One-to-many Embeddings of Hypercubes into Cayley Graphs Generated by Reversals

by

Linda Gardner[1], Zevi Miller[2], Dan Pritikin[2], I. Hal Sudborough[1]

[1] Computer Science Program, University of Texas at Dallas, Richardson, Texas 75083
[2] Dept. of Mathematics and Statistics, Miami University, Oxford, Ohio 45056

Abstract

Among Cayley graphs on the symmetric group, some that have noticeably low diameter relative to the degree of regularity are examples such as the "star" network and the "pancake" network, the latter a representative of a variety of Cayley graphs generated by reversals. These diameter and degree conditions make these graphs potential candidates for parallel computation networks. Thus it is natural to investigate how well they can simulate other standard parallel networks, in particular hypercubes. For this purpose, constructions have previously been given for low dilation embeddings of hypercubes into star networks. Developing this theme further, in this paper we construct especially low dilation maps (e.g. with dilation 1,2,3, or 4) of hypercubes into pancake networks and related Cayley graphs generated by reversals. Whereas obtaining such results by the use of "traditional" graph embeddings (i.e. one-to-one or many-to-one embeddings) is sometimes difficult or impossible, we achieve many of these results by using a nontraditional simulation technique known as a "one-to-many" graph embedding. That is, in such embeddings we allow each vertex in the guest (i.e. domain) graph to be associated with some nonempty subset of the vertex set of the host (i.e. range) graph, these subsets satisfying certain distance and connection requirements which make possible the simulation.

1. Introduction

1.1 Hypercubes and Pancake Networks:

Much recent attention has been given to n-dimensional interconnection networks having n! many processors, one for each permutation on n symbols [AK, BHOS, CB, FMC, HS, MPS1, MPS2, NSK]. Such networks, usually defined as Cayley graphs on the symmetric group of order n!, often have smaller diameter and fewer connections per processor than correspondingly large hypercubes. We consider the ability of some of these networks to efficiently simulate hypercubes. Specifically, we show how to embed hypercubes into them with very low dilation. We assume familiarity with basic concepts. Background not given here may be found in [L,MS1].

First we define the n-dimensional hypercube Q_n, $n \geq 1$, as follows. The vertex set $V(Q_n)$ of this graph is the set of all binary strings of length n. The edge set $E(Q_n)$ will be the collection of all unordered pairs xy, where $x,y \in V(Q_n)$ and x and y disagree in exactly one coordinate. We can view Q_n as the graphical Cartesian product $Q_n = Q_{n_1} \times Q_{n_2} \times \ldots \times Q_{n_m}$ where the n_i are any positive integers summing to n. Illustrations of Q_2 and Q_3 are given in Figure 2, where the vertices are indicated in the boxes, and the edges are indicated by lines joining appropriate pairs of boxed binary strings (the additional symbols in the drawing will be explained later). As a special case, we let Q_0 be the graph having one vertex and no edges.

Let Σ_n denote the group of all permutations on n symbols (with the operation of functional composition), i.e. the symmetric group of degree n. Let T_n be a set of generators for the group Σ_n. Then the Cayley graph determined by Σ_n and T_n, denoted by $G(\Sigma_n,T_n)$, is the directed graph whose vertex set is Σ_n, with two permutations/vertices α,β connected by a directed edge from α to β if and only if $\beta = \alpha \circ t$ for some generator $t \in T_n$. When the set T_n is closed under inverses this directed graph is symmetric, and thus each pair of oppositely oriented edges will be regarded as a single undirected edge, thereby allowing us to view $G(\Sigma_n,T_n)$ as an undirected graph.

We now introduce a family of Cayley graphs central to our study. For any integers $1 \leq i < j \leq n$, let Rev[i,j] denote the permutation $\begin{pmatrix} 1 & 2 & \cdots & i-1 & i & i+1 & \cdots & j-1 & j & j+1 & \cdots & n \\ 1 & 2 & \cdots & i-1 & j & j-1 & \cdots & i+1 & i & j+1 & \cdots & n \end{pmatrix}$. The Permutations Rev[i,j] are called substring reversals because they can be thought of as reversing the substring between the i-th and j-th symbol inclusive. The special substring reversal Rev[1,j] is called a prefix reversal since it reverses the prefix between the first and j-th symbols inclusive. Consider the set of generators $A_n = \{\text{Rev}[1,j] : j \geq 2\}$, and let P_n denote the corresponding Cayley graph $G(\Sigma_n,A_n)$, called the pancake network of dimension n. Our main goal is to construct efficient embeddings of hypercubes into pancake networks.

As is standard, we typically regard Σ_n as the set of all n! arrangements of a set of specified and ordered symbols (where the i-th symbol listed in an arrangement/permutation is regarded as the functional image of the i-th symbol of the given order), and we regard the generators as acting upon these symbol strings by appropriately shuffling around the n symbols. For example: in P_7 on the set of ordered symbols {A,B,C,D,E,F,G}, the vertices FDCBAGE and CDFBAGE are adjacent (via the generator Rev[1,3]). Abusing our notation slightly, we often specify a set of ordered symbols by writing its elements as a string whose left to right order conforms to the ordering of the elements (e.g. the string ABCDEFG denotes the set {A,B,C,D,E,F,G} above).

We also consider so-called "burnt" networks related to P_n. The n-dimensional burnt pancake network [CB,GP,HS], denoted by BP_n, is defined as follows. Given Σ_{2n} on the symbols $X_1,X_2,...,X_{2n}$, the vertex set of BP_n is the subgroup of $2^n n!$ arrangements of those 2n symbols so that each symbol X_{2i-1} appears consecutive to its paired symbol X_{2i}. The generator set for BP_n is {Rev[1,2i]: i ≥ 1}, thereby specifying the edges of the Cayley graph BP_n. For an arbitrary symbol string $S = Z_1Z_2...Z_m$ let S^r denote its "reverse," i.e. $S^r = Z_mZ_{m-1}...Z_1$. Using this notation we can view the vertex set of BP_n as being all arrangements of n of the 2n symbols $Y_1,Y_2,...Y_n,Y_1^r,Y_2^r,...Y_n^r$ in which exactly one symbol appears from each of the symbol pairs $\{Y_i,Y_i^r\}$, where Y_i denotes the string of two symbols $X_{2i-1}X_{2i}$ and Y_i^r denotes its reverse. In this alternative notation each vertex of BP_n is an arrangement of the n symbols Y_i, each of which is also assigned an orientation, i.e. is in either ordinary or reversed mode. When vertices of BP_n are regarded as arrangements of n "burnt"/oriented symbols instead of 2n non-oriented symbols we use the notation rev[i,j] (with a lower-case "r") to stand for the generator that corresponds to Rev[2i-1,2j] (the generator on 2n symbols). In particular rev[1,i] acts to reverse the order and change the orientation of the first i burnt symbols (as opposed to Rev[1,i] reversing the order of the first i unburnt symbols). Thus each vertex of BP_n can be thought of as a stack of n pancakes of various sizes, each burnt on one side, and the edges of BP_n are present to indicate the option of grabbing the top i pancakes and turning this substack of i upside-down, replacing them back on the rest of the stack. When such a substack is flipped, each of the i pancakes flipped changes orientation. The Cayley graph P_n has the similar interpretation, except that the pancakes have no orientation, i.e. turning a pancake over renders it indistinguishable from its previous state.

Having defined these networks, we pause to mention some desirable features they possess. Vertices of P_n have degree n-1, which is sublogarithmic relative to n!, the number of vertices. Its diameter is O(n) (see [HS] for the current best known implied constant), still sublogarithmic. Similarly, vertices of BP_n have degree n and diameter O(n), which are sublogarithmic relative to $2^n \cdot n!$, the number of vertices in BP_n. By contrast, the hypercube Q_n, being n-regular on 2^n vertices and having diameter n, has degree and diameter which are logarithmic relative to the number of vertices. Thus each of the pancake and burnt pancake networks has diameter which, relative to their number of vertices, is smaller than that of the hypercube while still retaining a smaller degree. From this limited viewpoint they are superior to the hypercube. Of course it takes much more for a network to prove itself nearly as useful as the hypercube, and the content of this paper is to show that it is possible to embed reasonably large hypercubes into these networks with very small dilation, a major step in showing that these networks can efficiently emulate hypercubes. Similar results have been shown for embedding hypercubes, binary trees, and cycles into "star networks" [BHOS,MPS1,MPS2,K], another family of Cayley networks on the symmetric group.

Since these networks are Cayley graphs, they are highly symmetric (e.g. vertex transitive), and their symmetries can be used to good effect in analyzing fault tolerance and broadcasting in these networks. Likewise, it is easy to see that P_n and BP_n are recursive: the set of vertices in P_n (resp. BP_n) in which the last "pancake" is fixed induces a subnetwork isomorphic to P_{n-1} (resp. BP_{n-1}). For other literature concerning pancake networks and related networks, see [CB,GP,HS] concerning sorting, and [AQS,BFP,GVV,La] for broadcasting, routing, and other problems.

