1)(k-2)$ paths are obtained by repeating the process of Step (3) using $u_{i-2}, u_{i_3}, ...u_{i_d}$ instead of v_{i_1}.

So far we have constructed $d(k-1)$ vertex disjoint paths.

Next we construct the remaining $(n-d)(k-1)$ vertex disjoint paths. Let the set of ranked symbols in u that are the same as v be $S = \{u_{j_1}, u_{j_2}, ..., u_{j_{n-d}}\}$. For these $(n-d)(k-1)$ paths, the first $(k-1)$ paths are constructed by first modifying the rank of the symbol of u at u_{j_1}, there are $k-1$ choices of ranks for modification. Next proceed as in Step (1), and finally the rank of the j_1-th symbol is changed back to u_{j_1}.

The next $(n-d-1)(k-1)$ paths are constructed in the same way by using $u_{j_2}, u_{j_3}, ..., u_{j_{n-d}}$ instead of u_{j_1}.

Totally, we have $(k-1)n$ vertex disjoint paths. $\square$

We shall apply the following theorem in [20] which also appears in [19]. It is a generalization of Menger's theorem [21].

Theorem 5.14 of [19]: A graph with at least $2m$ points is m-connected if and only if for any two disjoint sets V_1 and V_2 of m points each, there exist m disjoint paths joining these two sets of points. $\square$

Lemma 6 *Suppose $n \geq 3$, $k \geq 2$. Consider two arbitrary vertices u and v in G_n^k such that u and v belong to different g-cycles. Then there exist $2k$ vertex disjoint paths between u and v.*

Proof: From Lemma 5, if $n \geq 3$ then $2k$ distinct other g-cycles are f-adjacent to a vertex u or its 2 g-neighbours in G_n^k, let these g-cycles be $C_1, C_2, ...C_{2k}$. Similarly $2k$ distinct other g-cycles are f-adjacent to another vertex v or its 2 g-neighbours in G_n^k, let these g-cycles be $C'_1, C'_2, ..., C'_{2k}$. Note that $C_i = C'_j$ is possible for some i, j.

By Theorem 5.14 of [19] cited above, given two sets of vertices V_1 and V_2 such that $|V_1| = |V_2| = 2k$ in a $2k$-connected graph, if there is a set of p vertices, V_3, appearing in both V_1 and V_2, then there are $2k-p$ vertex disjoint paths connecting the vertices from $V_1 - V_3$ to those in $V_2 - V_3$. Then there will be $2k$ vertex disjoint paths from V_1 to V_2, if we consider going from a vertex x to itself is a path.

From Theorem 2, SG_n^k is $n(k-1)$ connected. Suppose c_i and c_i' are the vertices in SG_n^k that correspond to C_i and C_i' in G_n^k, respectively. Let $V_1 = \{c_1, ..., c_{2k}\}$ and $V_2 = \{c_1', ...c_{2k}'\}$ in the above argument. First assume that V_1 and V_2 are disjoint. If $n \geq 4$ and $k \geq 2$, then $n(k-1) \geq 4(k-1) \geq 2k + (2k-4) \geq 2k$. Therefore SG_n^k is at least $2k$ connected. Hence we have at least $2k$ vertex disjoint paths connecting u and v. Next, we prove the statement is also true if V_1 and V_2 are not disjoint. We allow a subset V_3 of size p to appear in both V_1 and V_2. Then the vertex sets $V_1 - V_3$ and $V_2 - V_3$ each has size of $2k - p$. The graph SG_n^k is at least $(2k-p)$-connected and there are at least $2k-p$ vertex disjoint paths between $V_1 - V_3$ and $V_2 - V_3$. Hence there are $2k - p$ vertex disjoint paths between u and v. The remaining p vertex disjoint paths between u and v are through the vertices that correspond to V_3.

For $n = 3$, $n(k-1) = 3(k-1) = 2k + (k-3)$, which is at least $2k$ if $k \geq 3$. Hence for $n = 3$ and $k \geq 3$, SG_n^k is at least $2k$ connected.

The remaining case is G_3^2, which belongs to the graph family of G_n^2 introduced in [7], in which it is shown to have a vertex connectivity of 4. □

Lemma 7 *Suppose $n \geq 3$, $k \geq 2$. Consider two arbitrary vertices u and v in G_n^k such that u and v belong to the same g-cycles. Then there exist $2k$ vertex disjoint paths between u and v.*

Proof: By Lemma 4 and argument similar to that in the proof of Lemma 6, there are $2(k-1)$ vertex disjoint paths through the g-cycles which are f-adjacent to u. The remaining 2 paths are within the same g-cycle. □

Theorem 3 *The graph G_n^k is $2k$-vertex-connected for any integers $n, k, n \geq 3, k \geq 2$.*

Proof: The theorem follows from the previous two lemmas. □

4 Optimal Routing and Diameter of G_n^k

In this section, we shall adopt the routing algorithms that has been develop for the sub-family of graphs G_n^2 in [7]. We show that the algorithms can be generalized to the cases of G_n^k. The routing problem we consider is the computation of a path between any 2 given vertices in the graph. An **optimal routing scheme** is one that finds the shortest path between any 2 vertices.

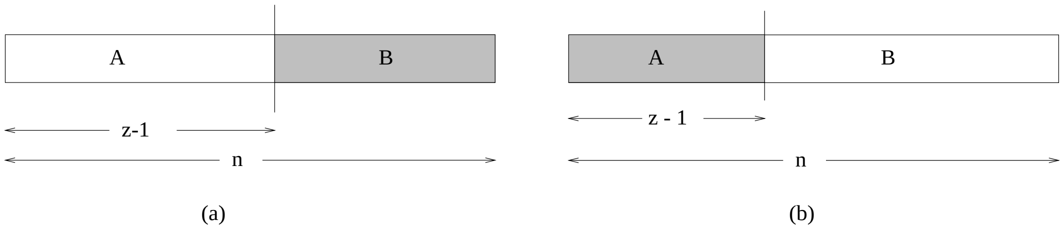

Figure 6: Illustration for the simple routing algorithm

Since G_n^k is a Cayley graph, it is vertex symmetric [2]. The problem is to find the distance d between any source vertex s and a destination vertex. The destination vertex can be renamed to become the identity vertex I, and the source vertex s will become s' under the same renaming scheme. The distance between s' and I is then d. To illustrate such a renaming, let $c^3a^1b^2$ be the source vertex and $b^1c^3a^3$ be the destination vertex. We can map the destination vertex to the identity vertex by renaming the symbols as $b^1 \to a^1$, $c^3 \to b^1$, $a^3 \to c^1$. This renaming implies the renaming of $b^2 \to a^2$, $b^3 \to a^3$, $c^1 \to b^2$, $c^2 \to b^3$, $a^1 \to c^2$, $a^2 \to c^3$. Hence the source vertex is renamed as $b^1c^2a^2$. Therefore the routing problem for G_n^k can be reduced to the problem of finding the distance between a source vertex and the identity vertex. In what follows we would only consider the distance between a source vertex $s = \alpha_1^{p_1}\alpha_2^{p_2}....\alpha_n^{p_n}$ and the identity vertex $I = t_1^1t_2^1...t_n^1$.

The first routing algorithm is quite simple and is illustrated in Figure 6. In this diagram, let $\alpha_z^{p_z} = t_1^i$ for some z, where $1 \leq z \leq n$, and $1 \leq i \leq k$. We split $s = \alpha_1^{p_1}\alpha_2^{p_2}....\alpha_n^{p_n}$ into two parts: $A = a_1^{p_1}...a_{z-1}^{p_{z-1}}$ and $B = a_z^{p_z}...a_n^{p_n}$. The algorithm is called Simple_Rout and is shown in Figure 7.

