CYCLES IN NETWORKS

Arnold L. Rosenberg

Computer and Information Science Department
University of Massachusetts

COINS Technical Report 91-20

CYCLES IN NETWORKS*

Arnold L. Rosenberg
Computer & Information Science
University of Massachusetts
Amherst, Massachusetts 01003

February 15, 1991

Abstract

We study the presence of cycles and long paths in graphs that have been proposed as interconnection networks for parallel architectures. The study surveys and complements known results.

1 Introduction

This paper is devoted to studying embeddings of the simplest possible guest graphs, the path $\mathcal{P}_N$ and the cycle $\mathcal{C}_N$, in graphs that have been proposed as interconnection networks for parallel architectures. In addition to their intrinsic interest, in terms of the development of algorithms on parallel architectures, these two guest graphs are important because of the fact that many structurally richer graphs can be constructed from paths and cycles by various product constructions. A few of the results we present are original; several appear in the literature and are duly cited; many belong to the folklore of the field. Indeed this paper is motivated by a desire to find a single repository for this important, yet scattered material.

Before proceeding further, we define formally the objects of our study.

*This research was supported in part by NSF Grants CCR-88-12567 and CCR-90-13184.

1.1 Background

Notation. A graph $\mathcal{G}$ comprises a set $\mathrm{N}(\mathcal{G})$ of *nodes* and a set $\mathbf{E}(\mathcal{G})$ of *edges*, each edge being an unordered pair of distinct nodes. In just one instance, we study a family of *directed* graphs, *digraphs* for short. For our purposes, a digraph has, in place of edges, a set of *arcs*, each of which is an element of $\mathrm{N}(\mathcal{G}) \times \mathrm{N}(\mathcal{G})$. (Note that digraphs can have *loops*, i.e., arcs of the form (u, u) for $u \in \mathrm{N}(\mathcal{G})$.)

Let n be a positive and k a nonnegative integer. We denote by Z_n the set $Z_n = \{0, 1, \ldots, n-1\}$, and by Z_n^k the set of length-k strings (equivalently, the set of k-place vectors) of elements of Z_n.

Paths and Cycles.

The N-**node path** $\mathcal{P}_N$ and the N-**node cycle** $\mathcal{C}_N$ both have node-set $\mathrm{N}(\mathcal{P}_N) = \mathrm{N}(\mathcal{C}_N) = Z_N$. The edges of $\mathcal{P}_N$ connect every pair of nodes x and $x+1$, for $x \in Z_N - \{N-1\}$; the edges of $\mathcal{C}_N$ connect every pair of nodes x and $x+1 \bmod N$, for $x \in Z_N$. $\mathcal{P}_N$ and $\mathcal{C}_N$ both have N nodes; $\mathcal{P}_N$ has $N-1$ edges; because each node of $\mathcal{C}_N$ has degree 2, the graph has N edges.

The number of edges in a path or cycle is often called its *length*.

Embeddings. An *embedding* of the graph $\mathcal{G}$ in the graph $\mathcal{H}$ comprises two mappings. The *node-assignment* α maps $\mathrm{N}(\mathcal{G})$ one-to-one into $\mathrm{N}(\mathcal{H})$. The *edge-routing* ρ assigns to each edge $\{u, v\} \in \mathbf{E}(\mathcal{G})$ a path in $\mathcal{H}$ connecting nodes $\alpha(u)$ and $\alpha(v)$.

Among the measures of the quality of a graph embedding, the one that dominates is the *dilation* of the embedding, namely, the maximum amount that any edge of $\mathcal{G}$ is "stretched" by the embedding. Formally, the dilation of the embedding (α, ρ) of the graph $\mathcal{G}$ in the graph $\mathcal{H}$ is given by

$$\mathrm{dilation}(\alpha, \rho) = \max_{\{u,v\} \in \mathbf{E}(\mathcal{G})} \mathrm{Length}(\rho(u, v))$$

Although other cost measures are also of interest [26] in general embeddings, when the guest graphs are as simple as paths and cycles, the other measures are usually favorable and can be safely ignored.

1.2 The Focus of Our Study

The main emphasis of our study is on two strong versions of the question of how efficiently paths and cycles can be embedded in a given N-node *host* graph $\mathcal{G}$. Say that $\mathcal{G}$ *contains* the graph $\mathcal{H}$ if $\mathcal{H}$ is a (partial) subgraph of $\mathcal{G}$, or, equivalently, if $\mathcal{H}$ can be embedded in $\mathcal{G}$ with unit dilation.

- $\mathcal{G}$ is *cycle-hamiltonian* if it contains the N-node cycle $\mathcal{C}_N$.
- $\mathcal{G}$ is *path-hamiltonian* if it contains the N-node path $\mathcal{P}_N$.
- $\mathcal{G}$ is *pancyclic* if it contains every M-node cycle $\mathcal{C}_M$, for $3 \le M \le N$.
- $\mathcal{G}$ is *even-pancyclic* if it contains every $2M$-node cycle $\mathcal{C}_{2M}$, for $4 \le 2M \le N$.

The "path" analogue of pancyclicity is uninteresting, because the N-node path $\mathcal{P}_N$ contains every ($M \le N$)-node path. In the same vein, all discussion of paths is uninteresting when a graph is cycle-hamiltonian, because the N-node cycle $\mathcal{C}_N$ contains the N-node path $\mathcal{P}_N$. In deference to common usage, the unqualified adjective "hamiltonian" means "cycle-hamiltonian."

When $\mathcal{G}$ is a *digraph*, then a stronger version of pancyclicity is possible:

- The N-node digraph $\mathcal{G}$ is *di-pancyclic* if it contains every M-node directed cycle $\mathcal{C}_M$, for $1 \le M \le N$.

The specific graphs $\mathcal{G}$ in which we seek cycles are chosen with a twofold motivation. On the one hand, we study the major families of graphs that have been proposed as interconnection networks for parallel architectures—which is one of the application areas in which the existence of cycles is important; on the other hand, we study graph families that allow us to exhibit most of the major ideas that are useful in identifying cycles in graphs.

It is worth mentioning what we do *not* try to cover in this study. First, we do not survey the many results that guarantee the presence of hamiltonian cycles in particular families of graphs (defined, e.g., by structure or by density), although such results abound [1], [9], [38], [41], [42]. Similarly, excepting Theorem 2.1, which indicates how "close" *general* graphs come to being pancyclic, we do not discuss results concerning small-dilation embeddings of cycles in graphs, although such results exist [16]. Finally, we do not discuss the computational difficulty of detecting various classes of cycles in various classes of host graphs. We do remark, however, that the general problem of deciding the containment of cycles in graphs is as computationally intractable as the general question of deciding the embeddability of one arbitrary graph in another. For instance, detecting either cycle- or path-hamiltonianicity in graphs is NP-complete [20], even for graphs with restricted structure [17], [40].

2 General Graphs

Rather than embark immediately on our study of what cycles and paths a graph *contains*, we begin with two results that place the question of cycles in graphs in perspective. The first result shows that even when a cycle of length M cannot be embedded in the $(N \geq M)$-node connected graph $\mathcal{G}$ with *unit* dilation, it can be so embedded with dilation no greater than 3. The second result shows that when the graph $\mathcal{G}$ contains a path of length M, then every cycle of length $\leq M$ can be embedded in $\mathcal{G}$ with dilation no greater than 2. For obvious reasons, we restrict attention to connected graphs $\mathcal{G}$.

The following theorem has been proved many times in the literature. As far as we know, its first appearance was in [34]. The algorithmic proof we present originates in [28].

Theorem 2.1 (Cycle-Embeddings for General Graphs)
(a) *For all N and $M \leq N$, one can embed the M-node cycle $\mathcal{C}_M$ in any N-node connected graph $\mathcal{G}$, with dilation ≤ 3.*
(b) *No smaller dilation works in general.*

Proof. There are a variety of inductive proofs for part (a), but the result is proved most elegantly via the following three-step algorithm that produces an embedding of $\mathcal{C}_N$ in $\mathcal{G}$. To accommodate a cycle of length $M < N$, one just applies this algorithm to any connected M-node subgraph of $\mathcal{G}$.

Step 1. Construct, in any way, a rooted, oriented[1] spanning tree $\mathcal{T}$ for $\mathcal{G}$ [11].

Step 2. Traverse $\mathcal{T}$ in *preorder*:

1. Visit the root r of $\mathcal{T}$; say that r has k children, numbered $0, 1, \ldots, k-1$.
2. For $i = 0, 1, \ldots, k-1$
   traverse the tree rooted at child i of r, in preorder; visit the root r.

During this traversal, each node of $\mathcal{T}$ that has k children will be encountered exactly $k+1$ times. Assign node 0 of $\mathcal{C}_N$ to node r of $\mathcal{T}$, and inductively:

- When the traversal encounters an *even*-level node $v \in \mathrm{N}(\mathcal{T})$ for the *first* time, assign the smallest-numbered not-yet-assigned node of $\mathcal{C}_N$ to v.
- When the traversal encounters an *odd*-level node $v \in \mathrm{N}(\mathcal{T})$ for the *last* time, assign the smallest-numbered not-yet-assigned node of $\mathcal{C}_N$ to v.

[1]A tree $\mathcal{T}$ is *oriented* if the children of each nonleaf node of $\mathcal{T}$ are labelled with distinct numbers.

Step 3. Route edges of $\mathcal{C}_N$ via shortest paths *within* $\mathcal{T}$.

It remains to assess the dilation of the described embedding, which requires an analysis of cases. Focus on a node $i \in \mathbf{N}(\mathcal{C}_N)$ that is assigned to a node $v \in \mathbf{N}(\mathcal{T})$.

Case 1: *Node v resides on an* even *level of $\mathcal{T}$.*

1.1: *v is a leaf of $\mathcal{T}$.*

1.1.1: *v is the highest numbered child of its parent w.*
In this case, node w is visited for the last time in the traversal just after v is visited (for the only time). Node $i+1 \in \mathbf{N}(\mathcal{C}_N)$ is assigned to w, so the edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along an edge of $\mathcal{T}$, hence incurs unit dilation.

1.1.2: *v is not the highest numbered child of its parent w.*
In this case, assuming that v is the jth child of w, the traversal proceeds by visiting node w, followed by the $(j+1)$th child of w, call it u. Since node u also resides at an even level of $\mathcal{T}$, node $i+1 \in \mathbf{N}(\mathcal{C}_N)$ is assigned to u, so the edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a length-2 path in $\mathcal{T}$, hence incurs dilation 2.

1.2: *v is an interior node of $\mathcal{T}$.*

1.2.1: *The* 0*th child u of v is a leaf of $\mathcal{T}$.*
In this case, u is visited just once in the traversal, so node $i+1 \in \mathbf{N}(\mathcal{C}_N)$ is assigned to u, so edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along an edge of $\mathcal{T}$, hence incurs unit dilation.

1.2.2: *The* 0*th child u of v is not a leaf of $\mathcal{T}$.*
In this case, u has a child in $\mathcal{T}$. Since this child resides at an even-numbered level of $\mathcal{T}$, node $i+1 \in \mathbf{N}(\mathcal{C}_N)$ is assigned to that child, so edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a path of length 2 in $\mathcal{T}$, hence incurs dilation 2.

Case 2: *Node v resides on an* odd *level of $\mathcal{T}$.*

2.1: *v has a higher-numbered sibling in $\mathcal{T}$.*
In this case, the parent of v has c children, and v is child $j < c-1$. After the traversal encounters v for the last time, say at step t in the traversal, it encounters the parent of v at step $t+1$ and, at step $t+2$, it encounters the next higher-numbered sibling of v, call it w, for the first time.

2.1.1: *w, the next higher-numbered sibling of v, is a leaf of $\mathcal{T}$.*
In this case, step $t+2$ is also the *last* time that the traversal encounters w, so node $i+1 \in \mathbf{N}(\mathcal{C}_N)$ is assigned to w; hence, edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a path of length 2 in $\mathcal{T}$, incurring dilation 2.