1.2 Graph embeddings and Simulation:

As motivation for later definitions, we recall some related background on "traditional" (i.e. O(1) to 1) graph embeddings. Let G and H be two graphs representing two networks of processors, and let $f:V(G)\rightarrow V(H)$ be a function mapping vertices of the "guest" graph G into those of the "host" graph H. Such a function f yields a model for studying simulations of G by H as follows. Consider an algorithm A designed to be run on G, which we wish to implement on H. The role of processor x in G will be played in the simulation by the processor f(x) in H. When a message between adjacent vertices x and y is sent in the execution of A on G, the equivalent message will be routed in the simulation of A on H along a designated path P(x,y) between f(x) and f(y) in H. We assume that at most one message can traverses any given edge during one step of any algorithm. We call f a graph embedding of G into H, though some authors may reserve that term for f together with the routing map associating each path P(x,y) with each edge $xy\in E(G)$.

There are costs associated with such a simulation. For example, a communication time of one step in sending a message across the edge xy in G is now "stretched" into taking a communication time proportional to the length of P(x,y) in H, thereby giving rise to a slowdown factor at least proportional to the maximum of the lengths of the paths P(x,y) over all edges xy in G. There may also be several messages routed through an edge e in H, only one of which can cross e in a given step of the simulation. Assuming f specifies these paths P(x,y) for each edge xy of G, the resulting congestion of f, defined as the maximum over all edges e in E(H) of the number of paths P(x,y) containing e, contributes further to the slowdown. Let $dist_H(u,v)$ denote the distance between vertices u and v in a graph H. Now define the dilation of the map f by dilation(f) = $\max\{dist_H(f(x),f(y)): xy\in E(G)\}$. When the result of T_G consecutive steps of a computation on G can be obtained by $T_H = s\cdot T_G$ consecutive steps of a computation on H (via the simulation afforded by f), then we say that G can be simulated by H with slowdown s. Letting the load L of f be the maximum number of vertices of G mapped to any vertex of H, a result of [LMR] implies that a map f of load L, congestion c, and dilation d gives rise to a simulation of G by H with slowdown $s = O(L+c+d)$.

1.3 Our one-to-many model:

In this paper we will consider an alternative to traditional emebeddings; that is, "one-to-many" embeddings from Q_d to P_n, i.e. where the image of each vertex of Q_d is now allowed to be an arbitrary nonempty subset of $V(P_n)$, these subsets being pairwise disjoint. Such embeddings are useful when a traditional embedding must have large dilation whereas a well constructed one-to-many embedding can have low dilation (in a sense to be made precise below). We will also be concerned with maximizing the size of the guest hypercube embedded into a given host pancake network.

Let 2^X denote the power set of X. Given two networks G = (V,E) and H = (V',E') and an integer d, a function $f : V \to 2^{V'} - \{\emptyset\}$ is called a one-to-many, dilation d embedding of G into H if for every pair of vertices x,y in V the following two conditions are satisfied:

1) $f(x) \cap f(y) = \emptyset$, and
2) If x and y are adjacent in G, then for each $x' \in f(x)$ there is a designated vertex $y'_{x'} \in f(y)$ for which $dist_H(x',y'_{x'}) \le d$.

This definition is illustrated in Figure 1. To define congestion for one-to-many embeddings, suppose that for each $xy \in E$ we have also specified a path $P(x',y'_{x'})$ (resp. $P(y',x'_{y'})$) in H of length at most d joining each pair of vertices $x' \in f(x)$ and $y'_{x'} \in f(y)$ (resp. $y' \in f(y)$ and $x'_{y'} \in f(x)$). Such a collection of paths will be called an edge routing for f. Then define the congestion of f to be the maximum, over all edges $e \in E'$, of the number of paths $P(x',y'_{x'})$ containing e, as xy varies over all edges of E.

A one-to-many embedding f of G into H gives rise to a simulation of G by H. A high level view of how the simulation operates is as follows.

a) For every step of the algorithm A on G and every vertex $x \in V(G)$, in the corresponding step of the simulation of A on H every vertex $x' \in f(x)$ executes the same program and has the same input to that program as does x. This will be automatically accomplished by requiring that x' follow the same communication pattern as x in the following sense.

b) If xy is an edge in G and y sends a message to x at some step of the algorithm A, then in the simulation every vertex $x' \in f(x)$ receives the same message (except for obvious changes in addressing) from the vertex $y'_{x'} \in f(y)$ in the corresponding step of the simulation of G by H.

Thus a) is implied by b) since each such x' has been receiving the same messages at the corresponding stages during the simulation of A as x was receiving during the execution of A itself.

Condition a) stipulates that all processors in f(x) have the same information as x at corresponding stages. Thus when a processor $x' \in f(x)$ receives information one may naively anticipate a communication cost from having x' relay that information to other vertices in f(x) so that those vertices have the same information (as required by a)). However it is precisely the point of stipulation b) that if a given $x' \in f(x)$ receives information at a given simulation step then every other $x'' \in f(x)$ receives the same information at that step. Hence there is no communication cost of this type.

We discuss briefly the implementation of condition b) and the resulting correspondence between steps of A on G and steps of its simulation on H.

Again letting $T = T_G$ be the running time of algorithm A on G, take any time t, $1 \le t \le T$. Consider all edges xy in G across which a message was sent, say from y to x, during the time interval [t-1,t]. Now suppose we had a packet routing scheme R_t on H which for each pair $x' \in f(x)$ and $y'_{x'} \in f(y)$ sends the packet from $y'_{x'}$ to x' which y sent to x during this time interval. The collection of such routing schemes R_t for all t then guarantees that condition b) will be satisfied, and hence make our simulation possible, as follows. We initialize each vertex $x' \in f(x)$ to be in the same state and possessing the same information and local program as $x \in G$ at guest time t = 0. Let s_i be the time required to perform the routing R_i, $1 \le i \le T_G$. Then at host time s_1 each $x' \in f(x)$ is in the same state and has the same information as x has at time 1, having received exactly the same messages (from designated points $y'_{x'}$ for each neighbor y of x) as x received the guest time interval [0,1]. A simple induction letting t-1 and t play the roles of t = 0 and t = 1 above shows that at host time $\sum_{i=1}^{t} s_i$ each $x' \in f(x)$ is in the same state and has the same information as x had at guest time t. Thus finally at time $\sum_{i=1}^{T} s_i$ (with $T = T_G$) each $x' \in f(x)$ is in the same final state as x, so that H has simulated G.

We can immediately obtain the slowdown and settle issues of synchronization in this implementation. Let $s = \max\{s_i: 1 \le i \le T_G\}$. We can achieve synchronization by agreeing to start the i'th round of routing R_i at host time (i-1)s, even though some packets of the (i-1)'st round R_{i-1} may have arrived at their destinations well before host time (i-1)s. Concerning slowdown, observe that the message passing during the guest time interval [i-1,i] is simulated by the i'th round of routing R_i requiring $s_i \le s$ host time units. Hence $T_H = s \cdot T_G$, so we have a slowdown of s in our simulation. (We have ignored here the time required for local computations at host vertices $x' \in f(x)$ since these times are identical to the local computation time for the corresponding guest vertex $x \in f(x)$, and hence do not contribute to the slowdown.)

1.4 Other one-to-many models:

Our model is a direct adaptation to dilation $d \ge 1$ of the one-to-many embedding model introduced in [F]. In that work the one-to-many map of dilation d=1 (using other language) is defined and explored for embeddings involving hypercubes, grids, and complete graphs.

A more general one-to-many simulation model was introduced in [KLMRRS]. In that model the graph H simulates G by creating and passing pebbles among its vertices, these pebbles representing computations and message passing in G. More precisely, for each vertex $v \in V(G)$ and time t, $0 \le t \le T_G$, there is a vertex pebble (v,t) (representing the state of v at time t), and for t > 0 and any edge e of G an edge pebble (e,t) (these pebbles representing the message passed across edge e at time t). The goal of the simulation is to create a pebble (v,t) for all v and t, where at any step of the simulation a host vertex $h \in V(H)$ may do the following operations

(1) copy a single edge pebble that it contains,

(2) send a single edge pebble to a neighbor in H,

(3) create either a vertex pebble (v,t) or edge pebble (e,t) for some edge e leaving v in G if h contains the pebbles (v,t-1) and $(e_i,t-1)$, $1 \le i \le k$, where e_i are the edges of G going into v.