Lemma 8 *For an arbitrary vertex $\alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ in G_n^k, the Algorithm Simple_Rout generates a path of length $\leq \left\lfloor \frac{3n}{2} \right\rfloor$.*

Proof: In Algorithm Simple_Rout, step (2) induces a sub-path of length at most $\lfloor n/2 \rfloor$. Step(3) induces a sub-path of length n. These two sub-paths constitute the entire path. $\square$

Corollary 1 *The upper bound on the diameter of G_n^k is $\left\lfloor \frac{3n}{2} \right\rfloor$.*

Next we present an optimal routing algorithm. The basic idea behind this algorithm is that some sequence of symbols in the source vertex may already match a sequence in the identity vertex. In particular we examine two cases as shown in Figure 8. In this figure, the sequence of

Algorithm Simple_Rout
begin
  (1) Find z such that $\alpha_z^{p_z} = t_1^i$ for $1 \leq i \leq k$.
  (2) **if** $(z-1) > \lfloor n/2 \rfloor$ **then**
    this case is illustrated in Figure 6(a)
    Rotate B in Figure 6 (a) to the left by the g^{-1} function, hence AB becomes BA.
  else
    this case is illustrated in figure 6(b)
    Rotate A in Figure 6 (b) to the right by the g function, hence AB becomes BA.
  (3) Rotate all symbols once by the g or f^i functions, where for each rotation,
  i or g is chosen so that the rotated symbol becomes t_j^1 for some j.
end

Figure 7: A simple routing algorithm

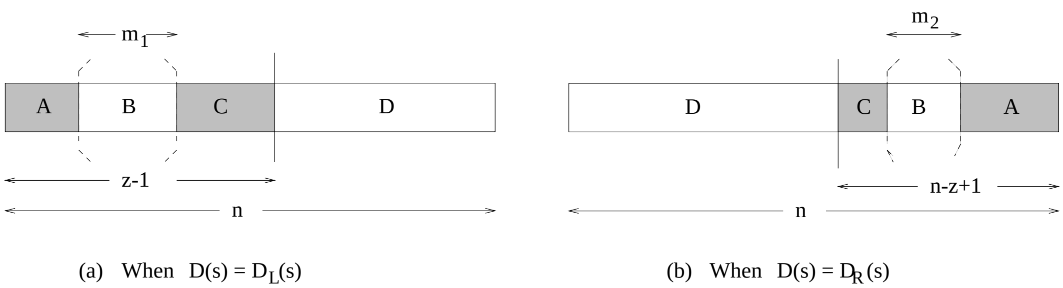

Figure 8: Illustration for the optimum routing algorithm

symbols of the source node is represented by a rectangle, with sub-sequences A, B, C, D or D, C, B, A. In each case, we find the location z such that $a_z^{p_z} = t_1^{p_z}$, so that the sequence is partitioned into two parts of lengths $z-1$ and $n-z+1$.

In case (a) we find a longest sequence B in the first part such that B is equivalent to some portion of the identity vertex I, subroutine $Rout_L$ in Figure 9 can be used to find a path to I. In case (b) we find a longest sequence B in the second part such that B is equivalent to some portion of the identity vertex I, and Subroutine $Rout_R$ in Figure 9 can be used to find a path to I. The lengths of the paths generated by Subroutines $Rout_L$ and $Rout_R$ is given by $D_L(s)$ and $D_R(s)$, respectively, which are defined in the following definition. Intuitively, one sees that in both cases, the symbols in sub-sequences A and C are rotated twice, sub-sequence D is rotated once and sub-sequence B is not rotated. The total number of rotations of the symbols gives the length of the total path.

Subroutine $Rout_L$

Refer to the illustration in Figure 8 (a)

(1) Rotate A to the right by g function so that ABCD becomes BCDA.

(2) Rotate A to the left by some f^{-i} or g functions so that BCDA becomes A'BCD, where A' is the same as A except the ranks of all symbols become 1.

(3) Rotate D and C to the left by some f^{-i} or g^{-1} functions so that A'BCD becomes C'D'A'B. where C'D' is the same as CD except the ranks of all symbols become 1.

(4) Rotate C' to the right by g function so that C'D'A'B becomes D'A'BC'.

Subroutine $Rout_R$

Refer to the illustration in Figure 8 (a)

(1) Rotate A to the left by g^{-1} function so that DCBA becomes ADCB.

(2) Rotate A to the left by some f^{-i} or g^{-1} functions so that ADCB becomes DCBA', where A' is the same as A except the ranks of all symbols become 1.

(3) Rotate D and C to the right by some f^{i} or g functions so that DCBA' becomes BA'D'C'. where D'C' is the same as DC except the ranks of all symbols become 1.

(4) Rotate C' to the left by g^{-1} function so that BA'D'C' becomes C'BA'D'.

Algorithm Opt_Rout(s)
begin
  if $D_L(s) < D_R(s)$ **then** $Rout_L$ **else** $Rout_R$
end

Figure 9: Optimal Routing

Definition 6 *Consider an arbitrary vertex $s = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ in G_n^k. There exists an unique integer z such that $\alpha_z^{p_z} = t_1^{p_z}$. We define* **left distance** *$D_L(s)$ and* **right distance** *$D_R(s)$ of the vertex s from the identity vertex as follows:*

$$\begin{aligned}
D_L(s) &= 2(z - m_1 - 1) + (n - z + 1) \\
D_R(s) &= 2(n - z - m_2 + 1) + (z - 1)
\end{aligned}$$

where

$$\begin{aligned}
m_1 &= \max_m\{\exists(i,j)|(1 \le j < z) \wedge ((j + m - 1) < z) \\
&\quad \wedge \left(\alpha_j^{p_j}\alpha_{j+1}^{p_{j+1}}...\alpha_{j+m-1}^{p_{j+m-1}} = t_i^1 t_{i+1}^1 ... t_{i+m-1}^1\right) \\
m_2 &= \max_m\{\exists(i,j)|(z \le j \le n) \wedge ((j + m - 1) \le n) \\
&\quad \wedge \left(\alpha_j^{p_j}\alpha_{j+1}^{p_{j+1}}...\alpha_{j+m-1}^{p_{j+m-1}} = t_i^1 t_{i+1}^1 ... t_{i+m-1}^1\right)
\end{aligned}$$

Then we define the distance of the vertex s from the identity vertex as

$$D(s) = \min\{D_L(s), D_R(s)\}$$

Lemma 9 *For an arbitrary vertex $s = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ in G_n^k, $D(s') = D(s) \pm 1$, where $s' = \delta(s)$ and $\delta \in \{g, g^{-1}, f^1, ...f^{k-1}, f^{-1}, ...f^{-(k-1)}\}$.*

Proof: We need to consider the following cases:

1. $(\delta = g)$. We have $z' = z - 1$ and there can be four possible changes in m_1 and m_2 in the definition of $D(s)$.
   (a) $m_1' = m_1 - 1$ and $m_2' = m_2 + 1$
   (b) $m_1' = m_1 - 1$ and $m_2' = m_2$
   (c) $m_1' = m_1$ and $m_2' = m_2 + 1$
   (d) $m_1' = m_1$ and $m_2' = m_2$
2. $(\delta = f^i)$ for some i, such that $1 \le i \le (k-1)$. We have $z' = z - 1$ and there can be three possible changes in m_1 and m_2.
   (a) $m_1' = m_1 - 1$ and $m_2' = m_2$
   (b) $m_1' = m_1$ and $m_2' = m_2 + 1$
   (c) $m_1' = m_1$ and $m_2' = m_2$

3. $(\delta = g^{-1})$. We have $z' = z + 1$ and there can be four possible changes in m_1 and m_2.
   (a) $m_1' = m_1 + 1$ and $m_2' = m_2 - 1$
   (b) $m_1' = m_1$ and $m_2' = m_2 - 1$
   (c) $m_1' = m_1 + 1$ and $m_2' = m_2$
   (d) $m_1' = m_1$ and $m_2' = m_2$
4. $(\delta = f^{-i})$ for some i, such that $1 \leq i \leq (k-1)$. We have $z' = z - 1$ and there can be three possible changes in m_1 and m_2.
   (a) $m_1' = m_1$ and $m_2' = m_2 - 1$
   (b) $m_1' = m_1 + 1$ and $m_2' = m_2$
   (c) $m_1' = m_1$ and $m_2' = m_2$