2.1.2: *w, the next higher-numbered sibling of v, is not a leaf of $\mathcal{T}$.*
In this case, the traversal visits the child of w, for the first time, at step $t+3$. Since this child resides at an even level of $\mathcal{T}$, node $i+1 \in \mathrm{N}(\mathcal{C}_N)$ is assigned to that child; hence, edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a path of length 3 in $\mathcal{T}$, incurring dilation 3.

2.2: *v has no higher-numbered sibling in $\mathcal{T}$.*
In this case, after the traversal encounters v for the last time, say at step t, it encounters the parent of v at step $t+1$ and, at step $t+2$, it encounters the parent u of v's parent, if u exists.

2.2.1: *The parent u of v's parent does not exist.*
In this case, the parent of v is the root node r. Since v has no higher-numbered sibling in $\mathcal{T}$, the traversal is complete. Hence, the node of $\mathcal{C}_N$ that is assigned to v is node $N-1$, so that edge $\{0, N-1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along an edge of $\mathcal{T}$, incurring unit dilation.

2.2.2: *The parent of v has a higher-numbered sibling in $\mathcal{T}$.*
In this case, at step $t+3$, the traversal encounters the next sibling of the parent of v, call it z, for the first time. Since z resides on an even-numbered level of $\mathcal{T}$, node $i+1 \in \mathrm{N}(\mathcal{C}_N)$ is assigned to z; hence, edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a path of length 3 in $\mathcal{T}$, incurring dilation 3.

2.2.3: *The parent of v has no higher-numbered sibling in $\mathcal{T}$.*
In this case, step $t+2$ is the last time that the traversal encounters u, so node $i+1 \in \mathrm{N}(\mathcal{C}_N)$ is assigned to node u; hence, edge $\{i, i+1\} \in \mathbf{E}(\mathcal{C}_N)$ is routed along a path of length 2 in $\mathcal{T}$, incurring dilation 2.

These cases exhaust all possibilities. We conclude that the described embedding incurs dilation ≤ 3, as was claimed.

We establish Part (b) by explicit example. Possibly the simplest graph $\mathcal{G}$ that requires dilation 3 is the 7-node tree consisting of a root node that has three children, each of which has one child (which is a leaf). Assume, for contradiction, that one can embed the cycle $\mathcal{C}_7$ in $\mathcal{G}$ with dilation 2. We show that this assumption forces enough of the node-assignments of this embedding to preclude the existence of the embedding.

Because of the node-transitivity[2] of $\mathcal{C}_7$, we lose no generality by assuming that node $0 \in \mathrm{N}(\mathcal{C}_7)$ is assigned to the root node r of $\mathcal{G}$.

Fact 2.1 *Node $1 \in \mathrm{N}(\mathcal{C}_7)$ is assigned to some leaf of $\mathcal{G}$.*

[2] A graph $\mathcal{G}$ is *node transitive* if, for every ordered pair of nodes $u, v \in \mathrm{N}(\mathcal{G})$, there is an automorphism of $\mathcal{G}$ that maps node u to node v. Cycles are transparently node transitive.

Verification. Say, for contradiction, that node 1 is assigned to a child v of the root of $\mathcal{G}$. Two possibilities arise. If node $2 \in \mathrm{N}(\mathcal{C}_7)$ is assigned to the leaf-child w of v, then node $3 \in \mathrm{N}(\mathcal{C}_7)$ must be assigned at distance ≥ 3 from node 2. If node $2 \in \mathrm{N}(\mathcal{C}_7)$ is assigned to any node of $\mathcal{G}$ other than to w, then the node $j \in \mathrm{N}(\mathcal{C}_7)$ that is assigned to w must reside at distance ≥ 3 from nodes $j \pm 1 \in \mathrm{N}(\mathcal{C}_7)$. Either of these contingencies contradicts the claim that the embedding of $\mathcal{C}_7$ in $\mathcal{G}$ has dilation 2.

Say, then, that node $1 \in \mathrm{N}(\mathcal{C}_7)$ is assigned to the leaf $w \in \mathrm{N}(\mathcal{G})$.

Fact 2.2 *Node* $2 \in \mathrm{N}(\mathcal{C}_7)$ *is assigned to the parent of leaf* w *in* $\mathcal{G}$.

Verification. Leaf w has one node of $\mathcal{G}$ at distance 1, namely, its parent; it has one node of $\mathcal{G}$ at distance 2, namely the root r. Therefore, if node $2 \in \mathrm{N}(\mathcal{C}_7)$ is not assigned to the parent of leaf w, it must be assigned at distance ≥ 3 from node 1.

Fact 2.3 *Node* $3 \in \mathrm{N}(\mathcal{C}_7)$ *is assigned to a child of the root of* $\mathcal{G}$.

Verification. Node 3 must be assigned within distance 2 of node 2, and only the remaining children of the root, among all unoccupied nodes of $\mathcal{G}$, satisfy this condition.

We have now painted ourselves into a corner. We must assign node $4 \in \mathrm{N}(\mathcal{C}_7)$ either to the leaf-child of the host node of node $3 \in \mathrm{N}(\mathcal{C}_7)$ or to the one remaining child of the root of $\mathcal{G}$. If we make the former assignment, then we cannot place node $5 \in \mathrm{N}(\mathcal{C}_7)$ within distance 2 of node 4. If we make the latter assignment, then we cannot assign any node of $\mathcal{C}_7$ to the leaf-child of the host node of node $3 \in \mathrm{N}(\mathcal{C}_7)$ while honoring the claimed dilation 2 of the embedding. □

One general situation in which one can improve on the bound of Theorem 2.1 is when the graph $\mathcal{G}$ contains a long path.

Proposition 2.1 (a) *If the graph* $\mathcal{G}$ *contains the* N*-node path* $\mathcal{P}_N$*, then any cycle of length* $3 \leq M \leq N$ *can be embedded in* $\mathcal{G}$ *with dilation* ≤ 2.
(b) *If the graph* $\mathcal{G}$ *contains the* N*-node cycle* $\mathcal{C}_N$*, then any cycle of length* $3 \leq M \leq N$ *can be embedded in* $\mathcal{G}$ *with dilation* ≤ 2.

Proof. The result is immediate from the strong version of *quasi-isometry*[3] enjoyed by paths and cycles; specifically, paths are (spanning) subgraphs of like-length cycles, and cycles can be embedded in paths with dilation 2, as we now see.

[3]Graphs $\mathcal{G}$ and $\mathcal{H}$ are *quasi-isometric* when each can be embedded in the other with dilation $O(1)$.

Proposition 2.2 (Quasi-isometry of Paths and Cycles)
(a) *For all N, the N-node cycle $\mathcal{C}_N$ contains the N-node path $\mathcal{P}_N$.*
(b) *For all N, $\mathcal{C}_N$ can be embedded in $\mathcal{P}_N$ with dilation 2.*

Proof. Part (a) being obvious, we concentrate on part (b).

We note first that cycles of lengths 1 and 2 are degenerate, hence cannot appear in graphs that lack loops and parallel edges.

The proof of part (b) is given most elegantly via an algorithm for effecting the desired embedding. To embed the N-node cycle $\mathcal{C}_N$ in $\mathcal{P}_N$, we take an N-step "walk" along $\mathcal{P}_N$, depositing nodes of $\mathcal{C}_N$ as we go. During step i of the "walk," $0 \leq i \leq N-1$, we visit node i of $\mathcal{P}_N$. When i is even, we deposit node $i/2$ of $\mathcal{C}_N$ at node i of $\mathcal{P}_N$; when i is odd, we deposit node $N - \lceil i/2 \rceil$ of $\mathcal{C}_N$ at node i of $\mathcal{P}_N$.

To see that this simple embedding has dilation 2, note that odd and even steps of the "walk" alternate, and that $(i+2)/2 = i/2+1$, whence also $N - \lceil (i+2)/2 \rceil = N - \lceil i/2 \rceil - 1$. □

For many familiar families of graphs, one can improve on the general embeddings of Theorem 2.1 and Proposition 2.1. The remainder of the paper expands on this observation.

3 Clique-Like Graphs

We start our study with two structurally simple host graph families, the complete graphs and the complete bipartite graphs.

The N-**node complete graph**, or, **clique** $\mathcal{K}_N$ has node-set $\mathrm{N}(\mathcal{K}_N) = Z_N$; its edges connect every pair of distinct nodes. Because $\mathcal{K}_N$ has N nodes, each of degree $(N-1)$, it has

$$\binom{N}{2}$$

edges.

The $m \times n$ **complete bipartite graph** $\mathcal{K}_{m,n}$ has node-set $(\{0\} \times Z_m) \cup (\{1\} \times Z_n)$; its edges connect every pair of nodes $u \in \{0\} \times Z_m$ and $v \in \{1\} \times Z_n$. $\mathcal{K}_{m,n}$ has $m+n$ nodes and mn edges.

While clique-like graphs are too dense to be considered seriously as a physical interconnection network (at least for large numbers of processors), one does try to approximate their characteristics with physically feasible networks, so it is worthwhile understanding those characteristics.

Theorem 3.1 (Cycle-Embeddings for $\mathcal{K}_N$)
For all N, the N-node clique $\mathcal{K}_N$ is pancyclic.

Proof. The proof is immediate by the fact that every pair of nodes of $\mathcal{K}_N$ are connected by an edge. To wit, for all integers $m \leq N$, there is a cycle in $\mathcal{K}_N$ connecting the following nodes, in the following order.

$$0, 1, \ldots, m-1, 0.$$

□

Theorem 3.2 (Cycle-Embeddings for $\mathcal{K}_{m,n}$)
(a) *For all $m, n, M \leq \min(m, n)$, the $m \times n$ complete bipartite graph $\mathcal{K}_{m,n}$ contains every $2M$-node cycle C_{2M}; in particular, $\mathcal{K}_{n,n}$ is even-pancyclic.*
(b) *$\mathcal{K}_{m,n}$ contains only those cycles described in Part (a).*

Proof. Part (a) is immediate by the fact that every pair of nodes $\{u, v\}$ where $u \in \{0\} \times Z_m$ and $v \in \{1\} \times Z_n$ are connected by an edge in $\mathcal{K}_{m,n}$. To wit, for all integers $M \leq \min(m, n)$, there is a cycle in $\mathcal{K}_N$ connecting the following nodes, in the following order.

$$(0,0), (1,0), (0,1), (1,1), \ldots, (0, M-1), (1, M-1), (0,0).$$

Part (b) is established via two observations. First, $\mathcal{K}_{m,n}$ is bipartite, hence contains no odd-length cycle. Second, every cycle in $\mathcal{K}_{m,n}$ alternates nodes from $\{0\} \times Z_m$ with nodes from $\{1\} \times Z_n$, hence can contain no more than $2\min(m, n)$ nodes. □

4 Meshes and Toroidal Meshes

We continue our study with two families of graphs whose attractiveness as interconnection networks derive from a combination of their amenability to efficient layout in the plane and their regular structure that simplifies the development of algorithms; cf. [5], [12].

4.1 Meshes

The **2-dimensional** $m \times n$ **rectangular mesh** $\mathcal{M}_{m,n}$ is the product graph $\mathcal{P}_m \times \mathcal{P}_n$. Therefore, $\mathcal{M}_{m,n}$ has node-set $\mathbf{N}(\mathcal{M}_{m,n}) = Z_m \times Z_n$; its edges come in two classes. $\mathcal{M}_{m,n}$ has a *row-edge* between every pair of nodes

$$(u,\ v) \text{ and } (u,\ w),$$

where $|v - w| = 1$; it has a *column-edge* between every pair of nodes

$$(u,\ v) \text{ and } (w,\ v),$$

where $|u - w| = 1$. $\mathcal{M}_{m,n}$ has mn nodes and $2mn - m - n$ edges.

We note first that meshes contain paths of all possible lengths.