We examine the generality in such a simulation. Observe that multiple copies of a pebble (v,t) can be created throughout the host H by passing of pebbles. This redundancy is in effect a simulation of guest vertex v at guest time t at multiple host vertices in H; i.e. a one-to-many simulation of G by H. But this simulation is dynamic in the following sense. For any host time τ, $0 \le \tau \le T_H$, let $H(v,\tau)$ be the set of vertices of H which at host time τ play the role of v, in the sense of creating at time τ a pebble (v,t) for some t. Further let $H(v) = \bigcup_{0 \le \tau \le T_H} H(v,\tau)$, the set of all vertices in H which at any point in the simulation play the role of v. Then the operations (1)-(3) allow the possibility $H(v,\tau) \ne H(v,\tau')$ for $\tau \ne \tau'$. That is, the set of host vertices simulating a given guest vertex v can vary with host time. These operations further allow the possibility of nonempty pairwise and arbitrary k-fold intersections among the sets H(v); in effect allowing a given host vertex h to play the role of $k \ge 1$ different guest vertices for arbitrary k during the simulation, though during any single step h can play the role of only one guest vertex. In our model we disallow the analogous possibilities, i.e. the variation with time of the set $H(v,\tau)$ and the nonempty intersection of sets H(v). This is because in our model the set of vertices of H playing the role of $v \in V(G)$ is fixed to be f(v) (where f is our one-to-many map) independent of time, and $f(v) \cap f(w) = \emptyset$ for $v \ne w$ by definition of f.

This dynamic redundancy in [KLMRRS] makes possible the construction of small slowdown simulations, in some cases even in the presence of high dilation. Thus, as one example, it is shown that an N-vertex butterfly B can simulate an N-vertex mesh M with O(1) slowdown despite the fact that a one to one embedding $M \rightarrow B$ must have dilation $\Omega(\log(N))$ [BCHLR]. Several other upper and lower bounds on the slowdown of simulations are derived involving trees, multidimensional grids, and hypercube-derived networks such as the butterfly and shuffle exchange. The simulations underlying the upper bounds are static in that they satisfy $H(v) \cap H(w) = \emptyset$, for $v \ne w$. The lower bounds (with one exception) apply to any simulation based on their model, in particular to ones allowing the dynamic possibilities $H(v) \cap H(w) \ne \emptyset$ and $H(v,\tau) \ne H(v,\tau')$ described in the preceding paragraph.

Other simulations based on one-to-many embeddings can also be found in [M1], [M2], and [MW]. In [M2] a bounded degree network H on $N^{1+\varepsilon}$ vertices is constructed, with $\varepsilon > 0$ fixed but arbitrarily chosen, that can simulate with O(1) slowdown any bounded network on N vertices. In [M1] it was shown that there is no N vertex bounded degree network which can simulate all N vertex bounded networks with slowdown less than $\Omega(\frac{\log(N)}{\log(\log N)})$.

1.5 Our Results:

The focus of this paper is on embedding hypercubes into pancake (and related) networks with very small dilation D, i.e. $1 \le D \le 4$. Simultaneously we aim to keep the expansion of these embeddings low, i.e. we want to make the guest hypercube as large as our methods allow given the host pancake network.

To achieve such results, a one-to-many embedding model seems necessary. We do not know how to construct one to one maps of such small dilation and reasonably low expansion for these networks. Indeed we know that there is no dilation D = 1 one to one map $Q_2 \to P_n$ for any n, and hence no such map $Q_d \to P_n$ for $d \ge 2$. The relation between d and n in the one-to-many maps of Q_d into P_n (of the required dilation) which we construct is summarized in the table of section 6. As D increases from 1 to 4, the dimension d of the hypercube we are able to embed into P_n with dilation D increases from $d = \Theta(\sqrt{n})$ to $d = n\log(n) - (2 + o(1))n$.

Concerning the congestion of our embeddings, in the case D = 1 we trivially have congestion 1, using the path x'-$y'_{x'}$ (of length 1) in P_n joining the images of the edge xy in Q_d. When D=2 we will show a natural routing of images of edges with a congestion of 4. Hence in these two cases, having constant dilation and congestion, a slowdown of O(1) follows in the resulting simulation of Q_d by P_n. For the cases D = 3 and D = 4 we have congestion $\Omega(\log(N) - \log(\log(N)))$ and $\Omega(\log(N))$ respectively, forced by the ratio of vertex degree in the guest Q_d to the vertex degree in the host P_n.

Finally we note that it can be misleading to look at our results purely from the standpoint of the slowdown in the simulation to which they give rise. The O(1) slowdown in the cases D=1 and D=2, while certainly a desirable consequence, is in a real sense a weaker result than the dilation 1 or 2 map from which it arises. Indeed if constant factors were unimportant, then a simulation of <u>any</u> constant dilation K and constant congestion c would yield a O(1) slowdown. Again we emphasize that our goal here is to explore the one-to-many mapping technique of [F] as a tool for constructing low dilation maps of reasonably low expansion among potential computation networks.

1.6 Additional Notation and Preliminary Results:

We write $G \overset{d}{\Rightarrow} H$ to indicate that there exists a one-to-many dilation d embedding of G into H. We write $G \overset{d}{\to} H$ to indicate that there exists a one-to-one dilation d embedding of G into H, i.e. the traditional sort of embedding. We write $G \subseteq H$ to indicate that there exists a one-to-one dilation 1 embedding of G into H, i.e. that G is isomorphic to a subgraph of H.

A one-to-many dilation d embedding of G = (V,E) into H = (V',E') is essentially a particular kind of relation $\mathfrak{R}$ from V to V', where $x \mathfrak{R} y$ if and only if $y \in f(x)$. Requirement 1) above stipulates that each element of V' is related to exactly one or zero elements of V, so that the inverse relation of $\mathfrak{R}$, when restricted to the range of $\mathfrak{R}$, is in fact a function. Therefore it is typically convenient to specify a one-to-many dilation d embedding of G into H by specifying the range $R \subseteq V'$ of $\mathfrak{R}$ and specifying the function $g: R \rightarrow V$ corresponding to the inverse of $\mathfrak{R}$, thereby implicitly specifying the 'actual' embedding by $f(x) = \{y \in V' : g(y) = x\}$, where R is the part of H actually used in embedding G within it (in fact $R = \bigcup_{x \in V} f(x)$). The function g and related notation are given as follows.

A co-embedding of dilation d of G = (V,E) into H = (V',E') is a pair (R,g) where $\emptyset \neq R \subseteq V'$ and $g : R \rightarrow V$ is an onto function such that for every $y \in R$ and neighbor x' of g(y) in G there exists some $y' \in R$ such that $g(y') = x'$ and $dist_H(y,y') \leq d$. Since R is the domain of g we usually refer to g as being the co-embedding, R understood implicitly. By the comments above we see that there exists a one-to-many, dilation d embedding f of G into H if and only if there exists a dilation d co-embedding g of G into H.

Cartesian products of graphs will appear often, and it will be convenient to have some special language for them in the context of one-to-many embeddings. First recall that for graphs $G_1, G_2, ..., G_n$ the Cartesian product graph $G = G_1 \times G_2 \times ... \times G_n$ has vertices and edges given by
$V(G) = \{v: v = (v_1,v_2,...,v_n), v_i \varepsilon G_i \text{ for } 1 \leq i \leq n\}$, and
$E(G) = \{vw: v,w \; \varepsilon \; V(G)$, and for some r we have $v_i = w_i$ for $i \neq r$ while v_r is adjacent to w_r in $G_r\}$.
Many of the embeddings we construct are of the type $G \overset{d}{\Rightarrow} H$ where G is such a product. (This type of embedding naturally includes maps $Q_n \overset{d}{\Rightarrow} H$ since we can view Q_n as the n-fold product $K_2 \times K_2 \times ... \times K_2$.) Using the equivalence above we specify such an embedding by constructing a dilation d co-embedding of G into H. To do this we will typically first give a function $g:R \rightarrow V(G)$, and second show that the defining condition for a dilation d co-embedding is satisfied by this function. To facilitate this second step in the case where G is the graph product above, the following language will be useful. Let $z \varepsilon R$ and let the i'th coordinate of g(z) be the vertex y of G_i. With d understood by context, to toggle the the i'th coordinate of z is to show that for any vertex $y' \varepsilon G_i$ adjacent to y in G_i, there exists $z' \varepsilon R$ with $dist_H(z,z') \leq d$ such that g(z) and g(z') disagree only in the i'th coordinate, and that the i'th coordinate of g(z') is y'. Note the slight abuse of this notation in that the coordinate in question is literally the i'th coordinate of g(z) rather than of z. Note too that when a factor of G is a complete graph, i.e. $G_i = K_{m_i}$, then the vertex y' is allowed to be any vertex of G_i, since any two vertices of such a G_i are joined by an edge. When each factor of G is K_2, i.e. $G = Q_n$, then each of the coordinates of g(z) can be taken to be either 0 or 1 so we will then speak of toggling the i'th bit of z.

From the definition of adjacency in Cartesian products, the ability to toggle the i'th coordinate of z for all i says that for any neighbor u' of g(z) in G, there exists z'εR such that $dist_H(z,z') \leq d$ and g(z') = u'. So showing how to toggle the i'th coordinate of z for any zεR and any i, 1≤i≤n, is really proving that the proposed map g is indeed a dilation d co-embedding of the Cartesian product G into H. Thus throughout this paper we routinely and implicitly specify one-to-many dilation d embeddings of such products G into H by specifying a map $g:R \rightarrow V(G)$, $R \subseteq V(H)$, along with a demonstration of how to toggle each of the n coordinates of z for each $z \in R$. The one-to-many embedding thereby specified will be a one-to-one embedding if and only if g is one-to-one.