It is straight-forward to verify that in each case, $D(s') = D(s) \pm 1$. □

Theorem 4 *For an arbitrary vertex $s = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ in G_n^k, the function $D(s)$ gives the shortest distance between the vertex s and the identity vertex $I = t_1^1 t_2^1 ... t_n^1$.*

Proof: Let the shortest distance of s from I be d. It is easy to see that Algorithm Opt_Rout constructs a path of length $D(s)$ from s to I. Hence $D(s)$ is an upper bound on the shortest distance of s from I, or $D(s) \geq d$. Suppose we have a shortest path from s to I, let the path be $s, v_1, v_2, ..., v_p, I$. Since v_p is adjacent to I in the graph, $D(v_p)$ is 1. Since v_i is adjacent to v_{i-1}, by Lemma 9, $D(v_{i-1}) = D(v_i) \pm 1$. Therefore the the value of $D(v_{p-1})$ is at most $D(v_p) + 1 = 2$, the value of $D(v_{p-2})$ is at most $D(v_{p-1}) + 1$, which is 3. Similarly, the value of $D(v_1)$ is at most $D(v_2) + 1$, which is p, and finally, the values of $D(s)$ is at most $p + 1 = d$. That is $D(s) \leq d$. We conclude that $D(s) = d$. □

Corollary 2 *Algorithm Opt_Rout is an optimal routing algorithm for vertices in the graph G_n^k.*

Theorem 5 *The diameter of the graph G_n^k is $\left\lfloor \frac{3n}{2} \right\rfloor$.*

Proof: Consider a particular source vertex $s = t^2_{\lfloor n/2 \rfloor + 1} \, ... \, t_n^2 t_1^2 \, ... \, t^2_{\lfloor n/2 \rfloor}$, $D(s) = \left\lfloor \frac{3n}{2} \right\rfloor$. From Corollary 1, the upper bound of the diameter is $\left\lfloor \frac{3n}{2} \right\rfloor$. Hence, the diameter is $\left\lfloor \frac{3n}{2} \right\rfloor$. □

4.1 Adaptive Routing

Adaptive routing [22] makes use of alternative paths between communicating processors, making more efficient use of network bandwidth and providing resilience to failure.

When routing is done concurrently for multiple messages, the routing protocol may wait for the required resources. Deadlock can result from cyclic dependencies. A deadlock-free routing method is an adaptive routing method that avoids deadlocks. Alternatively, an adaptive routing protocol may backtrack and explore other possible paths, or it may choose to abort.

The above proposed routing algorithms for G_n^k networks are not deadlock-free, since the routing will make use of paths in g-cycles and there can be cyclic dependency. Instead we suggest that a backtracking protocol can be used for handling the deadlock problem. From the proof of Theorem 3, we see that there are 2k vertex disjoint paths between any two vertices for G_n^k where $n \geq 3$ and $k \geq 2$. Therefore it is possible to backtrack and try a different paths from among these 2k paths.

5 Graph Embeddings

In this section we consider the graph embedding properties of G_n^k. Graph embedding is important because it allows the application of algorithms designed for other topologies.

Let G and H be two undirected graphs where G is called the **guest graph** and H is called the **host** graph. Let V_G, E_G, V_H, E_H denote the vertex and edge sets of G and H, respectively, and let P_H denote the set of all paths in H. Then an **embedding** of G in H is a one-to-one mapping of their vertices $f_V : V_G \to V_H$ and an edge-mapping $f_E : E_G \to P_H$, where

$$(a,b) \in E_G \Rightarrow f_E(a,b) = (x_1, ..., x_n) \text{ such that } (x_1, ...x_n) \in P_H, x_1 = f_V(a), \text{ and } x_n = f_V(b).$$

Given an embedding (f_V, f_E) of G in H, we have the following definitions: the **dilation** of an embedding is the length of the longest path in H that is associated with an edge in G by f_E. The **congestion** of an embedding is the maximum number of times a single edge of H belongs to paths in H associated with edges in G by f_E. The **expansion** of an embedding is the ratio of the number of nodes in the host graph to the number of nodes in the guest graph.

If an embedding of a graph G into a graph H can be found having dilation and congestion equal to one, then G is isomorphic to a subgraph of H. A recent comparison on graph embedding for different guest graph topologies can be found in [4].

5.1 Embedding a Ring: Hamiltonian property of $\mathbf{G}_n^k$

A **Hamiltonian path** in a graph is a path that includes all vertices in the graph. A **Hamiltonian cycle** in a graph is a Hamiltonian path such that the first vertex and the last vertex in the path are connected by an edge. A study of Hamiltonian properties of Cayley graphs can be found in [23]. In this sub-section, we show that G_n^k has a Hamiltonian cycle and therefore the embedding of a ring.

Definition 7 *An edge induced by the symmetric function f^i or f^{-i} is called an f^i-***edge***. Any cycle in G_n^k consisting of only f^1-edges is called an $f1$-***cycle***.*

Example: In Figure 4, the cycle $\{a^1b^1c^1,\ b^1c^1a^2,\ c^1a^2b^2,\ a^2b^2c^2,\ b^2c^2a^1,\ c^2a^1b^1,\ a^1b^1c^1\ \}$ is an $f1$-cycle.

Lemma 10 *By applying the function f^1 to a vertex v, we can generate an f^1-cycle and each f^1-cycle contains nk distinct vertices.*

Proof:

Consider a vertex $v = \alpha_1^{p_1}...\alpha_n^{p_n}$. If we apply f^1 to v i times where $1 \leq i < n$, we get a vertex $v_i = \alpha_{i+1}^{p_{i+1}}\alpha_{i+2}^{p_{i+2}}...\ \alpha_n^{p_n}\alpha_1^{p_1+1}...\ \alpha_i^{p_i+1}$. The vertices v_i where $1 \leq i < n$ are all distinct.

Suppose we apply f^1 to v j times where $1 \leq j < nk$, $j = qn + i$, $1 \leq i < n$, and $q < k$, we obtain a vertex $v_j = \alpha_{i+1}^{p_{i+1}+q}\alpha_{i+2}^{p_{i+2}+q}...\ \alpha_n^{p_n+q}\alpha_1^{p_1+q+1}...\ \alpha_i^{p_i+q+1}$. The vertices v_j where $1 \leq j < nk$ are all distinct.

Let $j = nk$, $v_j = \alpha_1^{p_1+nk}\alpha_2^{p_2+nk}...\alpha_n^{p_n+nk} = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n} = v$. Hence, we can obtain a cycle of nk distinct vertices by applying the function f^1 nk times. $\square$

Lemma 11 *Each $f1$-cycle contains a unique vertex that starts with t_1^1.*

Proof: Suppose $v = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$ is in an $f1$-cycle, and $\alpha_i = t_1$. We can apply f^1 to v j times such that $j = qn + i - 1$ and $p_i + q = k + 1$. $v_j = \alpha_i^{p_i+q}\alpha_{i+1}^{p_{i+1}+q}...\ \alpha_n^{p_n+q}\alpha_1^{p_1+q+1}...\ \alpha_{i-1}^{p_{i-1}+q+1}$. Hence, $v_j = t_1^1\alpha_{i+1}^{p_{i+1}+q}...\ \alpha_n^{p_n+q}\alpha_1^{p_1+q+1}...\ \alpha_{i-1}^{p_{i-1}+q+1}$. $\square$

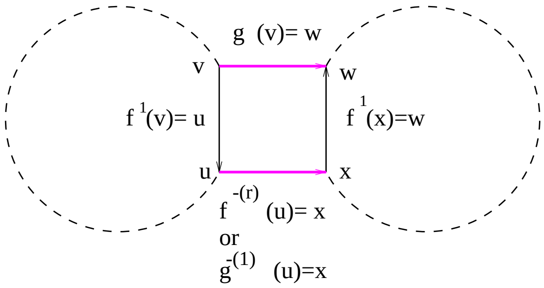

Figure 10: Merging of two $f1$-cycles

Definition 8 *Suppose $k \geq 2, n \geq 2$. For an $f1$-cycle in G_n^k, the unique vertex starting with t_1^1 is called the f-**leader** vertex.*

Lemma 12 *The nk^n vertices of G_n^k are partitioned into k^{n-1} vertex disjoint $f1$-cycles of length kn and the $f1$-cycles contain all the f^1-edges in G_n^k.*

Proof: From Lemma 10, each vertex v in G_n^k must belong to at least one $f1$-cycle because we can always apply the function f^1 to a vertex. From Lemma 11 after some application of the f^1 function to a vertex v, an unique f-leader is reached. Hence, v can only belong to one $f1$-cycle.