Theorem 4.1 (Path- and Cycle-Embeddings for $\mathcal{M}_{m,n}$)
For all m and n:
(a) *The $m \times n$ mesh $\mathcal{M}_{m,n}$ is path-hamiltonian.*
(b) *$\mathcal{M}_{m,n}$ contains no odd-length cycle; in particular, when mn is odd, $\mathcal{M}_{m,n}$ is* not *hamiltonian.*
(c) *$\mathcal{M}_{m,n}$ is even-pancyclic; in particular, when mn is even, $\mathcal{M}_{m,n}$ is hamiltonian.*

Proof. (a) The path-hamiltonianicity of $\mathcal{M}_{m,n}$ is proved most easily by "snaking" the path $\mathcal{P}_{mn}$ through $\mathcal{M}_{m,n}$, say, row by row. More formally, the embedding specifies a path in $\mathcal{M}_{m,n}$ that starts at node $(0,0)$ and proceeds as follows. For all $i \in Z_{\lfloor m/2 \rfloor}$, the path

- proceeds along row $2i$ from node $(2i, 0)$ to node $(2i, n-1)$
- terminates there if $m - 1 = 2i$; otherwise, it follows the edge from node $(2i, n-1)$ to node $(2i+1, n-1)$
- proceeds along row $2i+1$ from node $(2i+1, n-1)$ to node $(2i+1, 0)$.

This path always exists; by our description, it specifies an embedding of $\mathcal{P}_{mn}$ in $\mathcal{M}_{m,n}$, with unit dilation.

(b) For all m and n, $\mathcal{M}_{m,n}$ is bipartite, hence contains no odd-length cycle. One coloring that effects a bipartition of $\mathcal{M}_{m,n}$ assigns node $(u,v) \in \mathbf{N}(\mathcal{M}_{m,n})$ a "color" which is the parity of $u + v$, i.e., the quantity $u + v \bmod 2$. Since a hamiltonian cycle in $\mathcal{M}_{m,n}$ would contain the same number of nodes as $\mathcal{M}_{m,n}$, namely mn, $\mathcal{M}_{m,n}$ cannot be hamiltonian when mn is odd.

(c) The embeddings that establish the even-pancyclicity of $\mathcal{M}_{m,n}$ proceed by iteratively contracting a specific maximum-length cycle that $\mathcal{M}_{m,n}$ contains. When mn is even, this longest cycle has mn nodes, so that $\mathcal{M}_{m,n}$ is hamiltonian; when mn is odd, this longest cycle has $mn - 1$ nodes. Our construction of the desired maximum-length cycle builds upon the embedding of the mn-node path in $\mathcal{M}_{m,n}$, in Part (a). Constructing this cycle is our first goal.

Assume first that mn is even. Say, with no loss of generality, that m is even; otherwise, switch the roles of m and n. Isolate the $m \times (n-1)$ submesh $\mathcal{M}$ of $\mathcal{M}_{m,n}$ obtained by removing column 0. Embed the $m(n-1)$-node path $\mathcal{P}_{m(n-1)}$ with unit dilation in $\mathcal{M}$, using the embedding of Part (a). This embedding assigns node 0 of $\mathcal{P}_{m(n-1)}$ to node $(0,1)$ of $\mathcal{M}_{m,n}$, and it assigns node $m(n-1)-1$ of $\mathcal{P}_{m(n-1)}$ to node $(m-1,1)$ of $\mathcal{M}_{m,n}$. We complete the desired cycle by connecting the two ends of the embedded copy of $\mathcal{P}_{m(n-1)}$ via the path in $\mathcal{M}_{m,n}$ that proceeds from node $(0,1)$ to node $(0,0)$, thence along column 0 to node $(m-1,0)$, and finally goes to node $(m-1,1)$. The cycle so constructed is a hamiltonian cycle in $\mathcal{M}_{m,n}$.

When mn is odd, we isolate the $(m-2) \times (n-1)$ submesh $\mathcal{M}$ of $\mathcal{M}_{m,n}$ obtained by removing column 0 and rows $m-2$ and $m-1$. Embed the $(m-2)(n-1)$-node path $\mathcal{P}_{(m-2)(n-1)}$ with unit dilation in $\mathcal{M}$, using the embedding of Part (a). This embedding assigns node 0 of $\mathcal{P}_{m(n-1)}$ to node $(0,1)$ of $\mathcal{M}_{m,n}$, and it assigns node $(m-2)(n-1)-1$ of $\mathcal{P}_{m(n-1)}$ to node $(m-3,n-1)$ of $\mathcal{M}_{m,n}$. We complete the desired cycle by connecting the two ends of the embedded copy of $\mathcal{P}_{(m-2)(n-1)}$ via the path in $\mathcal{M}_{m,n}$ that proceeds from node $(0,1)$ to node $(0,0)$, thence along column 0 to node $(m-1,0)$, thence by a "sawtooth" path to node $(m-1,n-3)$, and finally by a length-4 "dogleg" path to node $(m-3,n-1)$. Each "tooth" in the "sawtooth" path is a length-4 path of the form

$$(m-1,2i) \longleftrightarrow (m-1,2i+1) \longleftrightarrow (m-2,2i+1) \longleftrightarrow (m-2,2i+2) \longleftrightarrow (m-1,2i+2);$$

the entire "sawtooth" path comprises $(n-3)/2$ teeth. The length-4 "dogleg" path is specified by

$$(m-1,n-3) \longleftrightarrow (m-1,n-2) \longleftrightarrow (m-2,n-2) \longleftrightarrow (m-2,n-1) \longleftrightarrow (m-3,n-1).$$

The cycle so constructed misses only node $(m-1,n-1)$ of $\mathcal{M}_{m,n}$.

Once having constructed the specified maximum-length cycle in $\mathcal{M}_{m,n}$, we construct cycles of all other even lengths by iterating the following contraction operations, in a judiciously but simply chosen order. The first contraction operation replaces a length-3 path of the form

$$(k,\ell) \longleftrightarrow (k,\ell+1) \longleftrightarrow (k+1,\ell+1) \longleftrightarrow (k+1,\ell)$$

by the edge

$$(k,\ell) \longleftrightarrow (k+1,\ell).$$

The second contraction operation replaces a length-3 path of the form

$$(k,\ell) \longleftrightarrow (k+1,\ell) \longleftrightarrow (k+1,\ell+1) \longleftrightarrow (k,\ell+1)$$

by the edge

$$(k,\ell) \longleftrightarrow (k,\ell+1).$$

The details will follow readily after trying a few examples. □

4.2 Toroidal Meshes

The **2-dimensional** $m \times n$ **toroidal mesh** $\widetilde{\mathcal{M}}_{m,n}$ is the product graph $\mathcal{C}_m \times \mathcal{C}_n$. Therefore, $\widetilde{\mathcal{M}}_{m,n}$ has node-set $\mathrm{N}(\widetilde{\mathcal{M}}_{m,n}) = Z_m \times Z_n$; its edges come in two classes. $\widetilde{\mathcal{M}}_{m,n}$ has a *row-edge* between every pair of nodes

$$(u,\ v) \text{ and } (u,\ v+1 \bmod n);$$

it has a *column-edge* between every pair of nodes

$$(u,\ v) \text{ and } (u+1 \bmod m,\ v).$$

Because $\widetilde{\mathcal{M}}_{m,n}$ has mn nodes, each of degree 4, it has $2mn$ edges.

The conditions under which the toroidal mesh $\widetilde{\mathcal{M}}_{m,n}$ contains a cycle is even more complicated than for the "flat" mesh $\mathcal{M}_{m,n}$.

Theorem 4.2 (Cycle-Embeddings for $\widetilde{\mathcal{M}}_{m,n}$)
For all m and n:
(a) *The $m \times n$ toroidal mesh $\widetilde{\mathcal{M}}_{m,n}$ is hamiltonian.*
(b) *$\widetilde{\mathcal{M}}_{m,n}$ is even-pancyclic.*
(c) *$\widetilde{\mathcal{M}}_{m,n}$ contains no odd-length cycle of length $< \min(m,n)$.*
(d) *When both m and n are even, then $\widetilde{\mathcal{M}}_{m,n}$ contains no odd-length cycle.*

Proof. **(a)** When mn is even, the hamiltonianicity of $\widetilde{\mathcal{M}}_{m,n}$ follows from Theorem 4.1(c), because $\mathcal{M}_{m,n}$ is a spanning subgraph of $\widetilde{\mathcal{M}}_{m,n}$.

When mn is odd, the result follows by combining the ideas behind the embeddings of Theorems 4.1(c) and 4.1(a). We begin by removing column 0 of $\widetilde{\mathcal{M}}_{m,n}$ and embedding the $m(n-1)$-node path $\mathcal{P}_{m(n-1)}$ with unit dilation in the resulting submesh of $\widetilde{\mathcal{M}}_{m,n}$, using the embedding of Theorem 4.1(a) (as we did in Theorem 4.1(c)). Because mn is odd, this embedding assigns node 0 of $\mathcal{P}_{m(n-1)}$ to node $(0,1)$ of $\widetilde{\mathcal{M}}_{m,n}$, and it assigns node $m(n-1)-1$ of $\mathcal{P}_{m(n-1)}$ to node $(m-1, n-1)$ of $\widetilde{\mathcal{M}}_{m,n}$. We now "connect" the endpoints of path $\mathcal{P}_{m(n-1)}$ together, turning the embedding into an embedding of $\mathcal{C}_{mn}$ in $\widetilde{\mathcal{M}}_{m,n}$, in three steps: First, we extend the path along the edge in $\widetilde{\mathcal{M}}_{m,n}$ from node $(m-1, n-1)$ to node $(m-1, 0)$; next, we extend the path along column 0 of $\widetilde{\mathcal{M}}_{m,n}$, from node $(m-1, 0)$ to node $(0,0)$; finally, we close the path (thereby forming a cycle), using the edge connecting nodes $(0,0)$ and $(0,1)$. We have thus described an embedding of $\mathcal{C}_{mn}$ in $\widetilde{\mathcal{M}}_{m,n}$, with unit dilation.

(b) The even-pancyclicity of $\widetilde{\mathcal{M}}_{m,n}$ follows from Theorem 4.1(c), because $\mathcal{M}_{m,n}$ is a spanning subgraph of $\widetilde{\mathcal{M}}_{m,n}$.

(c) Any embedding of a "short" cycle in $\widetilde{\mathcal{M}}_{m,n}$ in fact embeds the cycle in a subgraph of $\widetilde{\mathcal{M}}_{m,n}$ which is a mesh. By Theorem 4.1(b), this subgraph contains no odd-length cycle.

(d) When both m and n are even, $\widetilde{\mathcal{M}}_{m,n}$ is bipartite, hence contains no odd-length cycle. A two-coloring of $\widetilde{\mathcal{M}}_{m,n}$ that witnesses its bipartiteness is obtained by 2-coloring each of its rows, each of which is an even-length cycle, with the colors 1 and 2, switching the roles of the colors according to the parity of the row number. Since each column of $\widetilde{\mathcal{M}}_{m,n}$ is also an even-length cycle, this procedure yields a valid 2-coloring of $\widetilde{\mathcal{M}}_{m,n}$. □

5 Hypercubes

The next family in our study produces some of the most versatile interconnection networks, both in terms of their ability to emulate a large variety of other networks [8], [10], [18], [29] and in terms of their ability to support efficient interprocessor communication [14], [19], [30], [35].

The **n-dimensional (boolean) hypercube** $\mathcal{Q}_n$ is the n-fold product graph $\mathcal{K}_2 \times \mathcal{K}_2 \times \cdots \times \mathcal{K}_2 =_{\text{def}} \mathcal{K}_2^n$. Therefore, $\mathcal{Q}_n$ has node-set $\text{N}(\mathcal{Q}_n) = Z_2^n$ which comprises all length-n strings over the alphabet Z_2; $\mathcal{Q}_n$ has an edge between every pair of nodes $x\beta y$ and $x\gamma y$ that differ in precisely one bit-position, i.e., $x \in Z_2^k$ is some length-k string, $y \in Z_2^{n-k-1}$ is some length-$(n-k-1)$ string, and both β and $\gamma \neq \beta$ are in Z_2 (so, are bits, or, length-1 strings). Because $\mathcal{Q}_n$ has 2^n nodes, each of degree n, it has $n2^{n-1}$ edges.