Some of these notions are illustrated in the following proof. Let ⊕ denote the bitwise modulo-2 sum operation for bit strings, and let ";" denote the concatenation of strings. So for example letting A = 1100 and B = 1010, we have A⊕B = 0110 and A;B = 11001010.

Lemma 1: For any m,n,d ≥ 1,
a) If $Q_m \overset{d}{\Rightarrow} P_n$ then $Q_{m+1} \overset{d}{\Rightarrow} P_{2n}$. If $Q_m \overset{d}{\Rightarrow} BP_n$ then $Q_{m+1} \overset{d}{\Rightarrow} BP_{2n}$.
b) If $Q_m \overset{d}{\rightarrow} P_n$ then $Q_{m+1} \overset{d}{\rightarrow} P_{2n}$. If $Q_m \overset{d}{\rightarrow} BP_n$ then $Q_{m+1} \overset{d}{\rightarrow} BP_{2n}$.

Proof: Let (R',g') [resp. (R",g")] be a dilation d co-embedding of Q_m into P_n, where Σ_n is viewed as permutations of the symbol string $X_1X_2...X_n$ [resp. $X_{n+1}X_{n+2}...X_{2n}$]. With Σ_{2n} being permutations of $X_1X_2...X_{2n}$, let $R = \{\alpha;\beta^r : (\alpha \in R' \text{ and } \beta \in R'') \text{ or } (\alpha \in R'' \text{ and } \beta \in R')\}$. Let $h : R' \cup R'' \rightarrow Q_m$ be defined by $h(\alpha) = g'(\alpha)$ if $\alpha \in R'$ and $h(\alpha) = g''(\alpha)$ if $\alpha \in R''$. Then define $g : R \rightarrow Q_{m+1}$ by $g(\alpha;\beta^r) = (h(\alpha) \oplus h(\beta));B$, where B = 0 if $\alpha \in R'$ and B = 1 if $\alpha \in R''$. We show that g is a dilation d co-embedding. To toggle the (m+1)-th bit of $\alpha;\beta^r$ we apply Rev[1,2n] to obtain $\beta;\alpha^r$, noting that $g(\alpha;\beta^r)$ and $g(\beta;\alpha^r)$ differ in exactly the (m+1)-th bit. To toggle the i-th bit of $\alpha;\beta^r$ for $1 \leq i \leq m$ when $\alpha \in R'$ [resp. $\alpha \in R''$] we apply the same sequence of d or fewer generators that serve to toggle the i-th bit of α relative to the co-embedding g' [resp. g"]. Having shown that any bit of $\alpha;\beta^r$ can be toggled via a path of length d or less in P_{2n}, we have proven the first part of a). The proof of the second part of a) is essentially the same. As for b), the further supposition that co-embeddings g' and g" are one-to-one is the same as supposing the conditions in the premise of b), and since the resulting g constructed is then one-to-one we have also proven b). ∎

We now introduce some additional related Cayley graphs on Σ_n. Let (i,j) denote the permutation which transposes symbols i and j. Consider the following sets of generators for Σ_n.

$B_n = \{(i,j) : 1 \leq i < j \leq n\}$
$C_n = \{(1,j) : j \geq 2\}$
$D_n = \{Rev[i,j] : 1 \leq i < j \leq n\}$, where Rev[i,j] is defined as above}.
$E_n = \{Cyc_i : i \geq 2\}$, where

$$\mathrm{Cyc}_i = \begin{pmatrix} 1 & 2 & 3 & \dots & i\text{-}1 & i & i+1 & \dots & n \\ i & 1 & 2 & \dots & i\text{-}2 & i\text{-}1 & i+1 & \dots & n \end{pmatrix}$$

$F_n = \{\mathrm{Cyc}_i : i \geq 2\} \cup \{(\mathrm{Cyc}_i)^{-1} : i \geq 2\}$

The permutations Cyc_i or $(\mathrm{Cyc}_i)^{-1}$ are called prefix cycles. Each of A_n, B_n, C_n, D_n, F_n happens to be closed under the inverse operator, so their Cayley graphs listed below are regarded as undirected graphs.

$G(\Sigma_n,B_n)$ is called the transposition network of dimension n, $G(\Sigma_n,C_n)$ is called the star network of dimension n, $G(\Sigma_n,D_n)$ is called the substring reversal network of dimension n, denoted by SSR_n, and $G(\Sigma_n,F_n)$ is called the cycle prefix network of dimension n, denoted by CP_n. The first three of these networks can be found in [AK], and CP_n can be found in [FMC]. The directed Cayley graph $G(\Sigma_n,E_n)$ is called the dicycle prefix network of dimension n, denoted by DCP_n. In a manner entirely analogous to how BP_n was defined, the n-dimensional burnt substring reversal network, denoted $BSSR_n$, has generator set $\{\mathrm{Rev}[2i-1,2j]: 1 \leq i \leq j \leq n\} =$
$\{\mathrm{rev}[i,j]: 1 \leq i \leq j \leq n\}$. Of the networks in this paper, only DCP_n is a directed graph, the others being viewed as undirected by the convention described previously.

The following examples illustrate how these Cayley graphs may be viewed as arrangements of ordered symbols joined by edges corresponding to suitable generators.

In SSR_7 on the ordered symbol set ABCDEFG there exists a path of length 2 given by EGBDFCA to EGFDBCA, to ACBDFGE (applying generators Rev[3,5]) followed by Rev[1,7]). Generally observe that in SSR_n two symbol strings X,Y are adjacent if and only if X,Y are expressible in the forms X = A;B;C and Y = A;B^r;C where the substring B must have at least 2 symbols, and substrings A,C are possibly empty.

In CP_7 on the ordered symbol set ABCDEFG there exists a path of length 2 given by EGBDFCA to GBDFECA to AGBDFEC (applying generators $(\mathrm{Cyc}_5)^{-1}$ and Cyc_7). Again observe in general that in CP_n two symbol strings X,Y are adjacent if and only if X,Y are expressible in the forms X = a;B;C and Y = B;a;C where a is a single symbol and where the substring C is possibly empty.

In P_n two symbol strings X,Y are adjacent if and only if X,Y are expressible in the forms X = A;B and Y = A^r;B where the substring B is possibly empty.

These pancake-like networks are useful models for research in computational biology. Thus for example if a segment of a DNA strand were to become detached, then flip and reattach itself in reverse orientation, then the corresponding "mutation" can be viewed as a neighboring vertex of the original strand in a substring reversal network.

The table at the end of the paper summarizes the various reversal networks defined aboved.

As for how some of the Cayley graph networks defined above relate to one another, we have the following.

Theorem 1:

a) $CP_n \xrightarrow{2} P_n$.

b) $SSR_n \xrightarrow{3} P_n$.

c) $BSSR_n \xrightarrow{3} BP_n$.

Proof: In each case the embedding f is simply the identity map. Thus to verify that each is a dilation d embedding it suffices to show that each generator in the "guest" network is a composition of d or fewer generators in the "host" network. For a), Cyc_i = Rev[1,i] ∘ Rev[1,i-1] and $(Cyc_i)^{-1}$= Rev[1,i-1] ∘ Rev[1,i]. For b), Rev[i,j] = Rev[1,j] ∘ (Rev[1,j-i+1] ∘ Rev[1,j]). For c), rev[i,j] = rev[1,j] ∘ (rev[1,j-i+1] ∘ rev[1,j]). ∎

For the rest of the paper we concentrate on constructing efficient low dilation embeddings of hypercubes into pancake networks and their relatives.

2. Dilation 1 embeddings

We begin with a theorem concerning embeddings of hypercubes and other Cartesian products of cliques into dicycle prefix networks.

Theorem 2:

a) For all m ≥ 1, $Q_m \overset{1}{\Rightarrow} DCP_{2m}$.

b) For $m_1, m_2,...,m_n \geq 2$, $K_{m_1} \times K_{m_2} \times ... \times K_{m_n} \overset{1}{\Rightarrow} DCP_s$ where $s = m_1+m_2+...+m_n$.
Thus $G_1 \times G_2 \times ... \times G_n \overset{1}{\Rightarrow} DCP_s$ where each G_i is an arbitrary graph of order m_i.