From Lemma 10, each $f1$-cycle contains nk distinct vertices. Furthermore, there are k^{n-1} distinct f-leader vertices $t_1^1 t_2^{p_2} ... t_n^{p_n}$ with all combinations of p_2, ..., p_n, where p_i can be any value in $\{2, ..., k\}$. Hence, there are k^{n-1} vertex disjoint $f1$-cycles of length kn in G_n^k. Finally, each vertex can belong to only two different f^1-edges, the $f1$-cycles contain all the f^1-edges in G_n^k. $\square$

Lemma 13 *Let $k \geq 2, n \geq 2$. For any vertex v in G_n^k such that $f^j(v) = u$ and $g(v) = w$, where $1 \leq j \leq k-1$, there exists a vertex x such that $f^i(x) = w$ where $1 \leq i \leq k-1$, and either $g^{-1}(u) = x$ or $f^{-r}(u) = x$, where $r = (i+j) \bmod k$.*

Proof: Consider a vertex $v = \alpha_1^{p_1} \alpha_2^{p_2} ... \alpha_n^{p_n}$ in G_n^k, let $u = f^j(v) = \alpha_2^{p_2} ... \alpha_n^{p_n} \alpha_1^{p_1+j}$, $w = g(v) = \alpha_2^{p_2} ... \alpha_n^{p_n} \alpha_1^{p_1}$ and $x = \alpha_1^{p_1+k-i} \alpha_2^{p_2} ... \alpha_n^{p_n}$. Then $f^i(x) = w$. If $(i+j) \bmod k \neq 0$, then $f^{-(i+j) \bmod k}(u) = \alpha_1^{p_1+j-i-j+k} \alpha_2^{p_2} ... \alpha_n^{p_n}$ and is equal to x. If $(i+j) \bmod k = 0$, then $x = \alpha_1^{p_1+j} \alpha_2^{p_2} ... \alpha_n^{p_n}$ and $g^{-1}(u) = x$. $\square$

Corollary 3 *Suppose $k \geq 2, n \geq 2$. For an arbitrary vertex v in G_n^k such that $f^j(v) = u$ and $g(v) = w$, there exists a vertex x such that $f^1(x) = w$ and either $g^{-1}(u) = x$ or $f^{-(j+1)}(u) = x$.*

In Figure 10, we show the special case where v and w belongs to two $f1$-cycles, so that i and j in Lemma 13 are 1. One can see that the 2 $f1$-cycles can be merged by deleting edges $\{v, u\}$ and $\{w, x\}$ and adding $\{v, w\}$ and $\{u, x\}$.

Lemma 14 *The $f1$-cycles in G_n^k cannot be partitioned into 2 sets S_1 and S_2, such that there is no g-edge joining 2 vertices v_1, v_2, where v_1 is a vertex in an $f1$-cycle in S_1, and v_2 is a vertex in an $f1$-cycle in S_2.*

Proof: Suppose that we can do the partitioning. Let S_1 and S_2 be the two sets of $f1$-cycles. This implies that the vertices in the $f1$-cycles of each of S_1 and S_2 belong to two different sets of g-cycles. Let us contract the g-cycles in the way we form the graph SG_n^k. Similar as before, we label the contracted nodes by the ranking of the leader vertex in the g-cycles. Let the sets of contracted vertices formed from S_1 and S_2 be S_1' and S_2', respectively.

Let us define the distance between 2 symbols x_i and y_i in a label, where $y_i > x_i$, as follows: if $x_i = 1$ and $y_i = k$, then the distance is 1, otherwise, the distance is $y_i - x_i$.

First, we want to show that there must be two contracted vertices, one from S_1' and one from S_2', such that their labels differ only in one symbol and the distance between the differing symbols is one. Suppose, this is not true, we have two possible cases:

1. Every vertex in S_1' differs from every vertex in S_2' by at least two symbols. There exists x in S_1' and y in S_2' where $x = x_1x_2...x_n$, and $y = y_1y_2...y_n$. Without loss of generality, $x_1 \neq y_1$ and $x_2 \neq y_2$ and no vertex $z = x_1y_2....y_n$ is found in S_1' or S_2'. But z must exist in S_1' or S_2' because SG_n^k contains all permutations of the ranks.
2. Or, there exists a vertex in S_1' that differs from a vertex in S_2' in only one symbol, and for such a pair of vertices, and the distance between the the differing symbols is at least 2. There exists x in S_1' and y in S_2' where $x = x_1x_2...x_n$, and $y = y_1y_2...y_n$. Without loss of generality, $x_1 > ((y_1 + 1) \bmod k)$, and no vertex $z = ((y_1 \bmod k) + 1)\ y_2....y_n$ is found in S_1' or S_2'. But z must exist in S_1' or S_2' because SG_n^k contains all permutations of the ranks.

Hence, there exists two vertices x and y, $x \in S_1'$, $y \in S_2'$, such that their labels differ only in one symbol and the distance between the differing symbols is one. It means that there must be an f^1-edge between x and y. This completes the proof. □

In Figure 12, we show a recursive algorithm that keeps merging $f1$-cycles into a big cycle **C**, until all $f1$-cycles are merged. The merging process is illustrated in Figure 11.

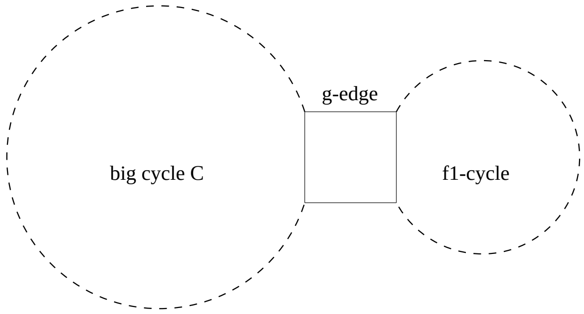

Figure 11: Merging of an $f1$-cycle into $\mathbf{C}$

Algorithm Merge_All_$f1$-cycles (G_n^k)

Begin

(1) Initialize $\mathbf{C}$ to be any $f1$-cycle.
    Let $\mathbf{R}$ be the set of remaining $f1$-cycles.

(2) Repeat step (*) until $\mathbf{R} = \phi$

(*) Find vertices $v \in \mathbf{C}$, $w \in \mathbf{R}$ such that $g(v) = w$.
    Find the vertices u and x as specified in Lemma 13.
    Merge the $f1$-cycle $c1$ that contains w into $\mathbf{C}$
        by deleting the edges $\{v, u\}, \{w, x\}$ and adding the edges $\{v, w\}, \{u, x\}$.
    Remove $c1$ from $\mathbf{R}$.