Our proof of part (a) of the following theorem derives from [18].

Theorem 5.1 (Cycle-Embeddings for $\mathcal{Q}_n$)
For all n:
(a) *The n-dimensional hypercube $\mathcal{Q}_n$ is even-pancyclic; in particular, $\mathcal{Q}_n$ is hamiltonian.*
(b) *$\mathcal{Q}_n$ contains no odd-length cycle.*

Proof. **(a)** A *length-m n-dimensional Gray code* is a cyclically ordered sequence of m distinct length-n binary words, having the property that words adjacent in the sequence differ in precisely one bit-position. In the obvious way, such a Gray code can be viewed as (specifying) an m-node cycle in the $\mathcal{Q}_n$.

A length-m n-dimensional Gray code can be specified by a length-m *transition sequence* of integers from the set Z_n: the ith element of the sequence specifies the bit-position that is flipped to obtain the $(i+1)$th word of the code from the ith word. A simple family of transition sequences derive from the following recursive construction.

$$\begin{aligned} \mathbf{S}_1 &= 0 \\ \mathbf{S}_{k+1} &= \mathbf{S}_k, k, \mathbf{S}_k \end{aligned}$$

We denote by $\mathbf{S}_r^{(i)}$ the ith element of $\mathbf{S}_r$ (counting, as usual, from 0). For any integer $k \le 2^{r-1}$, one uses $\mathbf{S}_r$ to construct a length-$2k$ n-dimensional Gray code $x_0, x_1, \ldots, x_{2k-1}$, as follows.

1. Select any length-n binary string as word x_0 of the code.
2. For $0 \le i < k-1$, generate word x_{i+1} by flipping bit-position $\mathbf{S}_r^{(i)}$ of x_i.
3. Generate word x_k by flipping bit-position n of x_{k-1}.
4. For $0 \le i < k-1$, generate word x_{k+i+1} by flipping bit-position $\mathbf{S}_r^{(i)}$ of x_{k+i}.

It follows automatically that word x_0 is obtained by flipping bit-position n of word x_{2k-1}. Let us denote by $\mathbf{S}_r[2k]$ the sequence of bit-positions flipped in this procedure:

$$\mathbf{S}_r[2k]^{(i)} = \begin{cases} \mathbf{S}_r^{(i)} & \text{if } i \in \{0, 1, \ldots, k-2\} \\ n & \text{if } i \in \{k-1, 2k-1\} \\ \mathbf{S}_r^{(i-k)} & \text{if } i \in \{k, k+1, \ldots, 2k-2\} \end{cases}$$

One sees that the defined transition sequences generate all even-length cycles in $\mathcal{Q}_n$ via a number of easily verified facts, whose verifications we just hint at.

First, we note a consequence of the fact that every two occurrences of an integer h in $\mathbf{S}_r$ are separated by an occurrence of some integer $\ge h+1$.

Fact 5.1 *Every contiguous subsequence of* $\mathbf{S}_r$ *contains at least one element an odd number of times.*

Using Fact 5.1 on prefixes of $\mathbf{S}_r$, one verifies that transition sequences do in fact generate Gray codes.

Fact 5.2 *The just-described procedure generates a length-$2k$ n-dimensional Gray code.*

Part (a) of the theorem follows directly from Fact 5.2.

(b) For all n, $\mathcal{Q}_n$ is bipartite, hence contains no odd-length cycle. A valid 2-coloring of $\mathcal{Q}_n$ results from labeling each node of $\mathcal{Q}_n$ with the parity of the number of 1s in its string-name. □

6 Butterflies and Cube-Connected Cycles

This section studies the cycle structure of two of the benchmark bounded-degree "approximations" of the hypercube [7], [24], [25], [32].

The **order-n butterfly graph** $\mathcal{B}_n$ and the **order-n cube-connected cycles graph** CCC_n both have node-set $N(\mathcal{B}_n) = N(CCC_n) = Z_n \times Z_2^n$. For each node, $(\ell, w) \in N(\mathcal{B}_n) = N(CCC_n)$, we call $\ell \in Z_n$ the *level* of the node and $w \in Z_2^n$ the *position-within-level string* (*PWL string*, for short) of the node. Each node, $(\ell, w) \in N(\mathcal{B}_n) = N(CCC_n)$ is connected in both graphs via a *straight-edge* to node $(\ell + 1 \bmod n,\ w)$. Additionally, each edge $(\ell,\ x\beta y) \in N(\mathcal{B}_n) = N(CCC_n)$, where $x \in Z_2^\ell$, $y \in Z_2^{n-\ell-1}$, and $\beta \in Z_2$, is connected in $\mathcal{B}_n$ via a *cross-edge* to node[4] $(\ell + 1 \bmod n,\ x\bar{\beta}y)$ and is connected in CCC_n via a *level-edge* to node$(\ell,\ x\bar{\beta}y)$. Both $\mathcal{B}_n$ and CCC_n have $n2^n$ nodes; in $\mathcal{B}_n$ each node has degree 4, so $\mathcal{B}_n$ has $n2^{n+1}$ edges; in CCC_n each node has degree 3, so CCC_n has $n(2^n + 2^{n-1})$ edges.

We study butterfly and cube-connected cycles graphs in the same section because the undirected order-n CCC CCC_n and the undirected order-n butterfly graph $\mathcal{B}_n$ are quasi-isometric; in fact, CCC_n is a (spanning) subgraph of $\mathcal{B}_n$, and one can embed $\mathcal{B}_n$ in CCC_n, with dilation 2. The strong quasi-isometry exposed in part (a) of the following Proposition originates in [15].

Proposition 6.1 (Quasi-isometry of $\mathcal{B}_n$ and CCC_n)
(a) *For all n, the undirected order-n butterfly graph $\mathcal{B}_n$ contains the undirected order-n cube-connected cycles graph CCC_n.*
(b) *For all n, one can embed $\mathcal{B}_n$ in CCC_n, with dilation 2.*

Proof. (a) Consider the following assignment of nodes of CCC_n to nodes of $\mathcal{B}_n$. If the PWL string $x \in Z_2^n$ of node $v = (\ell,\ x)$ of CCC_n contains an *even* number of 1s, then assign node v to node v of $\mathcal{B}_n$; if the PWL contains an *odd* number of 1s, then assign node v to node $(\ell + 1,\ x)$ of $\mathcal{B}_n$. We now verify that this assignment witnesses the claimed subgraph relation.

Inter-level Adjacencies. Every straight-edge of CCC_n maps onto a straight-edge of $\mathcal{B}_n$. This is true because each node v of CCC_n is assigned to the "column" of $\mathcal{B}_n$ that is defined by the same PWL string as v's; either all nodes of CCC_n assigned to that "column" of $\mathcal{B}_n$ remain in the same level they had in CCC_n, or they all shift one level. In either case, inter-level adjacencies are preserved.

Bijectiveness of Assignment. As a corollary of the preservation of inter-level adjacencies, our assignment of nodes is both one-to-one and onto.

[4] For $\beta \in Z_2$, $\bar{\beta} = 1 - \beta$.

Intra-level Adjacencies. Each level-edge of CCC_n connects a node u whose PWL string has an even number of 1s with a node v whose PWL string has an odd number of 1s. The same is true for each cross-edge of $\mathcal{B}_n$, but these latter edges also shift one level. To be more specific, let $x, x' \in Z_2^n$ be length-n binary strings that differ in bit-position $\ell \in Z_n$. On the one hand, there is an edge in CCC_n connecting nodes (ℓ, x) and (ℓ, x'); on the other hand, there is an edge in $\mathcal{B}_n$ connecting nodes (ℓ, x) and $(\ell + 1 \bmod n, x')$, as well as an edge connecting nodes (ℓ, x') and $(\ell+1 \bmod n, x)$. Since one of x and x' must have an even number of 1s while the other has an odd number of 1s, one of these cross-edges of $\mathcal{B}_n$ must be the induced image, under our node-assignment, of the level-edge of CCC_n.

Our node-assignment is thus both one-to-one and onto, and each edge of CCC_n is mapped by the assignment to an edge of $\mathcal{B}_n$. This completes the proof of Part (a).

(b) For Part (b), we employ the *identity* node-assignment. Ignoring straight-edges, which are common to $\mathcal{B}_n$ and CCC_n, hence can be routed using the identity routing, we route edge $((\ell, x),\ (\ell + 1 \bmod n, x'))$ of $\mathcal{B}_n$ along the following length-2 path in CCC_n:

$$(\ell, x) \leftrightarrow (\ell, x') \leftrightarrow (\ell + 1 \bmod n, x').$$

□

Proposition 6.1 demonstrates that the presence of any cycle in CCC_n betokens the presence of a like-length cycle in $\mathcal{B}_n$, and, contrapositively, the absence of any cycle in $\mathcal{B}_n$ precludes the presence of a like-length cycle in CCC_n. We turn finally to the cycle structure of the two graphs.

Theorem 6.1 (Cycle-Embeddings for $\mathcal{B}_n$ and CCC_n)
For all n:
(a) *The order-n cube-connected cycles graph CCC_n contains the m-node cycle C_m, for the following values of m:*

- $m = n$
- $m = n2^k - (n-2)c$ *for* $2 \le k \le n$ *and* $0 \le c \le 2^k$.

In particular, CCC_n is hamiltonian.
(b) *The order-n butterfly graph $\mathcal{B}_n$ contains the m-node cycle C_m, for the following values of m:*

- $m = n$
- $m = n2^k - (n-2)c$ *for* $1 \le k \le n$ *and* $0 \le c \le 2^k$.

In particular, $\mathcal{B}_n$ is hamiltonian.
(c) *Neither $\mathcal{B}_n$ nor CCC_n contains an odd-length cycle of length $< n$; in particular, neither graph is pancyclic.*
(d) *If n is even, neither $\mathcal{B}_n$ nor CCC_n contains an odd-length cycle.*

Proof. **(a)** We begin our proof by concentrating on the case $c \equiv 0$ in the expression for the length m of the contained cycle. Hence, we wish to establish the containment in CCC_n of cycles of lengths $n2^k$ for all $k \in Z_{n+1} - \{1\}$.

We adapt the proof from [37] that establishes the hamiltonianicity of CCC_n, by establishing the hamiltonianicity of every graph in the following family of subgraphs of CCC_n. For $k \in Z_{n+1}$, the graph $CCC_n{}^{(k)}$ is the maximal connected component containing node $\vec{0}$ of the induced subgraph of CCC_n on the set of nodes

$$V_k =_{\text{def}} Z_n \times \{x0^{n-k} \ : \ x \in Z_2^k\}.$$

Note that $CCC_n{}^{(k)}$ is obtained from CCC_n by deleting all level-edges of CCC_n at levels $k, k+1, \ldots, n-1$ and then deleting all nodes and edges that are no longer accessible from node $\vec{0}$; in particular, $CCC_n{}^{(0)}$ is (isomorphic to) the n-node cycle C_n, and $CCC_n{}^{(n)}$ is identical to CCC_n. We let $CCC_n{}^{(k)}$ inherit a level structure in the natural way from CCC_n.

We establish the hamiltonianicity of every graph $CCC_n{}^{(k)}$ by induction on k, with two base cases. The base case $CCC_n{}^{(0)}$ will yield the desired result for all even values of k; the base case $CCC_n{}^{(3)}$ will yield the desired result for all odd values of k. This unexpected need for two base cases arises from the fact that our induction is extended by successively incrementing k by 2, as we see now.

Lemma 6.1 *For all $k \in Z_{n-1} - \{1\}$, if $CCC_n{}^{(k)}$ is hamiltonian, then so also is $CCC_n{}^{(k+2)}$.*

Proof. Assume for induction that we are given a hamiltonian cycle C in $CCC_n{}^{(k)}$. Let the nodes of $CCC_n{}^{(k)}$ be numbered with the integers Z_{n2^k} in an order consistent with the cyclic order of C. We extend the induction by traversing the hamiltonian cycle in $CCC_n{}^{(k)}$ repeatedly. As an aid in describing the multiple traversals, we say that a traversal *proceeds up* the cycle when it proceeds along the cycle in the order $j, j+1, j+2, \ldots$ and that the traversal *proceeds down* the cycle when it proceeds along the cycle in the order $j, j-1, j-2, \ldots$, all addition being modulo $n2^k$.