Proof: For any permutation π in $R = \Sigma_{2m}$ of the 2m symbols $X_{1,0}X_{1,1}X_{2,0}X_{2,1}...X_{m,0}X_{m,1}$, let $g(\pi) = b_1b_2...b_m$, where b_i = 0 or 1 depending upon which appears first (leftmost) in π among the symbols $X_{i,0}$ and $X_{i,1}$. To toggle the i-th bit, apply the appropriate cyclic shift to bring to the front whichever of $X_{i,0}$ and $X_{i,1}$ occurs after the other. This reverses their order and leaves every other pair $X_{k,0}$ and $X_{k,1}$ in the same relative order, so g is a dilation 1 co-embedding, proving a). The proof of b) is similar, involving the s symbols $X_{1,0}X_{1,1}...X_{1,m_1-1}...X_{n,0}X_{n,1}...X_{n,m_n-1}$, where b_i is determined by the leftmost second-subscript k appearing in a symbol $X_{i,k}$. The second part of b) follows from $G_i \subseteq K_{m_i}$. ∎

A corollary of Theorem 2b) is that the n dimensional r-ary hypercube $(K_r)^n$ has a one-to-many dilation 1 embedding into DCP_{nr} .

Lemma 2:

a) If $Q_m \subseteq BSSR_k$ then $Q_{m+2} \subseteq BSSR_{k+1}$.

b) $Q_{m-1} \subseteq BSSR_m$ for each positive integer m, via an embedding f" such that no two vertices f"(x), f"(y), x≠y, are the same m-symbol string when regarded as strings in SSR_m, i.e. with orientations suppressed.

c) $Q_{m-1} \subseteq SSR_m$.

Proof: Suppose $f : Q_m \rightarrow BSSR_k$ embeds Q_m as a subgraph of $BSSR_k$, with Σ_k being defined on the k burnt symbols $X_1X_2...X_k$. Define $f' : Q_{m+2} \rightarrow BSSR_{k+1}$ as follows, with $BSSR_{k+1}$ being on the burnt symbols $X_1X_2...X_kA$. For $x \in Q_m$ let $f'(x;00) = f(x);A$, let $f'(x;01) = f(x);A^r$, let $f'(x;10) = A^r;[f(x)]^r$, and let $f'(x;11) = A;[f(x)]^r$. It is straightforward to verify that f' embeds Q_{m+2} as a subgraph of $BSSR_{k+1}$, proving a).

For b) we proceed by induction on m, noting that Q_0 has such an embedding in $BSSR_1$, serving as a basis step. Suppose f''_m is such an embedding of Q_{m-1} in $BSSR_m$, with Σ_m on the m burnt symbols $X_1X_2...X_m$. Define $f''_{m+1} : Q_m \rightarrow BSSR_{m+1}$ as follows, with Σ_{m+1} on the burnt symbols $X_1X_2...X_mA$. For $x \in Q_{m-1}$ let $f''_{m+1}(x;0) = f''_m(x);A$ and let $f''_{m+1}(x;1) = A^r;[f''_m(x)]^r$. It is straightforward to verify that f''_{m+1} is an embedding satisfying b), completing the induction.

Suppressing the orientations of the burnt symbols in each f"(x), we see that f" induces a dilation 1 embedding of Q_{m-1} in SSR_m. ∎

Lemma 2b) fuels the proof of the following lemma, showing how one can combine various dilation 1 embeddings of hypercubes into SSR networks to produce an embedding of a larger hypercube into a larger SSR network, the dimension of the large hypercube now embedded being a bit larger than might otherwise be anticipated.

Lemma 3: If $Q_{n_i} \subseteq SSR_{k_i}$ for i=1,2,...,m then $Q_{m-1+\Sigma n_i} \subseteq SSR_{\Sigma k_i}$.

Proof: Let $f'' : Q_{m-1} \rightarrow BSSR_m$ be the embedding from Lemma 2b), relative to the m burnt symbols $Y_1Y_2...Y_m$. Let $f_i : Q_{n_i} \rightarrow SSR_{k_i}$ be dilation 1 embeddings as assumed, where each SSR_{k_i} is on the symbols $X_{i,1}X_{i,2}...X_{i,k_i}$. Then our dilation 1 embedding $f:Q_{m-1+\Sigma n_i} \rightarrow SSR_{\Sigma k_i}$ is defined as follows. For $x \in Q_{m-1}$ and $x_i \in Q_{n_i}$ let $f(x;x_1;x_2;...;x_m)$ be the string resulting from f"(x) by replacing each Y_i in f"(x) by $f_i(x_i)$, each Y_i^r in f"(x) by $[f_i(x_i)]^r$. In other words, $f(x;x_1;x_2;...;x_m)$ is the concatenation of substrings $f_1(x_1),f_2(x_2),...,f_m(x_m)$ formed by arranging them in the order determined by f"(x), and reversing those $f_i(x_i)$'s for which Y_i appears reversed in f"(x). To see that f is a dilation 1 embedding, observe that if x,y are adjacent in Q_{m-1} then $f(x;x_1;x_2;...;x_m)$ and $f(y;x_1;x_2;...;x_m)$ differ by a substring reversal which reverses the concatenation of several substrings made up of Y_i's and/or (Y_i^r)'s [i.e. one can always toggle any one of the bits corresponding to x], and if x_i,y_i are adjacent in Q_{n_i} then $f(x;x_1;x_2;...;x_i;...;x_m)$ and

$f(x;x_1;x_2;...;y_i;...;x_m)$ differ by a substring reversal internal to the substring $f_i(x_i)$ or $[f_i(x_i)]^r$ of $f(x;x_1;x_2;...;x_i;...;x_m)$ [i.e. one can always toggle any one of the bits corresponding to x_i]. ∎

Theorem 3:

a) If $Q_n \subseteq SSR_k$ then $Q_{n+1} \subseteq SSR_{k+1}$.

b) If $Q_n \subseteq SSR_k$ then $Q_{mn+m-1} \subseteq SSR_{mk}$.

c) $Q_{2^n+2^{n-2}-1} \subseteq SSR_{2^n}$.

Proof: Claim a) follows from Lemma 3 via the parameters m=2, n_1=0, n_2=n, k_1=1, k_2=k. Claim b) follows from Lemma 3 via the parameters m, $n_1=n_2=...=n_m=n$, $k_1=k_2=...=k_m=k$. Claim c) is proven by induction on n. The basis case n=2 is handled by the embedding illustrated in Table 1. Assuming that $Q_{2^n+2^{n-2}-1} \subseteq SSR_{2^n}$, then from b) with m=2 we have
$Q_{2(2^n+2^{n-2}-1)+2-1} \subseteq SSR_{2^{n+1}}$, i.e. $Q_{2^{n+1}+2^{n-1}-1} \subseteq SSR_{2^{n+1}}$, completing the induction. ∎

Since Q_m has 4-cycles and P_n does not, it is never possible to embed Q_m into P_n for m>1 via a one-to-one dilation 1 embedding. Hence we consider building such embeddings by the one-to-many technique. Toward this end, the following "exponent" and "block" terminology will be convenient for describing certain sets of vertices in P_k. Suppose P_k is defined on the symbol set $A_1A_2...A_{k-r}B_1B_2...B_r$. We then write A^t as shorthand for a string composed of t many distinct A's. Every string $x \in V(P_k)$ is of the form

$$x = W_0(x);D_1(x);W_1(x);D_2(x);W_2(x);\cdots;D_r(x);W_r(x),$$

or for brevity $x = W_0;D_1;W_1;D_2;W_2;\cdots;D_r;W_r$ when x is understood by context, where each W_i is a substring of distinct A's such that $W_0;W_1;\cdots;W_r$ is a permutation of $A_1A_2...A_{k-r}$ and $D_1D_2\cdots D_r$ is a permutation of $B_1B_2...B_r$. Thus the W_i's are the maximal substrings of consecutive A's in x, and for any integer k≥1 the string W_k is the one immediately following the k-th B symbol from the left. The substring W_0 is called the front string of x, and for each i≥1 substring W_i is called a block of x, and for i≥0 we let $N_x(i)$ denote the number of A symbols in W_i. Given a sequence of nonnegative integers $x_0,x_1,...,x_r$ summing to k-r, we let $\underline{A^{x_0}BA^{x_1}B\cdots BA^{x_r}}$ denote the set of vertices in P_k for which $N_x(i) = x_i$ for all i. An unsubscripted B or A symbol with no exponent refers to a single symbol of the given type. As an example, the string $A_1A_4B_3A_3B_2A_8A_5B_1A_6A_7A_2$ belongs to the collection of vertices $A^2BABA^2BA^3$ in the pancake network P_{11} having the symbol set $\{A_i: 1≤i≤8\}\cup\{B_i: 1≤i≤3\}$.

Our main theorem in this section is Theorem 4, in which we give dilation 1, one-to-many embeddings of hypercubes into pancake networks. Theorem 4 will be a special case of the more general Lemma 4, the latter capturing the method underlying the construction of these embeddings. So as to motivate the ideas behind Lemma 4, we next prove a claim whose value is in that its proof will help with understanding Lemma 4. This claim happens to concern dilation 2 (not dilation 1) one-to-many embeddings, yet it prepares the ideas leading to Lemma 4.