End

Figure 12: Algorithm to merge all $f1$-cycles in G_n^k

Definition 9 *For the vertices v, u, w, x mentioned in Lemma 13, we call v the first node, u the second node, w the third node, and x the fourth node of the merge.*

Theorem 6 *The algorithm Merge_All_$f1$-cycles finds a Hamiltonian cycle for G_n^k in $O(V^2)$ time, where $V = nk^n$ is the number of vertices in G_n^k.*

Proof: From Lemma 12, G_n^k is partitioned into $f1$-cycles. If all $f1$-cycles can be merged into one big cycle, the algorithm can correctly find a Hamiltonian cycle. That is, if we can find the vertices u, v, x, w in step (*), then from Lemma 13, we can merge the $f1$-cycle $c1$ into the cycle $\mathbf{C}$ by deleting the edges $\{v, u\}$ and $\{w, x\}$ and adding the edges $\{v, w\}$ and $\{u, x\}$.

We need to show that the four nodes for the merge, v, w, u, x can always be found at each iteration until all $f1$-cycles are merged.

From Lemma 14 we can always find a g edge between **C** and **R**. This means that v and w as defined in the lemma can be found.

Suppose $v \in \mathbf{C}$ and v is the first node of the merge. We want to show that the vertex u that follows v, in **C**, must satisfy $f^j(v) = u$ for some $1 \leq j \leq k-1$. There are two cases. For the first case, **C** is an $f1$-cycle, $f^1(v) = u$. For the second case, **C** contains more than one merged $f1$-cycles. The vertex v cannot be involved in previous merges as the first node. If v is involved in previous merges as the first node, the edge $g(v)$ is already in **C**. If v is the second or third node of previous merges, $f^1(v) = u$. If v is the fourth node of previous merges, either $f^j(v) = u$, or v cannot be chosen as the first node for $g(v)$ is already in **C**. Hence, $u = f^j(v)$ in **C**. Finally, the fourth node for the merge can be found according to Corollary 3.

Next we consider the time complexity of the algorithm. The size of cycle C is at most V. Hence the complexity of the line (*) is at most $O(V)$. The execution of the lines following (*) takes constant time. From Lemma 12, a $f1$-cycle has length kn and the number of $f1$-cycle is k^{n-1}, hence the number of times line (2) is executed is given by $k^{n-1} = O(V)$. Therefore the complexity of the algorithm is $O(V^2)$. □

Corollary 4 *For $n \geq 3$, G_n^k has a hamiltonian cycle.*

Corollary 5 *For $n \geq 3, k \geq 2$, the graph G_n^k, has an embedding of a ring with dilation, expansion and congestion equal to one.*

5.2 Hamiltonian property of SG_n^k

In this section, we shall derive the Hamiltonian property of SG_n^k, which will be useful in the discussion on the embedding of meshes in G_n^k in the following sub-section.

Definition 10 *Let us refer to the vertices in SG_n^k by $x_1^{p_1}x_2^{p_2}...x_n^{p_n}$, where $1 \leq p_i \leq k$, so that $x_i^{p_i} = p_i$. We call the set of all vertices in the SG_n^k graph the x_0-partition. An x_i partition, for $1 \leq i \leq n$ is defined recursively: In an x_{i-1} partition, let us partition the set of vertices again according to the values of $x_i^{p_i}$, so that vertices in each partition has the same values of $x_1^{p_1}$, $x_2^{p_2}$, ..., $x_i^{p_i}$, and all combinations of the values of $x_{i+1}^{p_{i+1}}, ..., x_n^{p_n}$. Each resulting partition is called an x_i-partition. Note that an x_n-partition is a single vertex.*

Example: Figure 13 illustrates the above hierarchical partitioning for SG_3^3. In the figure, a vertex $i\ j\ k$ in SG_3^3, $1 \leq i, j, k \leq 3$ is represented by $A^iB^jC^k$. The x_1-partitions are represented

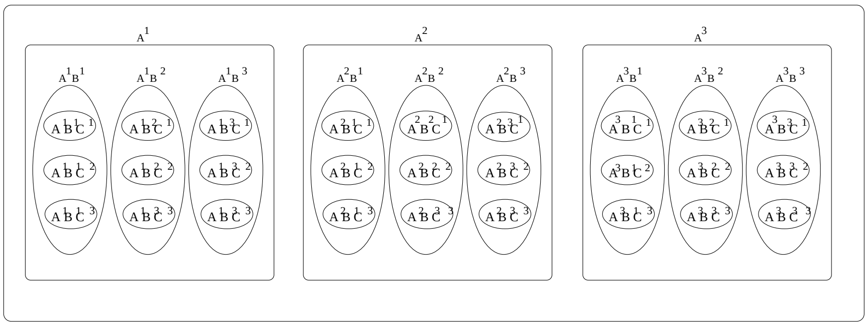

Figure 13: Hierarchical partitioning of SG_3^3

by the boxes labeled A^1, A^2, and A^3. The x_2-partitions in the x_1-partition of A^1 are represented by the oval shapes labeled A^1B^1, A^1B^2, A^1B^3. The x_3-partitions are the vertices themselves.

Definition 11 *Suppose two vertices u and v have all their ranked symbols equal except for the symbol $x_i^{p_i}$. Let s_1 and s_2 be the $x_i^{p_i}$ symbols of u and v, respectively. Then v is the x_{i+1}-neighbour of u if and only if $s_2 = (s_1 + 1) \bmod k$.*

Example: In Figure 13, vertex $A^1B^1C^2$ is the x_3-neighbour of $A^1B^1C^1$. Vertex $A^1B^2C^1$ is the x_2-neighbour of $A^1B^1C^1$. Vertex $A^2B^1C^1$ is the x_1-neighbour of $A^1B^1C^1$.

Remark 3 *For two x_i-partitions $P1$ and $P2$, suppose u in $P1$ is the x_i-neighbour of v in $P2$, u' is the x_{i+1}-neighbour of u in $P1$, and v' is the x_{i+1}-neighbour of v in $P2$. Then u' is the x_i-neighbour of v'.*

Theorem 7 *The graph SG_n^k has a Hamiltonian cycle, for $n \geq 2$.*

Proof: We shall prove by construction. We shall construct Hamiltonian cycles for vertices in each x_n-partition, followed by each x_{n-1}-partitions, followed by each x_{n-2}-partitions and so on until we reach the x_0-partition.

We begin with the x_n-partitions which is trivial since they are single vertices. For each x_{n-1}-partition P, let us pick the edges $\{u, v\}$ in P such that v is the x_n-neighbour of u. It is easy to see that all such x_n-edges form a cycle connecting all vertices in P. Let us call such a cycle an x_{n-1}-**cycle**.

Next we show that the x_{n-1}-cycles for vertices in an x_{n-2}-partition P can be merged to form an x_{n-2}-cycle. Any two x_{n-1}-cycles C_1 and C_2 in P must be connected by some x_{n-1}-edge since

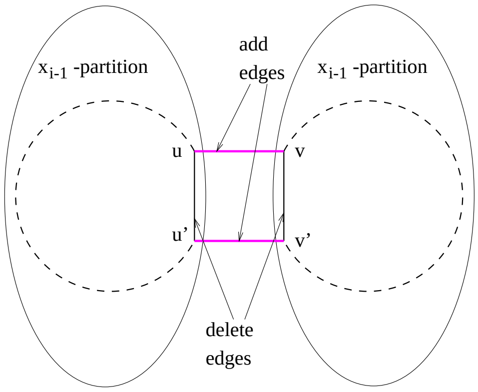

Figure 14: Merging two x_{i-1}-cycles

there must exist vertices u and v in C_1 and C_2 that u is the x_{n-1}-neighbour of v. From Remark 3, if we let $i = n-1$, we can find u' and v' having the properties stated in the remark. C_1 and C_2 are merged to form a larger cycle C if the edges $\{u, u'\}$ and $\{v, v'\}$ are deleted and the edges $\{u, v\}$ and $\{u', v'\}$ are added. This is illustrated in Figure 14. Let us call an edge connecting vertices of 2 different x_n partitions an x_n-**edge**. We can merge more x_{n-1}-cycles to keep enlarger C if we can find edges such as $\{u, u'\}$ in C. The latter condition is true because each x_{n-1}-cycle contains n x_n-edges and on merging 2 such cycles, we have $2n-2$ x_n-edges. The number of x_n-edges in C after merging j x_{n-1}-cycles into it is given by $jn - 2j$ which is greater than 1 if $n > 2$. We can merge all the n x_{n-1}-cycles in P in the above manner and it results in an x_{n-2}-cycle that connects all vertices in P, and this cycle contains $2(n-1)$ x_{n-1}-edges. Similarly we merge the x_n-cycles in each x_{n-1}-partition. We make sure that for each x_{n-1}-partition such as P, the vertices such as u, v, u', v' involved in the merging step are identical except for the symbol $x_{n-1}^{p_{n-1}}$.