Implicit in the formula for pruning CCC_n to produce $CCC_n{}^{(k)}$ is the fact that one can construct $CCC_n{}^{(k+2)}$ by taking four copies of $CCC_n{}^{(k)}$, call them Copies 00, 10, 01, and 11, and interconnecting them so as to obtain a copy of $CCC_n{}^{(k+2)}$. The interconnection begins with a renaming of the PWL strings of the nodes of $CCC_n{}^{(k)}$, as indicated in the following table.

In ${CCC_n}^{(k)}$	In Copy 00	In Copy 10	In Copy 01	In Copy 11
$x0^{n-k}$	$x0^{n-k}$	$x100^{n-k-2}$	$x010^{n-k-2}$	$x110^{n-k-2}$

Now interconnect the four copies by adding to them the level-k and level-$(k+1)$ edges of CCC_n in just the way that makes the resulting graph isomorphic to ${CCC_n}^{(k+2)}$.

One can now trace out a hamiltonian cycle in ${CCC_n}^{(k+2)}$, as follows. We refer freely to the four copies of ${CCC_n}^{(k)}$ that comprise ${CCC_n}^{(k+2)}$.

1. Start at node $(k,\ \vec{0})$ in Copy 00 of ${CCC_n}^{(k)}$, and proceed up its hamiltonian cycle until node $(k+1,\ \vec{0})$.

2. Cross from node $(k+1,\ \vec{0})$ in Copy 00 to node $(k+1,\ \vec{0}01\vec{0})$ in Copy 01.

3. Starting at node $(k+1,\ \vec{0}01\vec{0})$ in Copy 01, proceed down its hamiltonian cycle until node $(k,\ \vec{0}01\vec{0})$.

4. Cross from node $(k,\ \vec{0}01\vec{0})$ to node $(k,\ \vec{0}11\vec{0})$ in Copy 11.

5. Starting at node $(k,\ \vec{0}11\vec{0})$ in Copy 11, proceed up its hamiltonian cycle until node $(k+1,\ \vec{0}11\vec{0})$.

6. Cross from node $(k+1,\ \vec{0}11\vec{0})$ to node $(k+1,\ \vec{0}10\vec{0})$ in Copy 10.

7. Starting at node $(k+1,\ \vec{0}10\vec{0})$ in Copy 10, proceed down its hamiltonian cycle until node $(k,\ \vec{0}10\vec{0})$.

8. Cross from node $(k,\ \vec{0}10\vec{0})$ to node $(k,\ \vec{0})$ in Copy 00.

We claim that the above procedure does, indeed, specify a walk within ${CCC_n}^{(k+2)}$, i.e., that every prescribed step of the walk crosses just one edge of the graph. The one facet of this claim that is not completely evident resides in our procedure's implicit exploitation of the following fact.

Fact 6.1 *For all k, every pair of nodes, $(\ell,\ x)$ and $(\ell+1,\ x)$, where $\ell \in Z_n - Z_k$ and $x \in Z_2^n$, appear consecutively in the hamiltonian cycle for ${CCC_n}^{(k)}$ that our procedure produces.*

Fact 6.1 assures us that the pairs of nodes, $(\ell,\ x)$ and $(\ell+1,\ x)$, can be used to interconnect cycles in copies of ${CCC_n}^{(\ell)}$ in the way mandated by our procedure. One verifies the Fact by noting that at the point when our procedure produces a hamiltonian cycle for ${CC C_n}^{(k)}$, the nodes of interest are bivalent, hence must appear consecutively in the cycle.

The proof of the Lemma is now complete: the walk specified by our procedure interconnects the hamiltonian cycles in the four copies of ${CCC_n}^{(k)}$ in a way that yields a hamiltonian cycle in ${CCC_n}^{(k+2)}$. □-Lemma 6.1

The proof of part (a) is completed by establishing the base cases of the induction.

Lemma 6.2 *Both ${CCC_n}^{(0)}$ and ${CCC_n}^{(3)}$ are hamiltonian.*

Proof. A cycle in ${CCC_n}^{(0)}$. Because ${CCC_n}^{(0)}$ is (isomorphic to) the n-node cycle C_n, its hamiltonianicity is immediate. Let us concentrate, therefore, on finding a hamiltonian cycle in ${CCC_n}^{(3)}$.

A cycle in ${CCC_n}^{(3)}$. We produce a hamiltonian cycle in ${CCC_n}^{(3)}$ from copies of the hamiltonian cycle in ${CCC_n}^{(0)}$ in much the same way that we produced a hamiltonian cycle in ${CCC_n}^{(k+2)}$ from copies of a hamiltonian cycle in ${CCC_n}^{(k)}$ in Lemma 6.1, except that we need eight copies of the "seed" cycle here, as opposed to the four copies that sufficed there.

Let us take eight copies of the hamiltonian cycle in ${CCC_n}^{(0)}$, call them Copies 000, 001, ..., 111. Note that the nodes of the cycle comprise the set $\{(\ell, \vec{0}) \ : \ \ell \in Z_n\}$. Relabel the PWL strings in all copies of the cycle so that the nodes of Copy $\alpha\beta\gamma$ of the cycle ($\alpha, \beta, \gamma \in Z_2$) comprise the set $\{(\ell, \alpha\beta\gamma\vec{0}) \ : \ \ell \in Z_n\}$. Under this node-labeling, the node-set of ${CCC_n}^{(3)}$ is just the union of the node-sets of the eight cycles, and the edges of ${CCC_n}^{(3)}$ are the union of the edges of the cycles, plus the intra-level edges on levels 0, 1, 2 of CCC_n. Using the same notion of traversing a cycle by proceeding *up* the cycle or *down* the cycle as we used in Lemma 6.1, we can now specify a hamiltonian cycle in ${CCC_n}^{(3)}$, as follows.

1. Start at node $(1,\ 000\vec{0})$ and proceed down the cycle, until node $(2,\ 000\vec{0})$.
2. Cross from node $(2,\ 000\vec{0})$ to node $(2,\ 001\vec{0})$ in Copy 001.
3. Proceed from node $(2,\ 001\vec{0})$ in Copy 001 up the cycle, until node $(1,\ 001\vec{0})$.
4. Cross from node $(1,\ 001\vec{0})$ to node $(1,\ 011\vec{0})$ in Copy 011.
5. Proceed from node $(1,\ 011\vec{0})$ in Copy 011 up the cycle, until node $(0,\ 011\vec{0})$.
6. Cross from node $(0,\ 011\vec{0})$ to node $(0,\ 111\vec{0})$ in Copy 111.
7. Proceed from node $(0,\ 111\vec{0})$ in Copy 111 down the cycle, until node $(1,\ 111\vec{0})$.

8. Cross from node $(1,\ 111\vec{0})$ to node $(1,\ 101\vec{0})$ in Copy 101.

9. Proceed from node $(1,\ 101\vec{0})$ in Copy 101 down the cycle, until node $(2,\ 101\vec{0})$.

10. Cross from node $(2,\ 101\vec{0})$ to node $(2,\ 100\vec{0})$ in Copy 100.

11. Proceed from node $(2,\ 100\vec{0})$ in Copy 100 up the cycle, until node $(1,\ 100\vec{0})$.

12. Cross from node $(1,\ 100\vec{0})$ to node $(1,\ 110\vec{0})$ in Copy 110.

13. Proceed from node $(1,\ 110\vec{0})$ in Copy 110 up the cycle, until node $(0,\ 110\vec{0})$.

14. Cross from node $(0,\ 110\vec{0})$ to node $(0,\ 010\vec{0})$ in Copy 010.

15. Proceed from node $(0,\ 010\vec{0})$ in Copy 010 down the cycle, until node $(1,\ 010\vec{0})$.

16. Cross from node $(1,\ 010\vec{0})$ to node $(1,\ 000\vec{0})$ in Copy 000.

The described walk constitutes a hamiltonian cycle in ${CCC_n}^{(3)}$. The existence of this cycle establishes the base case of the induction. □-Lemma 6.2

Now we extend our result to the entire claimed set of contained cycles, by allowing the constant c to assume nonzero values. Note that whenever the walks that define the cycles of lengths $n2^k$ encounter a cycle of CCC_n defined by a given PWL string x, call the cycle $C(x)$, they enter $C(x)$ at some node $(\ell,\ x)$ and leave it from node $(\ell',\ x) \in \{(\ell+1,\ x),\ (\ell-1,\ x)\}$, after having traversed all $(n-1)$ edges of $C(x)$ *other than* the one that connects these two nodes. If we alter the walk so that it enters $C(x)$ at node $(\ell,\ x)$ and leaves it from node $(\ell',\ x)$ *one step later*, i.e., by traversing just the one edge of $C(x)$ that connects these two nodes, then the length of the entire traversed cycle is decreased by precisely $n-2$. Pruning the walk in this way for c PWL strings yields the generalized result.

(b) As in Part (a), we focus first on the case $c \equiv 0$.

The fact that $\mathcal{B}_n$ contains the $n2^k$-node cycle for $k \in Z_{n+1} - \{1\}$ follows from Part (a) coupled with an invocation of Proposition 6.1. We capture the case $k = 1$, thereby completing the proof of Part (b) for the case $c \equiv 0$, by an explicit construction. Using the same notions of proceeding *up* and *down* cycles of straight-edges as in Part (a), we describe a length-$2n$ cycle in $\mathcal{B}_n$ via the following walk in the graph.

1. Start at node $(1,\ \vec{0})$, and proceed up the cycle of nodes having PWL string $\vec{0}$ until node $(0,\ \vec{0})$.

2. Cross from node $(0,\ \vec{0})$ to node $(1,\ 1\vec{0})$.

3. Proceed from node $(1,\ 1\vec{0})$, and proceed up the cycle of nodes having PWL string $1\vec{0}$ until node $(0,\ 1\vec{0})$.

4. Cross from node $(0,\ 1\vec{0})$ to node $(1,\ \vec{0})$.

The described walk constitutes a length-$2n$ cycle in $\mathcal{B}_n$.

We now extend the result to the complete range of claimed cycle lengths, by admitting nonzero values of the constant c, by using just one edge of an encountered PWL-cycle, as described in the proof of Part (a).

The special case of this result that establishes the hamiltonianicity of $\mathcal{B}_n$ is strengthened in [1], where it is shown that $\mathcal{B}_n$, when viewed as a digraph, contains a *directed* hamiltonian cycle.

(c) Any such short cycle would, in fact, be embedded in a leveled—hence, bipartite—subgraph $\mathcal{B}$ of $\mathcal{B}_n$ comprising some $n-1$ levels of $\mathcal{B}_n$. Since $\mathcal{B}$ is bipartite, it contains no odd-length cycle as a subgraph.

(d) When n is even, $\mathcal{B}_n$ is bipartite. A valid 2-coloring of $\mathcal{B}_n$ results from labeling each node of $\mathcal{B}_n$ with the parity of its level. □

7 X-Trees

We turn now to a family of graphs that can be viewed as a compromise between the logarithmic diameter, yet excessively sparse structure, of trees, and the rich interconnections with efficient layout, yet large diameter, of meshes [13], [22].

The **height-h complete binary tree $\mathcal{T}_h$** has node-set

$$\mathrm{N}(\mathcal{T}_h) = \bigcup_{i=0}^{h} Z_2^i,$$

the set of binary strings of length at most h; it has an edge between every pair of nodes x and $x\beta$, where $x \in \bigcup_{i=0}^{h-1} Z_2^i$ is some binary string of length $< h$, and $\beta \in Z_2$ (so is a bit, or, length-1 string). One conventionally partitions the nodes of $\mathcal{T}_h$ into *levels* by length; the *root* of $\mathcal{T}_h$ is the unique node of length 0; the leaves of $\mathcal{T}_h$ are the 2^h nodes of length h. Thus, $\mathcal{T}_h$ has $2^{h+1}-1$ nodes and (as with all trees) one fewer edge than nodes.
The **height-h X-tree $\mathcal{X}_h$** is obtained from the height-h complete binary tree $\mathcal{T}_h$ by adding edges that create a path along each level of $\mathcal{T}_h$, with the nodes occurring *in lexicographic order.*

Despite their obvious tree-like structure, X-trees also enjoy a rich cycle structure.