Claim: Let $G_1,G_2,...,G_m$ be graphs, and let $\nu_i = |V(G_i)| \geq 2$. Suppose for each i that I_i is an interval of ν_i many positive integers, the largest element M_i in each I_i satisfying $M_i \geq \nu_i$-1 . Let $M = \sum_{i=1}^{m} M_i$. Then $G_1\times G_2\times G_3\times...\times G_m \overset{2}{\Rightarrow} P_{m+M}$.

Proof: Let P_{m+M} be defined on the symbol set $A_1A_2...A_MB_1B_2...B_m$. We assume that the vertex set of each G_i is $\{0,1,2,...,\nu_i-1\}$. This choice of vertex set for G_i is convenient since vertices of $G = G_1\times G_2\times G_3\times...\times G_m$ are then merely m-tuples of whole numbers, allowing the possibility that we specify a particular coordinate of a desired vertex by a simple count. We construct a co-embedding g for a dilation 2 one-to-many embedding of G into P_{m+M} . Let the domain R of g be the set of vertices in P_{m+M} of the form $A^{x_0}BA^{x_1}BA^{x_2}B...BA^{x_m}$ for which $x_i \in I_i$ for i = 1,2,..,m. That is, a string $x\in V(P_{m+M})$ is in R if and only if the length of block W_i is an element of interval I_i for i = 1,2,...,m. Note that R is nonempty, since its subset $A^0BA^{M_1}BA^{M_2}B...BA^{M_m}$ is nonempty. The rule for the function g is given by

$$g(x) =(M_1–N_x(1), M_2–N_x(2), ..., M_m–N_x(m)).$$

That is, the i-th coordinate of g(x) is the amount by which $N_x(i)$ falls short of its maximum allowable value M_i . Thus the ν_i consecutive integer values in I_i correspond (in reverse order) to the ν_i possible values of the i-th coordinate of a vertex in G.

To show that g is a dilation 2 co-embedding as claimed, we show for each $y\in R$ how to toggle any coordinate of y. So let x' be any neighbor of x=g(y) in G, and we show that there exists some $y'\in R$ such that g(y') = x', where y' can be obtained from y using at most two prefix reversals. Suppose that the sole coordinate in which x and x' differ is the i-th coordinate. Let $c \leq \nu_i$-1 be the i-th coordinate of x', where we recall that $M_i–N_y(i) \neq c$ is the i-th coordinate of x. The following sequence of two prefix reversals illustrates how such a y' can be found starting with arbitrary y, and the reader may find this useful in following the remainder of the proof.

$y = W_0 B W_1 B W_2 B...B W_{i-1} B W_iB...W_m$ (abbreviating $W_i(y)$ as W_i throughout)

$\rightarrow z = BW_{i-1}^r BW_{i-2}^r B...BW_1^r BW_0^r W_iB...W_m$ (note that $W_0^r W_i = A^{N_z(i)-(M_i-c)};A^{M_i-c}$)

$\rightarrow y' = A^{N_z(i)-(M_i-c)}BW_1BW_2B...BW_{i-1}BA^{M_i-c}B...W_m$

As our first step, we apply to y the prefix reversal which reverses all symbols to the left of W_i, letting z be the string thus obtained. Blocks $W_1(y)$ through $W_{i-1}(y)$ of y appear in z internally reversed and collectively in reverse order (as the blocks $W_{i-1}(z)$ through $W_1(z)$ of z respectively), and the front string $W_0(y)$ of y appears in z now merged with $W_i(y)$ to form a new block $W_i(z)$ consisting of $N_y(0)+N_y(i)$ many A symbols. Now, $N_y(0) = M - \sum_{k=1}^{m} N_y(k) = \sum_{k=1}^{m} (M_k–N_y(k)) \geq$

$M_i – N_y(i)$, so $W_i(z)$ contains at least M_i many A symbols. Since $M_i \geq \nu_i -1 \geq c$, we have that $M_i – c$ is a nonnegative integer. As our second step, we apply to z the prefix reversal which reverses all symbols to the left of the rightmost $M_i – c$ symbols of $W_i(z)$ (i.e. affecting all the leftmost $N_z(i)$ - $(M_i$-c) many symbols of $W_i(z)$), letting y' be the string thus obtained. It is easy to see that g(y') = x' as desired, since in transforming z to obtain y' we have restored blocks W_1 through W_{i-1} of y as appearing as the first i-1 blocks from left to right in y', we have left undisturbed blocks W_{i+1} through W_m of y, and have rendered the i-th block from the left in y' as having precisely $M_i – c$ many A symbols so that the i-th coordinate of g(y') is indeed c. ∎

We summarize this proof as an interchange of A symbols between blocks and the front string. The front string W_0 acted as a "reservoir" composed of $N_y(0) = \sum_{k=1}^{m} (M_k – N_y(k))$ many A symbols, a supply sufficiently large that (should the need arise) we could use the A's in W_0 to increase the values $N_y(1)$, $N_y(2)$,..., $N_y(m)$ to their "maximum" allowed values M_1, M_2, ..., M_m respectively. Should we wish to reduce a value $N_y(i)$ we need only transfer some of the A symbols of W_i into the front string. In the span of two prefix reversals we saw that we could leave undisturbed all blocks other than W_i and in the process change the value of $N_x(i)$ to any desired value in I_i , thereby proving that we could alter any one coordinate of g(y) to any desired value by using as input some vertex y' at distance at most 2 from y.

In Lemma 4 we will succeed in implementing the spirit of the above proof so as to construct dilation 1, one-to-many embeddings of arbitrary Cartesian products of graphs into pancake networks of reasonably small dimension. Each string $x \in R$ will still be partitioned (via m many B symbols) into a front string W_0 followed by m blocks W_i, $1 \leq i \leq m$, where the block W_i consists of $N_x(i)$ many A symbols. The co-embedding g to be defined at each $x \in R$ will still depend only on the list of numbers $N_x(1), N_x(2), ..., N_x(m)$, and g(x) will in fact closely resemble the list of amounts $M_1 – N_x(1)$, $M_2 – N_x(2)$, ..., $M_m – N_x(m)$ by which the block lengths fall short of their maximum allowable values, much as before. But there are two major obstacles to overcome in modifying the above proof so that it operates under the heavy constraint that the dilation involved be only 1, not 2.

(i) We still wish to alter the number of A symbols in only one block, but now all in the span of a single prefix reversal. Yet how can we in a single prefix reversal alter the coordinate in g(y) corresponding to some block W_i of y and leave all other coordinates unchanged, despite the fact that blocks to the left of W_i will appear in reversed order in the desired result y'?

(ii) When we had 2 prefix reversals available we could afford to spend one of them merging W_0 with W_i and spend another one sending some A symbols from the resulting block back to the front string. Now we must accomplish this give and take in only 1 prefix reversal. Thus our difficulty is to manage in one step to use the $N_y(i)$ many A symbols from W_i to "replenish" the "reservoir" W_0, and also to leave precisely the desired number $N_{y'}(i)$ many A symbols in the i-th block.

To overcome obstacle (i), we construct the set R and the co-embedding g so that rearranging the order and orientation in which blocks W_1 through W_{m-1} appear has no effect upon the output of g. To accomplish this, we begin with a new set of intervals I_i, $1 \le i \le m$, with so far only the requirement that their maximum elements M_i satisfy $M_1 \le M_2 \le \ldots \le M_{m-1}$. Next for $x \in R$ and $i=1,2,\ldots,m-1$ we let $O_x(i)$ denote the i-th smallest of the lengths $N_x(i)$, $1 \le i \le m-1$, of the first m-1 blocks of x, breaking ties arbitrarily. That is, $O_x(1) \le O_x(2) \le \ldots \le O_x(m-1)$, where the sequence $O_x(1), O_x(2), \ldots, O_x(m-1)$ is simply the result of sorting the sequence $N_x(1), N_x(2), \ldots, N_x(m-1)$ so that its terms are ordered from smallest to largest. Finally we redefine the domain R of G so that a string $x \in V(P_{m+M})$ is in R if and only if $N_x(m) \in I_m$ and $O_x(i) \in I_i$ for each $1 \le i \le m-1$ (instead of requiring each $N_x(i)$ to be in I_i). We thus have a way of associating to each block $W_j(x)$ an interval I_i. That is, the block $W_m(x)$ corresponds to I_m, and for $1 \le j \le m-1$ the block $W_j(x)$ corresponds to I_i where $N_x(j)$ is the i-th smallest among the blocks $W_1(x), \ldots, W_{m-1}(x)$. Moreover this association remains nearly invariant under the action of applying a single prefix reversal to x, thereby overcoming the bulk of obstacle (i). In particular, the sequence $O_x(1), O_x(2), \ldots, O_x(m-1)$ remains unperturbed, except for the possible relocation and change in value of a single term, when we apply a prefix reversal to x. Consequently we base our co-embedding g upon this sequence of numbers, as follows. We redefine g(x) for each $x \in R$ by the rule $g(x) = (M_1 - O_x(1), M_2 - O_x(2), \ldots, M_{m-1} - O_x(m-1), M_m - N_x(m))$. Note that once again the coordinates of g(x) are simply the amounts by which the block lengths fall short of their maximum allowable values. Now if we apply a prefix reversal to $y \in R$ by "reaching into" the i-th smallest among the blocks W_j with j<m and thereby obtain a string $y' \in R$, it will be the case (despite the fact that the first j-1 blocks of y get scrambled) that the sequences $O_y(1), O_y(2), \ldots, O_y(m-1)$ and $O_{y'}(1), O_{y'}(2), \ldots, O_{y'}(m-1)$ will differ only in the i-th term (i.e. g(y) and g(y') agree in all but their i-th coordinates) so long as we arrange that the new length of the altered block W_j is still between $O_y(i-1)$ and $O_y(i+1)$ inclusive. Now, presuming that y' is in R (as required in showing that a proposed g is a one-to-many co-embedding), we can ensure that this new length $N_{y'}(j)$ meets this condition by making the following stipulations about the intervals I_i. We stipulate that intervals $I_1, I_2, \ldots, I_{m-1}$ are arranged "head to tail" in the following sense. We require that $M_1 \le M_2 \le \ldots \le M_{m-1}$ and that $I_i = [M_{i-1}, M_i]$ for $i=2,3,\ldots,m-1$, so that consecutive intervals I_i, I_{i+1} have intersection $\{M_i\}$ and are therefore "nearly disjoint". So suppose that y and y' are as above. These requirements concerning the M_i's and I_i's are such that <u>if</u> $N_{y'}(j) \in I_i$ (a condition we will later ensure) then it will follow that $O_y(i-1) \le N_{y'}(j) \le O_y(i+1)$ because $O_y(i-1)$ and $O_y(i+1)$ are themselves in I_{i-1} and I_{i+1} respectively. Thus g(y) and g(y') will differ only in the i-th coordinate, overcoming obstacle (i).