The process can be repeated with the x_{n-2}-partitions, the x_{n-3}-partitions, ..., the x_0-partition. When this is done, a Hamiltonian cycle for the SG_n^k graph is formed. $\square$

Next we generalize the definition of a Gray code. A **Gray code** is a b-ary code in which each number from 1 through m, where $m = b^r$, is coded as some r-digit (of radix b) sequence with the property that for any consecutive numbers $i, i+1$, and $m, 1$ the corresponding binary sequences are the same in all but one position. The original definition of a Gray code is usually for a binary code [24]. In the Hamiltonian cycle that we found for SG_n^k, the difference between any two successive vertex labels must be at only one symbol. The sequence of vertex labels form a Gray code.

Corollary 6 *A Gray code can be formed by the set of vertex labels of SG_n^k for $n \geq 2$.* [1]

5.3 Embedding a Mesh

A $K_m \times K_{m-1} \times ... \times K_1$-**mesh** is an interconnection topology consisting of $\Pi_{i=1}^m K_i$ vertices. Each vertex is labeled with an m-digit address $a_{m-1}a_{m-2}...a_0$ where a_i is radix K_i. Each vertex has, at most, a connection to $2m$ other vertices. Given a vertex x with address $a_{m-1}a_{m-2}...a_0$, for each i, $0 \leq i < m$, x has a connection to the vertex with address $a_{m-1}...a_{i+1}(a_i+1)a_{i-1}....a_0$ if $a_i < K_i$ and to the vertex with address $a_{m-1}...a_{i+1}(a_i-1)a_{i+1}....a_0$ if $a_i > 0$.

Let M be a $2^{q_1}k^{n_1} \times 2^{q_2}k^{n_2} \times ... \times 2^{q_m}k^{n_m}$- mesh, where $n_i \geq 3$ for $1 \leq i \leq m$, $n = \sum_{i=1}^m n_i = 2^q$, $\sum_{i=1}^m q_i = q$. We show how to embed M into G_n^k.

If we partition the label of a vertex in SG_n^k into m portions of lengths $n_1, n_2, ...n_m$, in that ordering, then the symbols in the i-th portion can form a label for a graph $SG_{n_i}^k$. From Corollary 6, we can find a Hamiltonian cycle and hence a Gray code for each of the m portions of the label.

For each dimension i of M, $i < m$, we take a set of radix k, n_i digit numbers from 0 to $(k^{n_i}-1)$, which is identical to the set of labels of $SG_{n_i}^k$. We can find a Gray code sequence for $SG_{n_i}^k$. Let the Gray code sequence be $Z_1, Z_2, ...Z_{k^{n_i}}$. Next we find a sequence of q_i binary numbers ranging from 0 to $2^{q_i}-1$. Let the binary sequence be $B_1, B_2, ...B_{2^{q_i}}$ Dimension i is labeled by ordered pairs of: $(Z_1, B_1), (Z_1, B_2), ...(Z_1, B_{2^{q_i}}), (Z_2, B_{2^{q_i}}), (Z_2, B_{2^{1_i}-1}), ...$, arranged in that order. A vertex in M is labeled by the set of labels that are given to its projected locations on dimensions 1 through m.

Let x be a vertex of M with label $(x_1, x_2, ..., x_m)$, where $x_i = (z_i, b_i)$. We refer to the values of z_i in the label as the Z factor and the values of b_i's as the B factor of v. Let $z = z_1z_2...z_m$ be the concatenation of $z_1, z_2, ..., z_m$. Then z is a label of a vertex in SG_n^k. Let c_z be the g-cycle that corresponds to this vertex. Let $b = b_1b_2...b_m$ be the concatenation of $b_1, b_2, ..., b_m$, we start with the leader vertex in c_z and apply the g-function b times. The resulting vertex is $f_V(x)$.

An example is shown in Figure 15 for a $2(2^2) \times 2(2^2)$-mesh that is embedded into G_4^2. In the figure, the vertical labels are for dimension 1 and the horizontal labels are for dimension 2. In particular the labels in both dimensions 1 and 2 follow a Gray code for SG_2^4, which is $\{11, 21, 22, 12\}$. We have highlighted a vertex y at the junction of 22, 1 and 12, 1, $f_V(y)$ is given by starting with $t_1^2t_2^2t_3^1t_4^2$ and applying the g-function 3 times, resulting in $t_4^2t_1^2t_2^2t_3^1$.

In G_n^k, to go from one vertex in a g-cycle C_1 to another vertex in an f-adjacent g-cycle C_2, one

[1] Another way to arrive at the same conclusion is to note that the k-ary n-cubes proposed in [13] is a subgraph of SG_n^k and it is shown in [13] that the k-ary n-cube has a Hamiltonian cycle.

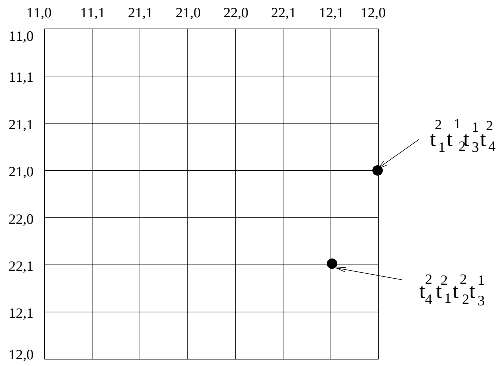

Figure 15: Embedding of $2(2^2) \times 2(2^2)$ mesh in G_4^2

may need to traverse a maximum of n vertices. Adjacent vertices u which is mapped to $F_V(u) = (z_{u1} z_{u2} \cdots z_{um}, b_{u1} b_{u2} \cdots b_{um})$ and v which is mapped to $F_V(u) = (z_{u1} z_{u2} \cdots z_{um}, b_{u1} b_{u2} \cdots b_{um})$ in the mesh can be of two forms:

- u and v only differ in the Z factor in one dimension. $f_V(u)$ and $f_V(v)$ are in the g-cycles C_1, and C_2 that corresponds to two adjacent vertices in SG_n^k, and they are at the same distance from the leader vertex in their g-cycles. One way to go from $f_V(u)$ to $f_V(v)$ in G_n^k, is to traverse at most $n/2$ edges in C_1, traverse one $f1$-edge, and then at most $n/2$ edges in C_2. This distance between $f_V(u)$ and $f_V(v)$ in G_n^k is at most $n+1$ and at most $n/2$ edges in a g-cycle is traversed.
- u and v only differ in the B factor in one dimension. Then vertices $f_V(u)$ and $f_V(v)$ are in the same g-cycle. The distance between $f_V(u)$ and $f_V(v)$ in G_n^k is $\lfloor n/2 \rfloor$ and at most $n/2$ edges in a g-cycle are traversed.