Theorem 7.1 *For all h, the height-h X-tree $\mathcal{X}_h$ is pancyclic.*

Proof. The proof proceeds by induction on the height of the X-tree. We need a two-element base for the induction in order to capture both odd and even heights, because the induction is extended by increasing the height h by 2. The structure of the cycles we form demands a property stronger than pancyclicity, which we call *Property X*.

The X-tree $\mathcal{X}_h$ has *Property X* if, for all $3 \leq k \leq 2^{h+1} - 1$, it contains a length-$k$ cycle that passes through two *adjacent* nodes on level h.

We shall prove that every X-tree has Property X. We begin with the base cases for our induction.

Lemma 7.1 *The height-1 X-tree $\mathcal{X}_1$ and the height-2 X-tree $\mathcal{X}_2$ have Property X.*

Proof. The results being trivial for $\mathcal{X}_1$, which is a 3-cycle, we concentrate on $\mathcal{X}_2$.

3-cycle. $\mathcal{X}_1$ contains the 3-cycle

$$0 \leftrightarrow 01 \leftrightarrow 00 \leftrightarrow 0$$

4-cycle. $\mathcal{X}_2$ contains the 4-cycle

$$0 \leftrightarrow 1 \leftrightarrow 10 \leftrightarrow 01 \leftrightarrow 0$$

5-cycle. $\mathcal{X}_2$ contains the 5-cycle

$$0 \leftrightarrow 1 \leftrightarrow 10 \leftrightarrow 01 \leftrightarrow 00 \leftrightarrow 0$$

6-cycle. $\mathcal{X}_2$ contains the 6-cycle

$$0 \leftrightarrow 1 \leftrightarrow 11 \leftrightarrow 10 \leftrightarrow 01 \leftrightarrow 00 \leftrightarrow 0$$

7-cycle. $\mathcal{X}_2$ contains the (hamiltonian) 7-cycle

$$\lambda \leftrightarrow 1 \leftrightarrow 11 \leftrightarrow 10 \leftrightarrow 01 \leftrightarrow 00 \leftrightarrow 0 \leftrightarrow \lambda$$

□-Lemma 7.1

We now assume that the X-tree $\mathcal{X}_h$ of height h enjoys Property X and infer that the X-tree $\mathcal{X}_{h+2}$ must share this Property.

Lemma 7.2 *If the height-h X-tree $\mathcal{X}_h$ has Property X, then so also does the height-$(h+2)$ X-tree $\mathcal{X}_{h+2}$.*

Proof. We focus first on cycles of lengths $3 \leq \ell \leq 2^{h+2} + 2^{h+1}$, and show that we can form such cycles in $\mathcal{X}_{h+2}$ using just the two highest-numbered levels of $\mathcal{X}_{h+2}$, namely, the 2^{h+1}-node level $h+1$ and the 2^{h+2}-node level $h+2$.

Pick any integer $1 \leq k \leq 2^{h+1}$.

- To form a cycle of length $3k$ in $\mathcal{X}_{h+2}$,
  1. follow the rightward length-k path along level $h+1$ of $\mathcal{X}_{h+2}$, starting at the leftmost node 0^{h+1}, and proceeding thence to node 0^h1, $0^{h-1}10$, and so on, until one reaches the kth node, call it x,
  2. proceed from node x to node $x1$ in level $h+2$,
  3. proceed leftward along level $h+2$, from node $x1$, to node $x0$, and so on, until one reaches the leftmost node 0^{h+2},
  4. proceed from node 0^{h+2} to node 0^{h+1}.
- To form a cycle of length $3k-1$ in $\mathcal{X}_{h+2}$, amend the length-$3k$ cycle just described, by proceeding from node x on level $h+1$ to node $x0$ on level $h+2$, bypassing node $x1$.
- To form a cycle of length $3k-2$ in $\mathcal{X}_{h+2}$, amend the length-$(3k-1)$ cycle just described, by proceeding from node $0^{h+1}1$ on level $h+2$ to node 0^{h+1}, bypassing node 0^{h+2}.

Using this recipe, one finds cycles of all lengths $3 \leq \ell \leq 2^{h+2} + 2^{h+1}$ in $\mathcal{X}_{h+2}$.

In order to form the cycle of length $2^{h+2} + 2^{h+1} + 1$ in $\mathcal{X}_{h+2}$, we take the cycle of length $2^{h+2} + 2^{h+1}$ and amend it by interposing the path from node 0^{h+1} on level $h+1$ to node 0^h on level h to node 0^h1 on level $h+1$ between nodes 0^{h+1} and 0^h1, in stage 1 of the cycle.

In order to form the cycle of length $2^{h+2} + 2^{h+1} + 2$ in $\mathcal{X}_{h+2}$, we take the cycle of length $2^{h+2} + 2^{h+1} + 1$ and amend it by interposing the path from node 1^h0 on level $h+1$ to node 1^h on level h to node 1^{h+1} on level $h+1$ between nodes 1^h0 and 1^{h+1}, in stage 1 of the cycle.

We form cycles of all remaining lengths $2^{h+2} + 2^{h+1} + 3 \leq \ell \leq 2^{h+3} - 1$ in $\mathcal{X}_{h+2}$, by invoking the inductive hypothesis that $\mathcal{X}_h$ enjoys Property X and noting that the induced subgraph of $\mathcal{X}_{h+2}$ on levels $0, 1, \ldots, h$ is isomorphic to $\mathcal{X}_h$.

In order to form a cycle of length $2^{h+2} + 2^{h+1} + c$ in $\mathcal{X}_{h+2}$, $3 \leq c \leq 2^{h+1} - 1$, we take the cycle of length $2^{h+2} + 2^{h+1}$ that involves only levels $h+1$ and $h+2$ of $\mathcal{X}_{h+2}$, we take a cycle of length c within levels $0, 1, \ldots, h$ of $\mathcal{X}_{h+2}$ that involves two adjacent nodes on level h—call these nodes x and y from left to right, and we join these two cycles into a single cycle by adding the edges $\{x, x1\}$ and $\{y, y0\}$ and deleting the edges $\{x, y\}$ and $\{x1, y0\}$.

We have now formed cycles in $\mathcal{X}_{h+2}$ of all lengths $3 \leq \ell \leq 2^{h+3} - 1$, and every cycle involves at least two adjacent nodes on level $h+2$ of $\mathcal{X}_{h+2}$. This extends the induction and completes the proof. □-Lemma 7.2

The proof is complete. □

8 De Bruijn Graphs

The next family of graphs we study epitomize the "shuffle-oriented" graphs. These graphs are known to be able to emulate the much larger butterfly graph and its "butterfly-oriented" relatives efficiently on a large class of computations [3], [39]; hence, these graphs have been widely proposed as interconnection networks for parallel architectures [6], [31], [33], [36].

The **order-n de Bruijn graph** $\mathcal{D}_n$ is usually presented as a *directed* graph. The digraph $\mathcal{D}_n$ has node-set $\mathrm{N}(\mathcal{D}_n) = Z_2^n$; its arcs lead every node βx, where $x \in Z_2^{n-1}$ and $\beta \in Z_2$ to nodes $x\beta$ and $x\bar{\beta}$. Because $\mathcal{D}_n$ has 2^n nodes, each of indegree and outdegree 2, it has 2^{n+1} arcs.

De Bruijn graphs enjoy the strongest cycle structure of any of the sparse graphs we consider here in that their pancyclicity persists even when they are viewed as directed graphs. The following theorem originates in [43]; the proof we present derives from [21], wherein the result is generalized to de Bruijn graphs of arbitrary base.

Theorem 8.1 (Cycle-Embeddings for $\mathcal{D}_n$)
For all n, the order-n de Bruijn digraph $\mathcal{D}_n$ is di-pancyclic.

Proof. We remark that the result remains true for de Bruijn digraphs of arbitrary bases [21]. We restrict attention here to the base 2 case, because this case exposes all of the ideas essential to prove the stronger result, in a clerically simpler setting.

We proceed by induction on the order n of $\mathcal{D}_n$.

The result being easily verified for de Bruijn digraphs of small order (say, orders 1 and 2), let us assume for induction that the result holds for the order-n de Bruijn digraph $\mathcal{D}_n$.

In order to extend the result to the order-$(n+1)$ de Bruijn digraph $\mathcal{D}_{n+1}$, we argue separately about the presence of *small* cycles, i.e., those of length $\leq 2^n$, and of *large* cycles, i.e., those of length between $2^n + 1$ and 2^{n+1}.

Case 1: *Small Cycles.*

The presence of all small cycles as subgraphs of $\mathcal{D}_{n+1}$ is an easy consequence of the intimate connection between de Bruijn graphs of successive orders.

Lemma 8.1 *For all n, the order-$(n+1)$ de Bruijn digraph $\mathcal{D}_{n+1}$ is the line-graph of the order-n de Bruijn digraph $\mathcal{D}_n$.*

Proof. Associate node $\beta x \gamma \in \mathbf{N}(\mathcal{D}_{n+1})$, where $\beta, \gamma \in Z_2$ and $x \in Z_2^{n-1}$, with the arc

$$(\beta x,\ x\gamma) \in \mathbf{E}(\mathcal{D}_n).$$

Note first that each arc of $\mathcal{D}_{n+1}$ has the form

$$(\delta y \epsilon,\ y \epsilon \phi)$$

for some $\delta, \epsilon, \phi \in Z_2$ and some $y \in Z_2^{n-1}$. By our association of nodes of $\mathcal{D}_{n+1}$ with arcs of $\mathcal{D}_n$, this arc of $\mathcal{D}_{n+1}$ does indeed correspond to two successive arcs of $\mathcal{D}_n$, namely, one that is incident *into* node $y\epsilon \in \mathbf{N}(\mathcal{D}_n)$, followed by one that is incident *out of* node $y\epsilon$.

Note next that, given any two successive arcs of $\mathcal{D}_n$, say

$$(\rho\sigma z,\ \sigma z \tau)$$

and

$$(\sigma z \tau,\ z \tau \xi),$$

where $\rho, \sigma, \tau, \xi \in Z_2$ and $z \in Z_2^{n-2}$, there is, indeed, an arc of $\mathcal{D}_{n+1}$ of the form

$$(\rho\sigma z \tau,\ \sigma z \tau \xi).$$

The desired correspondence is verified. □-Lemma 8.1

Since the line-graph of a k-node cycle is again a k-node cycle, the assumed pancyclicity of $\mathcal{D}_n$ joins with Lemma 8.1 to verify that $\mathcal{D}_{n+1}$ contains all small cycles as subgraphs.

Case 2: *Large Cycles.*

Let us focus on the integer $m = 2^n + k$, where $0 < k \leq 2^n$, and let us verify that $\mathcal{D}_{n+1}$ contains a cycle having m nodes.

By Case 1 and our inductive hypothesis, $\mathcal{D}_{n+1}$ contains a cycle having $M = 2^{n+1} - m = 2^n - k$ nodes, hence M arcs. By Lemma 8.1, this cycle in $\mathcal{D}_{n+1}$ implies that $\mathcal{D}_n$ contains a connected *eulerian*[5] subgraph having M arcs. We claim that $\mathcal{D}_n$ *also* contains a connected eulerian subgraph having m arcs. By Lemma 8.1, the existence of this latter subgraph would establish the existence of a cycle in $\mathcal{D}_{n+1}$ having $m = 2^n + k$ nodes. Hence, the following lemma suffices to complete the proof of the Theorem.