As for obstacle (ii), let's examine (for $i \ne m$) the action on $y \in R$ of a prefix reversal which "reaches into" $W_j(y)$ (where $N_y(j) = O_y(i)$), so as to alter $O_y(i)$, thus creating the string $y' \in R$. By this action, the block $W_j(y)$ must absorb the entire front string $W_0(y)$, but in exchange we can send to the front string anywhere between none and all of the $O_y(i) = N_y(j)$ many A's in $W_j(y)$. Let $T = \sum_{i=1}^{m}(\nu_i\text{-}1)$. We simplify our analysis by doing bookkeeping separately for various parts of the front string of y. We regard the $N_y(0)$ many A symbols of $W_0(y)$ as composed of

a) $M_i–O_y(i)$ many A symbols "reserved" for the occasion where we are to increase $O_y(i)$ to its maximum value M_i , and

b) the remaining $N_y(0) –(M_i–O_y(i)) = \sum_{k\neq i}(M_k–O_y(k))$ many A symbols "reserved" for similarly altering other values $O_y(k)$ for $k\neq i$.

Now, the vertex y could well be such that each $O_y(k)$ for $k\neq i$ is at its minimum allowable value in I_k, namely $M_k–(\nu_k\text{-}1)$, in which case $W_0(y)$ contains the complementarily large number $\sum_{k\neq i}(M_k–O_y(k)) = \sum_{k\neq i}(\nu_k\text{-}1) = T–(\nu_i\text{-}1)$ many A symbols "reserved" for altering values $O_y(k)$ for $k\neq i$, in addition to the "reserved" $M_i–O_y(i)$ many A symbols mentioned above. Therefore our construction will not succeed unless the interval I_i is chosen so that its least element is $T–(\nu_i\text{-}1)$ or larger so that we can "replenish" the portion of "reservoir" $W_0(y)$ reserved for intervals I_k, $k\neq i$. This act of "replenishing" amounts to an even exchange of $N_y(0) –(M_i–O_y(i))$ many A symbols of $W_0(y)$ with $N_y(0) –(M_i–O_y(i))$ many A symbols of $W_j(y)$. Being an exchange of equally many A symbols from one place to another, this component of the transaction has no effect on the number $O_y(i)$. However, in addition to this even exchange we note that such a prefix reversal applied to y sends the $M_i–O_y(i)$ many A symbols of $W_0(y)$ [formerly "reserved" for altering $O_y(i)$] into $W_j(y)$ in forming $W_j(y')$, in exchange for sending yet more A symbols from $W_j(y)$ into the new front string $W_0(y')$. This aspect of the bookkeeping is completely independent of most of the symbols in y. We need only be concerned with the $M_i–O_y(i)$ many A symbols of $W_0(y)$ and the $O_y(i) – (T–(\nu_i\text{-}1))$ or more A symbols of $W_j(y)$ that we are free to use, having reserved $T–(\nu_i\text{-}1) \geq N_y(0) –(M_i–O_y(i))$ others for the "even exchange" in which we swap only $N_y(0) –(M_i–O_y(i))$ many A's form this reserved section of $W_j(y)$ with the same number of A's from $W_0(y)$ [leaving some t many A's in this reserved section for some $t \geq 0$]. See Figure 3 for an illustration. Thus we are led to the problem of how we can exchange $M_i–O_y(i)$ many A symbols in a front string with some of the available $O_y(i) – (T–(\nu_i\text{-}1))$ A symbols in a "back string" (for a total of $M_i–T+\nu_i–1$ many A symbols) so that via a single prefix reversal we can perform transfers reflecting the structure of graph G_i. This leads to the following useful definition.

As a building block from which we can construct more complicated embeddings in the manner motivated above, we introduce a simple kind of co-embedding whose output is merely the length of the front string in the input string. For a graph G with no isolated vertices, we call a dilation 1, one-to-many embedding $f:G \to P_n$ a B-embedding and its associated co-embedding g a B-map if they satisfy the following conditions:

1) $V(G) = \{0,1,...,|V(G)|-1\}$.
2) P_n is defined on the symbol set $A_1A_2...A_{n-1}B$.
3) The co-embedding g has domain $R = \{x \in V(P_n): N_x(0) < |V(G)|\}$ and follows the simple rule $g(x) = N_x(0)$, i.e. maps each string $x \in R$ to the vertex of G given by the length of the front string of x, i.e. $g(A^{x_0}BA^{x_1}) = x_0$ for all $0 \le x_0 < |V(G)|$.

Figures 2a and 2b show B-embeddings $f:Q_3 \to P_{10}$ and $f:Q_2 \to P_5$ respectively. Given a graph G, since 2) and 3) entirely specify g once an assignment of vertex labels $0,1,2,...,|V(G)|-1$ to V(G) has been selected, it is a simple matter to decide whether G has a B-map into P_n for given n. In fact, it is easy to verify the following: for a graph G on vertex set $\{0,1,...,|V(G)|-1\}$, there exists a B-map of G into P_n if and only if $i+j < n$ for every edge $\{i,j\}$ of G. Several observations follow quickly from this characterization:

a) If G has minimum degree δ and a B-embedding into P_n then $n \ge |V(G)|+\delta-1$ (just use $i=|V(G)|-1$ above).
b) From a), $n \ge |V(G)|$ for any B-map.
c) There exists a B-map for embedding the clique K_m into P_{2m-2}, and by a) this is optimal.
d) The hypercube embeddings in Figure 2 are optimal as B-embeddings.

Before we continue, we give an important yet simple restatement of 3), as follows. For a B-map g for embedding G in P_n, where $V(G) = \{0,1,...,|V(G)|-1\}$ and where P_n is defined on the symbol set $A_1A_2...A_{n-1}B$, let I denote the interval of integers $I = [n-|V(G)|, n-1]$. Condition 3) is equivalent to the alternate condition that g has domain $R = \{x \in V(P_n): N_x(1) \in I\}$ and follows the simple rule $g(x) = n-1 - N_x(1)$, i.e. maps each string $x \in R$ to the vertex of G given by the amount by which $N_x(1)$ falls short of its maximum allowable value n-1, i.e. $g(A^{n-1-x_1}BA^{x_1}) = n-1-x_1$ for all $x_1 \in I$. This alternate rule will be used in Lemma 4 along the following general lines. We will be given B-maps g_i for each of graphs G_i, $1 \le i \le m$. In that lemma a string $x \varepsilon R$ will represent a vertex in $G_1 \times G_2 \times G_3 ... \times G_m$ by having each of its blocks represent a vertex of some particular factor G_i. This vertex is determined by the amount by which the length of that block falls short of a specified maximum M_i. The possible such amounts will be in one to one correspondence with the possible values n_i-1-x_1 for the B-map g_i associated with G_i, and in this way will be able to toggle the i'th coordinate in x by closely mimicking the action of g_i.