From the above two cases, the dilation of the embedding is given by $n+1$. Next we consider the congestion of the mapping. If we modify the mesh by adding edges between vertices $a_{m-1} a_{m-2} \ldots a_{i+1} K a_{i-1} \ldots a_0$, and $a_{m-1} a_{m-2} \ldots a_{i+1} 1 a_{i-1} \ldots a_0$, for each i, $0 \leq i < m$, then we have a *vertex symmetric network* [2], meaning that the network viewed from any vertex of the network looks the same. We also note that each g-edge in G_n^k is being used in the same way by the mapping function f_V if applied to the modified mesh. Therefore the work load for g-edges is evenly distributed. In the analysis of the dilation, each edge in the mesh will be mapped to a path containing at most n g-edges in G_n^k. Since there are at most $2m$ edges branching out from one vertex in the mesh, the load created for each g-edge is bounded by mn. The congestion for f-edges would be smaller. Therefore the congestion for the embedding is bounded by mn.

Since any Cayley graph is vertex symmetric [2], and the mapping f_V, if applied for the modified mesh, is also vertex symmetrical, in that each vertex is mapped in the same way, the congestion

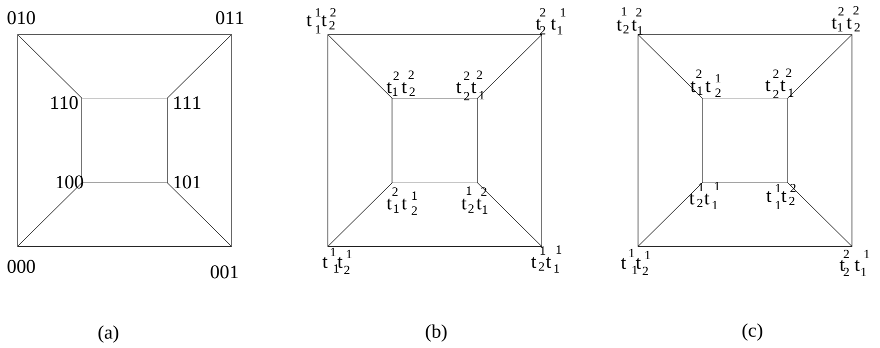

Figure 16: Embedding of a 3-dimensional hypercube in G_2^2

problem for vertices is minimized since the load will be distributed uniformly through all the vertices.

Since M has nk^n vertices and the host graph G_n^k also has nk^n vertices, the expansion of the above embedding is 1.

A dilation of one is not achievable in general since a mesh M as defined above may not be a subgraph of a corresponding G_n^k. Suppose we have a $2(4^2) \times 2(4^2)$ mesh M, the diameter of G_4^4 is 6, and we cannot add edges to the vertices with degree less than 8 in M so that the diameter will become 6 or less.

5.4 Embedding a Hypercube

Given a G_n^k with $k = 2^p \geq 2$, $n = 2^q \geq 3$, for some positive integers p, q. There are $n(k^n) = 2^q \times 2^{pn} = 2^{q+pn}$ vertices in G_n^k. Let Q be a hypercube with a dimension of $q + pn$. We show how to embed Q into G_n^k.

Each vertex of Q can be labeled as a $q + pn$ bit binary number. If $x = x_1 ... x_{q+pn}$, where $x_i \in \{0, 1\}$, is a vertex in Q, then since $k = 2^p$, $x_1 ... x_p$ can be represented by a radix k digit. Let y_i, for $1 \leq i \leq n$, be the radix k digit that corresponds to $x_{(i-1)p+1} x_{(i-1)p+2} ... x_{ip} + 1$. Let z be the number that corresponds to $x_{np+1} ... x_{np+q}$. Since $n = 2^q$, $0 \leq z \leq n - 1$. Define $f_V(x) = g^z(t_1^{y_1} ... t_n^{y_n})$, where g^z indicates the application of the g function for z times. Note that the concatenation of y_1, y_2, ... y_n is the label of the vertex of SG^k which corresponds to the g-cycle that contains x.

In Figure 16, we show the special case of embedding a 3-dimensional hypercube in G_2^2. The labeling of (a) is by binary code for the hypercube, (b) shows the labels of $f_V(x)$ in G_2^2 for each vertex x in (a). Figure 16(c) shows an embedding that has unit dilation and congestion. However, an embedding of unit dilation will not be possible for higher dimensions as the degree of G_n^k will in general be smaller than that of a corresponding hypercube.

Next we consider the dilation for the above embedding scheme. For two vertices x and y that are adjacent in Q, only one of the binary digits is different, there are two cases:

- x and y differ at one of the first pn binary digits. Then $f_V(x)$ and $f_V(y)$ are at the same distance from the leader vertex of 2 g-cycles, C_1 and C_2, which correspond to two adjacent vertices in SG_n^k. To go from $f_V(u)$ to $f_V(v)$ in G_n^k, we have to traverse at most $n/2$ edges in C_1, traverse one $f1$-edge, and then at most $n/2$ edges in C_2. The distance between $f_V(u)$ and $f_V(v)$ in G_n^k is at most $n+1$ and at most $n/2$ edges in a g-cycle is traversed. The distance between $f_V(x)$ and $f_V(y)$ in G_n^k is at most $n+1$.
- x and y differ at one of the last q binary digits. Then $f_V(x)$ and $f_V(y)$ are at the same g-cycle, and the distance between them is at most $\lfloor n/2 \rfloor$.

From the above two cases, the dilation of the embedding is given by $n+1$. Furthermore the hypercube is a *vertex symmetric network* [2]. We also note that each g-edge in G_n^k is being used in the same way by the mapping function f_V. Therefore the work load for g-edges should be evenly distributed. In the analysis of the dilation, each edge in the hypercube will be mapped to a path containing at most n g-edges in G_n^k. Since there are $q+pn$ edges branching out from each vertex in the hypercube, the load created for each g-edge is bounded by $\frac{1}{2}(q+pn)n = \frac{1}{2}(\log_2 n + n\log_2 k)n$. The congestion for f-edges would be smaller. Therefore the congestion for the embedding is bounded by $\frac{1}{2}(\log_2 n + n\log_2 k)n$.

The vertex congestion problem is minimized since the load will be distributed uniformly through all the vertices. Finally the expansion of the embedding is 1.

6 Conclusion

We have introduced a new family of Cayley graphs, called cyclic-cubes, which contains sub-families of graphs with any even fixed-degrees of values greater than or equal to 4. In order to build a realistic interconnection network, it is desirable to have a fixed-degree graph since the number of I/O ports will be independent of the number of nodes in the network. Besides having a fixed degree, a graph in our proposed family has a diameter logarithmic in the number of vertices, and is maximally fault tolerant. It would be a possible choice for the design of interconnection networks when small diameter and high fault tolerance are desirable features. The cyclic-cubes have smaller diameters than the known fixed degree Cayley graphs containing a comparable numbers of vertices, and possess other desirable properties that are shared by Cayley graphs.

By showing that a Cayley graph G_n^k has a Hamiltonian cycle, we show that it has an embedding of a ring. We also consider embeddings of meshes and hypercubes. In the future, we may find methods to develop graphs with any number of nodes based on the cyclic-cube structure. We may look for deadlock free routing scheme. We may also look for optimal graph embeddings and embeddings with fixed dilation and fixed congestion. More study can be done on embedding of other graphs like complete binary trees, pyramids, etc. Some reference on graph embedding are [6, 13, 4]. Another future study for our proposed graphs will be on fault tolerant routing algorithms. Fault tolerant broadcasting would also be an interesting topic. Resource placement [25, 13] is yet another problem to examine for this family of graphs.

Acknowledgements

We thank the anonymous referees for their careful review and very constructive suggestions and comments on the initial draft of the paper.

References

[1] S.B. Akers and B. Krishnamurthy, "On group graphs and their fault tolerance," *IEEE Transactions on Computers*, vol. c-36, pp. 885–888, July 1987.

[2] S.B. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, vol. c-38, pp. 555–566, July 1989.