Lemma 8.2 *If $\mathcal{D}_n$ contains a connected eulerian subgraph $\mathcal{G}$ having p arcs ($0 \leq p \leq 2^{n+1}$), then it also contains such a subgraph having $2^{n+1} - p$ arcs.*

Proof. Let $\mathcal{D}_n$ contain a p-arc connected eulerian subgraph $\mathcal{G}$. Since $\mathcal{D}_n$ is eulerian, and since each node of $\mathcal{G}$ has equal indegree and outdegree (in $\mathcal{G}$ as well as in $\mathcal{D}_n$), if we remove from $\mathcal{D}_n$ the p arcs of $\mathcal{G}$, then we are left with a (not necessarily connected) eulerian subgraph $\mathcal{H}$ of $\mathcal{D}_n$ that has the desired number of arcs. Let $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_r$ denote all those maximal connected components of $\mathcal{H}$ that are *nontrivial* in that they contain more than one node each (hence, contain at least one arc each); each $\mathcal{F}_i$ is, of course, connected and eulerian.

If $r = 1$, then $\mathcal{H}$ is the sought connected eulerian subgraph of $\mathcal{D}_n$, so we are done. Assume, therefore, that $r > 1$.

Because $\mathcal{D}_n$ is connected, and all of its arcs reside either in $\mathcal{G}$ or in $\mathcal{H}$, there must be some arc of $\mathcal{G}$ of the form (u, v), where $u \in \mathrm{N}(\mathcal{F}_i)$ and $v \in \mathrm{N}(\mathcal{F}_j)$ for $i \neq j$. The existence of this arc implies that node u has outdegree ≤ 1 in $\mathcal{H}$, even though it has outdegree 2 in $\mathcal{D}_n$. Because $\mathcal{H}$ is eulerian (so node u has equal indegree and outdegree in $\mathcal{H}$) and has at least one arc and is connected (so node u cannot be isolated in $\mathcal{H}$), node u must have outdegree *exactly* 1 in $\mathcal{H}$. Therefore, there must be an arc (u, w) in $\mathcal{F}_i$ for some node $w \in \mathrm{N}(\mathcal{F}_i)$. By similar reasoning (using indegrees instead of outdegrees) there must be an arc (t, v) in $\mathcal{F}_j$ for some node $t \in \mathrm{N}(\mathcal{F}_j)$. Since arcs (u, v), (u, w), and (t, v) all reside in $\mathcal{D}_n$, there must exist strings $x \in Z_2^{n-1}$ and $\beta, \gamma, \delta, \epsilon \in Z_2$ such that

$$\begin{aligned} t &= \beta x \\ u &= \gamma x \\ v &= x\delta \\ w &= x\epsilon. \end{aligned}$$

It follows that there is an arc (t, w) in $\mathcal{G}$: this arc exists in $\mathcal{D}_n$ by definition, and it cannot reside in either $\mathcal{F}_i$ or $\mathcal{F}_j$, because it connects these two components which are disconnected in $\mathcal{H}$.

[5] Digraph $\mathcal{G}$ is *eulerian* if each node $v \in \mathrm{N}(\mathcal{G})$ has indegree(v) = outdegree(v). Each connected component of $\mathcal{G}$ then admits a directed walk that crosses each arc of the component precisely once.

We now transform the subgraph $\mathcal{H}$ in the following way. We remove from $\mathcal{H}$ the arcs (u, w) from $\mathcal{F}_i$ and (t, v) from $\mathcal{F}_j$, and we add in their places the arcs (t, w) and (u, v). The digraph $\mathcal{H}'$ so produced

- has the same number of arcs as does $\mathcal{H}$;
- is eulerian, because we just exchanged one arc into each of nodes v and w for another and made a similar switch for arcs out of each of nodes t and u;
- is connected, because each of $\mathcal{F}_i$ and $\mathcal{F}_j$, being eulerian, admit directed walks that cross each arc precisely once, and our exchanged arcs connect these two directed walks into a composite directed walk through the new component;
- has one fewer maximal connected nontrivial component than does $\mathcal{H}$.

If we now iterate the just-described transformation, each iteration yields an eulerian subgraph of $\mathcal{D}_n$ having the desired number of arcs and having one fewer nontrivial connected component than its predecessor. After $r - 1$ iterations, therefore, we achieve the desired connected eulerian subgraph of $\mathcal{D}_n$. □-Lemma 8.2

The m-arc connected eulerian subgraph of $\mathcal{D}_n$ guaranteed by Lemma 8.2 implies the existence of an m-node cycle in $\mathcal{D}_{n+1}$. Since k, hence m, was arbitrary, the proof of the Theorem is complete. □

9 Product-Shuffle Graphs

The final family of graphs we study has never been seriously proposed as an interconnection network for parallel architectures. However, it has been shown to combine important structural characteristics of both "butterfly-oriented" and "shuffle-oriented" graphs [27], while simultaneously enjoying the direct-product structure that has been shown to have significant algorithmic consequences [2], [4]. For our purposes the importance of the family is that it demonstrates how pancyclicity interacts with the direct-product operation.

The $m \times n$ **product-shuffle graph** is the direct product of de Bruijn graphs $\mathcal{PS}_{m,n} = \mathcal{D}_m \times \mathcal{D}_n$. Product-shuffle graphs enjoy a version of the pancyclicity of de Bruijn graphs that is weakened in two respects. First, we establish the pancyclicity only of the undirected version of product-shuffle graphs—note, therefore, that we view $\mathcal{D}_n$ as an *undirected* graph in this section; second, we must exclude the case $m = n = 1$. The reader should note that the property of de Bruijn graphs exposed in Lemma 9.1 plays a crucial role in our proof of the pancyclicity of product-shuffle graphs; the fact that $\mathcal{PS}_{m,n}$ is a direct product of pancyclic graphs does not guarantee its pancyclicity. The following theorem originates in [27].

Theorem 9.1 *For all m, n except for $m = n = 1$, the product-shuffle graph $\mathcal{PS}_{m,n}$ is pancyclic.*

Proof. Let us begin by noting that $\mathcal{PS}_{1,1}$ is (essentially) a 4-cycle, whence its exclusion from the Theorem.

For any choice of m, n other than $m = n = 1$, and for any integer $1 \le c \le 2^{m+n}$, we show algorithmically that the cycle $\mathcal{C}_c$ is a subgraph of $\mathcal{PS}_{m,n}$. Our algorithm builds on the fact that Theorem 8.1 provides an algorithm for producing cycles in de Bruijn graphs.

Assume, with no loss of generality, that $m \le n$ (or else, interchange the roles of m and n in what follows). If the desired cycle-length c satisfies $1 \le c \le 2^n$, then $\mathcal{C}_c$ is a subgraph of $\mathcal{PS}_{m,n}$, by Theorem 8.1. Let us restrict attention, therefore, to values of c in the range $2^n < c \le 2^{m+n}$, in which case we must have $m > 0$.

Now, every integer c in the indicated range admits a unique representation in the form

$$c = a2^n + b$$

with $0 < a \le 2^m$ and $0 \le b < 2^n$. The overall strategy of our algorithm is to "hook together" hamiltonian cycles from a of the 2^m copies of $\mathcal{D}_n$ that comprise $\mathcal{PS}_{m,n}$, together with a length-b cycle from one additional copy of $\mathcal{D}_n$ whenever $b > 0$. (In fact, technical difficulties in "hooking up" these cycles will cause us to deviate from this strategy slightly.) To the end of implementing this strategy, we invoke Theorem 8.1 to find a length-d cycle in $\mathcal{D}_m$, where

$$d = \begin{cases} a & \text{if } b = 0 \\ a+1 & \text{if } b > 0, \end{cases}$$

and we use this cycle in the natural way to select and order d "consecutive" copies of $\mathcal{D}_n$, from the 2^m copies that comprise $\mathcal{PS}_{m,n}$; call the ordered copies $\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(d-1)}$.

We describe the mechanism for "hooking the cycles together" via an analysis of cases.

Case 1: *$b = 0$, so $d = a$ and $a > 1$.*

This is the easiest case, since we have only to "hook together" a set $C_0, C_1, \ldots, C_{a-1}$ of cycles, each C_i being a copy within $\mathcal{D}^{(i)}$ of a hamiltonian cycle C of $\mathcal{D}_n$. We start by selecting any two *independent* edges (x, y) and (u, v) of $\mathcal{D}_n$,[6] that both lie on the cycle C; since $n \ge 2$, we are sure that these edges exist. Next, we let x_i, y_i, u_i, v_i $(0 \le i < a)$ denote the instances of the nodes x, y, u, v, respectively, in copy $\mathcal{D}^{(i)}$ of $\mathcal{D}_n$. Assume that

[6] "Independence" implies that $\{x, y\} \cap \{u, v\} = \emptyset$.

the nodes x, y, u, v lie in (say, for definiteness) clockwise order around the cycle C in $\mathcal{D}_n$, so that each cycle C_i has the form

$$y_i, P_i, u_i, v_i, Q_i, x_i$$

where P_i and Q_i are the intermediate paths that define the cycle.

We are now ready to find a length-c cycle in $\mathcal{PS}_{m,n}$.

1. Trace the cycle C_0 in $\mathcal{D}^{(0)}$ in clockwise order, from node y_0 to node x_0, leaving out the edge that connects the two nodes.

2. Trace the following path to complete the cycle:

$$\begin{array}{l} x_0 \leftrightarrow x_1 \leftrightarrow Q_1 \leftrightarrow v_1 \leftrightarrow v_2 \leftrightarrow Q_2 \leftrightarrow x_2 \leftrightarrow x_3 \leftrightarrow Q_3 \leftrightarrow v_3 \leftrightarrow \cdots \\ \leftrightarrow q_{a-1} \leftrightarrow Q_{a-1} \leftrightarrow r_{a-1} \leftrightarrow s_{a-1} \leftrightarrow P_{a-1} \leftrightarrow t_{a-1} \leftrightarrow \cdots \\ u_3 \leftrightarrow P_3 \leftrightarrow y_3 \leftrightarrow y_2 \leftrightarrow P_2 \leftrightarrow u_2 \leftrightarrow u_1 \leftrightarrow P_1 \leftrightarrow y_1 \leftrightarrow y_0 \end{array}$$

   where

$$q, r, s, t = \begin{cases} x, v, u, y & \text{respectively, if } a \text{ is even} \\ v, x, y, u & \text{respectively, if } a \text{ is odd.} \end{cases}$$

The paths P_i and Q_i and the edges (x_i, y_i) and (u_i, v_i) come from the copies of $\mathcal{D}_n$, while the edges (x_i, x_{i+1}), (y_i, y_{i+1}), (u_i, u_{i+1}), and (v_i, v_{i+1}) come from the copy of $\mathcal{D}_m$ we used to order the copies of $\mathcal{D}_n$.

Case 2: *$0 < b < 2^n$, so $d = a + 1$ and $0 < a < 2^m$.*

The added challenge in this case arises from the need to append a cycle of length b to the chain of a hamiltonian cycles created in Case 1. The mechanism we use depends on the value of b.

2.1: $b \geq 3$

We must alter the procedure of Case 1 in two ways: we must find a copy of a length-b cycle in copy $\mathcal{D}^{(a)}$ of $\mathcal{D}_n$, and we must ensure that we can "hook" this new cycle to the chain of hamiltonian cycles. The first of these tasks is immediate, by Theorem 8.1; let us call the length-b cycle B. In order to accomplish the second task, we invoke the strong property of $\mathcal{D}_n$ exposed in the following lemma.

Lemma 9.1 *For any path $x \leftrightarrow y \leftrightarrow z$ in $\mathcal{D}_n$ involving three distinct nodes, there is a hamiltonian cycle of $\mathcal{D}_n$ that contains either the edge (x, y) or the edge (y, z).*

Proof. The result is true by inspection when $n = 2$. When $n > 2$, the result is a consequence of the following facts about de Bruijn graphs.

Fact 9.1 *One can construct a hamiltonian cycle in $\mathcal{D}_n$ from any eulerian cycle in $\mathcal{D}_{n-1}$.*

Fact 9.2 *Any* 3-node *path*

$$x \leftrightarrow y \leftrightarrow z$$

in $\mathcal{D}_n$ results from a 3-edge *path*

$$W \leftrightarrow X \leftrightarrow Y \leftrightarrow Z \tag{1}$$

in $\mathcal{D}_{n-1}$.

Both Facts 9.1 and 9.2 are consequences of $\mathcal{D}_n$'s being the line-graph of $\mathcal{D}_{n-1}$ (Lemma 8.1).

Fact 9.3 *Given any length-2 path π in $\mathcal{D}_n$ whose removal does not disconnect $\mathcal{D}_n$, one can construct an eulerian cycle in $\mathcal{D}_n$ which contains π.*

Fact 9.3 follows from the algorithm for constructing an eulerian cycle in an eulerian graph (cf. [23]), which allows one to choose edges leaving nodes at random among the as-yet unused edges.

The final Fact we need is a simple structural property of $\mathcal{D}_n$.

Fact 9.4 *The only length-2 paths whose removal disconnect $\mathcal{D}_n$ are the paths both of whose edges are incident to either node $\vec{0}$ or node $\vec{1}$.*

Since at most two of the edges of path 1 can both be incident to either node $\vec{0}$ or node $\vec{1}$ in $\mathcal{D}_{n-1}$, it follows from Facts 9.3 and 9.4 that there is an eulerian cycle in $\mathcal{D}_{n-1}$ passing through either the path

$$W \leftrightarrow X \leftrightarrow Y$$

or the path

$$X \leftrightarrow Y \leftrightarrow Z.$$

Fact 9.1 assures us that, in the former case, there is a hamiltonian cycle in $\mathcal{D}_n$ passing through edge (x, y), while in the latter case, there is a hamiltonian cycle passing through edge (y, z). □-Lemma 9.1

By dint of Lemma 9.1 and the fact that $b \geq 3$, we can find an edge e of the length-b cycle B in $\mathcal{D}^{(a)}$, that lies on a hamiltonian cycle of $\mathcal{D}_n$. Let us choose edge e as the edge

(r, s) of Case 1. We then alter the trajectory of the length-c cycle after the initial path within $\mathcal{D}^{(a-1)}$, by replacing the length-1 path

$$r_{a-1} \leftrightarrow s_{a-1}$$

with the length-$(b+1)$ path

$$r_{a-1} \leftrightarrow r_a \leftrightarrow S \leftrightarrow s_a \leftrightarrow s_{a-1}$$

where S is the length-$(b-1)$ path within cycle B that connects nodes r_a and s_a in $\mathcal{D}^{(a)}$ once edge (r_a, s_a) is removed.

2.2: $b = 2$

We proceed exactly as in Case 1, except that we alter the trajectory of the length-$a2^n$ cycle of that Case by replacing the length-1 path

$$r_{a-1} \leftrightarrow s_{a-1}$$

with the length-3 path

$$r_{a-1} \leftrightarrow r_a \leftrightarrow s_a \leftrightarrow s_{a-1}$$

where r, s are as in Case 1.

2.3: $b = 1$

We branch immediately on the value of n.

2.3.1: When $n = 2$, we proceed exactly as in Case 1, until we have to deal with copy $\mathcal{D}^{(a-1)}$ of $\mathcal{D}_n$. At that point we replace the length-3 path

$$q_{a-1} \leftrightarrow r_{a-1} \leftrightarrow s_{a-1} \leftrightarrow t_{a-1}$$

from Case 1 with a length-4 path of one of the forms

$$q_{a-1} \leftrightarrow s_{a-1} \leftrightarrow s_a \leftrightarrow t_a \leftrightarrow t_{a-1}$$

or

$$q_{a-1} \leftrightarrow q_a \leftrightarrow s_a \leftrightarrow s_{a-1} \leftrightarrow t_{a-1}$$

within copies $\mathcal{D}^{(a-1)}$ and $\mathcal{D}^{(a)}$ of $\mathcal{D}_n$. One verifies readily that one of these paths exists.

2.3.2: When $n > 2$, we alter Case 1 by insisting that at least one of the independent edges (x, y) and (u, v) not touch either node $\vec{0}$ or node $\vec{1}$ of $\mathcal{D}_n$. (Note that this is impossible when $n = 2$.) Say, without loss of generality, that node $\vec{0}$ is not touched by either edge.

Having thus restricted the choice of these edges, we proceed exactly as in Case 2.b ($b = 2$), with the following exception. Once having found the cycle produced in Case 2.b (which has length $c+1$), we remove the instance of node $\vec{0}$ of $\mathcal{D}_n$ from whichever of P_{a-1} or Q_{a-1} contains an instance of $\vec{0}$. (One of them must, because of our restriction.) Since every hamiltonian cycle in $\mathcal{D}_n$ contains the path

$$1\vec{0} \leftrightarrow \vec{0} \leftrightarrow \vec{0}1,$$

the elision of node $\vec{0}$ does not cut our cycle: it just shortens it, as desired.

This case analysis completes the proof. □

Acknowledgments. The author thanks Fred Annexstein and Marc Baumslag for pointing out certain sources, and Bojana Obrenić for a careful reading of the manuscript.

References

[1] F.S. Annexstein and M. Baumslag (1988): Hamiltonian circuits in Cayley digraphs. Tech. Rpt. 88-40, Univ. Massachusetts; submitted for publication.

[2] F.S. Annexstein and M. Baumslag (1990): A unified approach to global permutation routing on parallel networks. *2nd ACM Symp. on Parallel Algorithms and Architectures*, 398-406.

[3] F.S. Annexstein, M. Baumslag, A.L. Rosenberg (1990): Group action graphs and parallel architectures. *SIAM J. Comput. 19*, 544-569.

[4] F.S. Annexstein, M. Baumslag, M.C. Herbordt, B. Obrenić, A.L. Rosenberg, C.C. Weems (1990): Achieving multigauge behavior in bit-serial SIMD architectures via emulation. *3rd IEEE Symp. on Frontiers of Massively Parallel Computation*, 186-195.

[5] M.J. Atallah and S.E. Hambrusch (1986): Solving tree problems on a mesh-connected processor array. *Inform. Computation 69*, 168-187.

[6] J.-C. Bermond and C. Peyrat (1989): The de Bruijn and Kautz networks: a competitor for the hypercube? In *Hypercube and Distributed Computers* (F. Andre and J.P. Verjus, eds.) North-Holland, Amsterdam, 279-293.

[7] S.N. Bhatt, F.R.K. Chung, J.-W. Hong, F.T. Leighton, B. Obrenić, A.L. Rosenberg, E.J. Schwabe (1991): Optimal emulations by butterfly-like networks. *J. ACM*, to appear.

[8] S.N. Bhatt, F.R.K. Chung, F.T. Leighton, A.L. Rosenberg (1991): Efficient embeddings of trees in hypercubes. *SIAM J. Comput.*, to appear.

[9] J.A. Bondy and U.S.R. Murty (1976): *Graph Theory with Applications.* North-Holland, New York.

[10] M.Y. Chan (1989): Embedding of d-dimensional grids into optimal hypercubes. *1st ACM Symp. on Parallel Algorithms and Architectures*, 52-57.

[11] T. Cormen, C.E. Leiserson, R.L. Rivest (1990): *Introduction to Algorithms.* MIT Press, Cambridge, Mass.

[12] W.J. Dally and C.L. Seitz (1986): The torus routing chip. *J. Distributed Systems 1*, 187-196.

[13] A.M. Despain and D.A. Patterson (1978): X-tree – a tree structured multiprocessor architecture. *5th Symp. on Computer Architecture*, 144-151.

[14] V. Faber (1991): Global communication algorithms for hypercubes and other Cayley coset graphs. *SIAM J. Discr. Math.*, to appear.

[15] R. Feldmann and W. Unger (1990): The cube-connected cycles network is a subgraph of the butterfly network. Typescript, Univ. Paderborn.

[16] H. Fleischner (1974): The square of every two-connected graph is hamiltonian. *J. Comb. Th. (B) 16*, 29-34.

[17] M.R. Garey and D.S. Johnson (1979): *Computers and Intractability.* W.H. Freeman and Co., San Francisco.

[18] D.S. Greenberg, L.S. Heath and A.L. Rosenberg (1990): Optimal embeddings of butterfly-like graphs in the hypercube. *Math. Syst. Th. 23*, 61-77.

[19] S.L. Johnsson and C.-T. Ho (1989): Optimum broadcasting and personalized communication in hypercubes. *IEEE Trans. Comp. 38*, 1249-1268.

[20] R.M. Karp (1972): Reducibility among combinatorial problems. In *Complexity of Computer Computations* (R.E. Miller and J.W Thatcher, eds.), Plenum Press, N.Y., pp. 85-103.

[21] A. Lempel (1971): m-ary closed sequences. *J. Comb. Th. (A) 10*, 253-258.

[22] B. Monien (1991): Simulating binary trees on X-trees. Typescript, Univ. Paderborn.

[23] O. Ore (1962): *Theory of Graphs.* American Math. Soc. Colloquium Publications, Vol. XXXVIII. American Mathematical Society, Providence, R. I.

[24] F.P. Preparata and J.E. Vuillemin (1981): The cube-connected cycles: a versatile network for parallel computation. *C. ACM 24*, 300-309.

[25] R.D. Rettberg (1986): Shared memory parallel processors: the Butterfly and the Monarch. *4th MIT Conf. on Advanced Research in VLSI.*

[26] A.L. Rosenberg (1981): Issues in the study of graph embeddings. In *Graph-Theoretic Concepts in Computer Science: Proceedings of the International Workshop WG80*, Bad Honnef, Germany (H. Noltemeier, ed.) *Lecture Notes in Computer Science 100*, Springer-Verlag, NY, 150-176.

[27] A.L. Rosenberg (1991): Product-shuffle networks: toward reconciling shuffles and butterflies. *Discr. Appl. Math.*, to appear.

[28] A.L. Rosenberg and L. Snyder (1978): Bounds on the costs of data encodings. *Math. Syst. Th. 12*, 9-39.

[29] Y. Saad and M.H. Schultz (1988): Topological properties of hypercubes. *IEEE Trans. Comp. 37*, 867-872.

[30] Y. Saad and M.H. Schultz (1989): Data communication in hypercubes. *J. Parallel Distr. Comput. 6*, 115-135.

[31] M.R. Samatham and D.K. Pradhan (1989): The de Bruijn multiprocessor network: a versatile parallel processing and sorting network for VLSI. *IEEE Trans. Comp. 38*, 567-581.

[32] E.J. Schwabe (1989): Normal hypercube algorithms can be simulated on a butterfly with only constant slowdown. *Inform. Proc. Let.*

[33] J.T. Schwartz (1980): Ultracomputers. *ACM Trans. Prog. Lang. 2*, 484-521.

[34] M. Sekanina (1960): On an ordering of the set of vertices of a connected graph. *Publ. Fac. Sci. Univ. Brno*, No. 412, 137-142.

[35] C. Stanfill (1987): Communications architecture in the Connection Machine system. Tech. Rpt. HA87-3, Thinking Machines Corp.

[36] H. Stone (1971): Parallel processing with the perfect shuffle. *IEEE Trans. Comp., C-20*, 153-161.

[37] R. Stong (1987): On Hamiltonian cycles in Cayley graphs of wreath products. *Discr. Math. 65*, 75-80.

[38] W.T. Trotter, Jr., and P. Erdös (1978): When the cartesian product of directed cycles is hamiltonian. *J. Graph Th. 2*, 137-142.

[39] J.D. Ullman (1984): *Computational Aspects of VLSI.* Computer Science Press, Rockville, Md.

[40] A. Wigderson (1982): The complexity of the hamiltonian circuit problem for maximal planar graphs. Princeton Univ. EECS Dept. Report 298.

[41] D. Witte (1982): On Hamiltonian circuits in Cayley diagrams. *Discrete Math. 38*, 99-108.

[42] D. Witte and D. Gallian (1984): A survey: Hamiltonian cycles in Cayley graphs. *Discrete Math. 51*, 293-304.

[43] M. Yoeli (1962): Binary ring sequences. *Amer. Math. Monthly 69*, 852-855.