[3] S. Lakshmivarahan, J.S. Jwo, and S.K. Dhall, "Symmetry in interconnection networks based on cayley graphs of permutation groups: a survey," *Parallel Computing*, vol. 19, pp. 361–407, 1993.

[4] S. Ohring and S.K. Das, "Folded Peterson cube networks: New competitors for the hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 7:2, pp. 151–168, Feb 1996.

[5] S.R. Ohring, F. Sarkar, S.K. Das, and D. H. Hohndel, "Cayley graph connected cycles: A new class of fixed-degree interconnection networks," in *Proc. 28th Annual Hawaii Int. Conf. on System Sciences*, 1995, pp. 479–488.

[6] F. Preparata and J. Vuillemin, "The cube-connected cycles versatile network for parallel computation," *Communications of ACM*, vol. 24(5), pp. 30–39, May 1981.

[7] P. Vadapalli and K. Srimani, "A new family of Cayley graph interconnection networks of constant degree four," *IEEE Transactions on Parallel and Distributed Systems*, vol. 7:1, pp. 26–32, Jan 1996.

[8] C.L. Seitz et al., "The architecture and programming of the Ametek series 2010," in *Proc. Third Conf. Hypercube Concurrent Computers and Applications*, Jan 1988, pp. 33–37.

[9] C.L. Seitz et al., "Submicron systems architecture project semiannual technical report," in *Technical report Caltec-CS-TR-88-18, California Inst. of Technology*, Nov 1988.

[10] W.J. Dally et al., *The J-Machine: A fine-grain concurrent computer*, Elsevier Science Publishers B.V., 1989.

[11] W.J. Dally et al., "The message-driven processor: A multicomputer processing node with efficient mechanisms," *IEEE Micro*, vol. 12(2), pp. 23–39, Apr 1992.

[12] S. Borkar et al, "iWarp: An integrated solution to high-speed parallel computing," in *Proc. Supercomputing'88*, Nov 1988, pp. 330–339.

[13] B. Bose, B. Broeg, Y. Kwon, and Y. Ashir, "Lee distance and topological properties of k-ary n-cubes," *IEEE Transactions on Computers*, pp. 1021–1030, Aug 1995.

[14] N. Tzeng and S. Wei, "Enhanced hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 40:3, pp. 284–294, March 1991.

[15] A. El-Amawy and S. Latifi, "Properties and performance of folded hypercubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 2:1, pp. 31–42, 1991.

[16] D.K. Pradhan, "Fault tolerant VLSI architectures based on de Bruijn graphs (Galileo in the mid nineties)," *DIMACS Series Discrete Math*, vol. vol.5, 1991.

[17] B. Alspach, "Cayley graphs with optimal fault tolerance," *IEEE Transactions on Computers*, pp. 1337–1339, Oct 1992.

[18] L.N. Bhuyan and D.P. Agrawal, "Generalized hypercube and hyperbus structures for a computer network," *IEEE Transactions on Computers*, vol. c-33, pp. 323–333, April 1984.

[19] F. Harary, *Graph Theory*, Addison Wesley, 1971.

[20] G. A. Dirac, "Generalizations du theoreme de Menger," *C.R. Acad. Sci Paris*, pp. 4252–4253, 1960.

[21] K. Menger, "Zur allgemeinen kurventheorie," *Fund. Math*, vol. 10, pp. 96–115, 1927.

[22] P.T. Gaugham and S. Yalamanchalli, "Adaptive routing protocols for hypercube interconnection networks," *IEEE Computer*, May 1993.

[23] M. Marsic, "Hamiltonian circuits in cayley graphs," *Discrete Mathematics*, pp. 49–54, 1982.

[24] A. Tucker, *Applied Combinatorics, 2nd Edition*, John Wiley Sons, 1984.

[25] P. Ramanathan and S. Chalasani, "Resource placement with multiple adjacency constraints in k-ary n-cubes," *IEEE Transactions on Parallel and Distributed Systems*, vol. 6, pp. 511–519, May 1995.

7 Appendix: Proofs of some lemmas

Lemma 2 *A g-cycle C_1 that contains vertex $u = \alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$, in G_n^k is f-adjacent to the g-cycle C_2 that contains $v = \alpha_1^{p_1}...\alpha_{j-1}^{p_{j-1}}\alpha_j^{q_j}\alpha_{j+1}^{p_{j+1}}...\alpha_n^{p_n}$, where for some j, $1 \leq j \leq n$, $q_j \neq p_j$. The g-cycles such as C_2 are all the g-cycles f-adjacent to C_1.*

Proof: Since the g-cycle C_1 contains all circular permutations of $\alpha_1^{p_1}\alpha_2^{p_2}...\alpha_n^{p_n}$, it contains the vertex $u_1 = \alpha_j^{p_j}\alpha_{j+1}^{p_{j+1}}...\alpha_n^{p_n}\alpha_1^{p_1}...\alpha_{j-1}^{p_{j-1}}$. The g-cycle C_2 contains all circular permutations of v and hence it contains the vertex $v_1 = \alpha_{j+1}^{p_{j+1}}...\alpha_n^{p_n}\alpha_1^{p_1}...\alpha_{j-1}^{p_{j-1}}\alpha_j^{q_j}$ Hence $v_1 = f^m(u_1)$, where $m = (k + q_j - p_j) \bmod k$. Therefore C_1 is f-adjacent to C_2. □

Lemma 3 *Each g-cycle in G_n^k is f-adjacent to $(k-1)n$ different g-cycles.*

Proof Let $u_1 = \alpha_1^{j_1}\alpha_2^{j_2}...\alpha_n^{j_n}$ be a vertex in the g-cycle, then the g-cycle contains the following vertices: $u_2 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{j_1}$, $u_3 = \alpha_3^{j_3}...\alpha_n^{j_n}\alpha_1^{j_1}\alpha_2^{j_2}$, ..., $u_n = \alpha_n^{j_n}\alpha_1^{j_1}...\alpha_{n-1}^{j_{n-1}}$. From Lemma 2, the vertex u_i, $1 \leq i \leq n$ is f-adjacent to the $(k-1)$ different g-cycles that contains respectively:
$v_1 = \alpha_{i+1}^{j_{i+1}}...\alpha_n^{j_n}\alpha_1^{j_1}...\ \alpha_i^{(j_i+1)}$, $v_2 = \alpha_{i+1}^{j_{i+1}}...\alpha_n^{j_n}\alpha_1^{j_1}...\ \alpha_i^{(j_i+2)}$, ..., $v_{k-1} = \alpha_{i+1}^{j_{i+1}}...\alpha_n^{j_n}\alpha_1^{j_1}\ ...\ \alpha_i^{(j_i+k-1)}$.
The leader vertices in all such g-cycles are distinct, and hence these g-cycles are also distinct. □

Lemma 4 *Each vertex u in a g-cycle in G_n^k is f-adjacent to $2(k-1)$ different g-cycles.*

Proof: From Lemma 2, if $u = \alpha_1^{j_1}\alpha_2^{j_2}...\alpha_n^{j_n}$, then it is f-adjacent to vertices
$v_1 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+1)}$, $v_2 = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+2)}$, ..., $v_{k-1} = \alpha_2^{j_2}...\alpha_n^{j_n}\alpha_1^{(j_1+k-1)}$,
$v_{-1} = \alpha_n^{(j_n-1)}\alpha_1^{j_1}\alpha_2^{j_2}...\alpha_{n-1}^{j_{n-1}}$, $v_{-2} = \alpha_n^{(j_n-2)}\alpha_1^{j_1}\alpha_2^{j_2}...\alpha_{n-1}^{j_{n-1}}$, ..., $v_{-(k-1)} = \alpha_n^{(j_n-k+1)}\alpha_1^{j_1}\alpha_2^{j_2}...\ \alpha_{n-1}^{j_{n-1}}$.
These vertices belong to g-cycles with different leader vertices and therefore all the g-cycles are distinct. □