Lemma 4: Let $G_1,G_2,...,G_m$ be graphs with no isolated vertices, let $\nu_i = |V(G_i)|$, and let

$T = \sum_{i=1}^{m}(\nu_i\text{-}1)$. For each i suppose there exists a B-embedding $f_i:G_i \rightarrow P_{n_i}$. Suppose for $1 \leq i \leq m$ that $I_i = [M_{i-1}, M_i]$ is an interval of ν_i many positive integers, with the largest element M_i in each I_i satisfying $M_i \geq T+n_i-\nu_i$. Let $M = \sum_{i=1}^{m} M_i$. Then $G_1 \times G_2 \times G_3 \times ... \times G_m \overset{1}{\Rightarrow} P_{m+M}$.

Proof: Let P_{m+M} be defined on the symbol set $A_1A_2...A_MB_1B_2...B_m$. Let $G = G_1 \times G_2 \times ... \times G_m$. To go along with the B-embeddings, we assume for each i that $V(G_i) = \{0,1,2,...,\nu_i\text{-}1\}$, and let $g_i:R_i \rightarrow G_i$ be the B-map corresponding to each f_i. We let R be the set of all $x \in V(P_{m+M})$ for which $N_x(m) \in I_m$ and $O_x(i) \in I_i$ for each i=1,2,...,m-1. Note that R is nonempty, since its subset $A^0BA^{M_1}BA^{M_2}B...BA^{M_m}$ is nonempty. Define g:R→G by the rule

$$g(x) = (M_1-O_x(1), M_2-O_x(2), ..., M_{m-1}-O_x(m\text{-}1), M_m-N_x(m)).$$

That is, the i-th coordinate of g(x) is the amount by which the length of the block of x associated with the graph factor G_i falls short of its maximum value M_i, so that the rule for g(x) is essentially derived from the various rules for the g_i (see the discussion immediately preceding the Lemma).

We will show how to toggle any coordinate of a given $y \in R$. So let x' be any neighbor of g(y) = x in G, and we will show the existence of the required $y' \in R$, i.e. satisfying g(y') = x' with y' obtainable from y via a single prefix reversal. Let x and x' have their sole disagreement in the i-th coordinate, with $i \neq m$. Also let $d \leq \nu_i\text{-}1$ be the i-th coordinate of x', noting that the i-th coordinate $M_i-O_y(i)$ of x is not c. Let block $W_j(y)$ have length $O_y(i)$, j<m, so that this block is i-th smallest among the first m-1 blocks of y. From observation b) concerning the B-map g_i, we have $n_i \geq \nu_i$. Since $M_i \geq T+n_i-\nu_i$, we have $O_y(i) - (T-(\nu_i-1))) \geq n_i-1-(M_i-O_y(i)) \geq 0$. Therefore $W_j(y)$ is expressible in the form $A^{T-(\nu_i-1)}\, A^{n_i-1-(M_i-O_y(i))}A^r$ with $r \geq 0$. [Here the section of $T-(\nu_i-1)$ many A's are the ones available for the "even exchange" as in the discussion preceding the lemma. This leaves $n_i\text{-}1\text{-}(M_i-O_y(i))+r$, expressed as $O_y(i) - (T-(\nu_i-1))$ in that same discussion, many A's which we are free to use for adjusting the i-th coordinate of g(y).] Meanwhile, $W_0(y)$ is expressible in the form $A^{N_y(0)-(M_i-O_y(i))}\, A^{M_i-O_y(i)}$. Since g_i is a B-map we are given that it is possible to exchange the $A^{M_i-O_y(i)}$ portion of $W_0(y)$ with exactly d of the A symbols in the $A^{n_i-1-(M_i-O_y(i))}$ portion of $W_j(y)$ because $d + (M_i-O_y(i)) \leq n_i-1$ (from the property of B-maps that i+j < n for every edge ij). Observe now that

$$N_y(0) -(M_i-O_y(i)) = \sum_{k=1}^{m-1}(M_k-O_y(k)) + M_m\text{-}\ N_y(m) - (M_i-O_y(i)) \leq \sum_{k \neq i}(\nu_k\text{-}1) = T-(\nu_i-1).$$

Thus we know that the $A^{T-(\nu_i-1)}$ portion of $W_j(y)$ is sufficiently large that we can perform an even exchange (as in the discussion preceding this lemma) of the entire $A^{N_y(0)-(M_i-O_y(i))}$ portion of $W_0(y)$ with $N_y(0) -(M_i-O_y(i))$ many A symbols from the $A^{T-(\nu_i-1)}$ portion of $W_j(y)$. Thus block $W_j(y)$ contains sufficiently many A symbols that we can perform a prefix reversal which "reaches into" that block and does the following. It evenly exchanges $N_y(0)-(M_i-O_y(i))$ many A's from $W_0(y)$ with the same number of A's from $W_j(y)$, while exchanging the remaining $M_i-N_y(j)$ (i.e. $M_i-O_y(i)$) many A's reserved in $W_0(y)$ for adjusting the i-th coordinate of y with d many A's taken from the $A^{n_i-1-(M_i-O_y(i))}$ portion of $W_j(y)$. See Figure 4 for an illustration. Thus the resulting string y' has its number of A symbols in $W_j(y')$ given by

$$N_{y'}(j) = N_y(j) + (M_i-N_y(j)) - d = M_i - d.$$

Note that $M_i - d \in I_i$ since $0 \le d \le \nu_i-1$, fulfilling our promise that $N_{y'}(j)\in I_i$ (see the discussion prior to the lemma). That is, the block $W_j(y)$ (the i-th smallest among the first m-1 blocks of y) has been transformed into the i-th smallest among the first m-1 blocks of y'. Therefore g(y) and g(y') agree in all but the i-th coordinate. This coordinate becomes d in the resulting string y' obtained from y by the prefix reversal. The same argument applies when i=m, replacing each j by m and each O by N. Therefore g is a dilation 1 co-embedding. ∎

Now we reach our main result of this section, yielding reasonably efficient dilation 1 one-to-many embeddings of hypercubes and products of cliques into pancake networks. Its proof simply requires that we select appropriate choices for the numbers M_i and B-embeddings f_i as in Lemma 4, verify that the hypotheses of the lemma are satisfied, and compute the resulting parameters of the embedding produced by the lemma. After the proof, we will say more about optimality in part a).

<u>Theorem 4</u>:

a) $Q_n \overset{1}{\Rightarrow} P_k$ with congestion 1, where $n \ge 8$ and $k = \frac{9}{2}m^2 + 2n^2 - 4mn + \frac{21}{2}m - 6n + 7$, where m is the integer nearest $\frac{8n-21}{18}$ (arbitrarily breaking any ties for nearest integer). Thus we have $Q_n \overset{1}{\Rightarrow} P_k$, with $k = \frac{10}{9}n^2 + O(n)$ as n grows.

b) For the m-fold Cartesian product $(K_n)^m$ of n-cliques, $(K_n)^m \overset{1}{\Rightarrow} P_{(n-1)(3m^2-m+2)/2}$ with congestion 1. Thus $G_1 \times G_2 \times ... \times G_m \overset{1}{\Rightarrow} P_{(n-1)(3m^2-m+2)/2}$ where each G_i is an arbitrary graph of order n.

<u>Proof</u>: The claims on congestion follow immediately if we demonstrate the indicated dilation 1 embeddings, for then we use the edge routing in which $P(x',y'_{x'})$ is just the edge x'-$y'_{x'}$.

We apply Lemma 4, factoring Q_n as a product of lower dimensional hypercubes. The various dimensions of the factors in this product, and the order in which these factors are listed (i.e. which factor plays the role of G_i for each i in the statement of that Lemma) are choices we must make. These choices help determine the dimension m+M of the pancake network into which Q_n is embedded by the Lemma, and we will make these choices with a view to minimizing m+M. The extent to which we succeed in making the optimal choices will be discussed later.

In our choice of factorization, we begin by requiring each factor to be isomorphic to either Q_2 or Q_3. Fixing the number of such factors to be m, it follows that the numbers of factors isomorphic to Q_2 and to Q_3 are 3m-n and n-2m respectively.

Next we give a particular ordering of these factors and show how this ordering (together with the choice for the dimensions of factors above) yields, via Lemma 4, a dimension m+M equal to the expression for k given in part a). We take the first factor to be Q_3, each of the next 3m-n factors to be Q_2, and each of the final n-2m-1 factors to be Q_3. That is, we let $G_1 = Q_3$, $G_i = Q_2$ for each i in the interval [2,3m-n+1], and $G_i = Q_3$ for each i in the interval [3m-n+2, m]. For each factor G_i isomorphic to Q_2 let f_i be the B-embedding from Figure 2b so that $n_i=5$ and $\nu_i=4$ for each such i, in the notation of Lemma 4. Likewise, for each factor G_i isomorphic to Q_3 let f_i be the B-embedding from Figure 2a so that $n_i=10$ and $\nu_i=8$ for each such i. Therefore $T = \Sigma(\nu_i-1) = 3(3m-n) + 7(n-2m) = 4n-5m$. For $1 \le i \le m$ we can now specify the intervals I_i, where each I_i may be viewed as corresponding to the i-th factor G_i, by specifying the numbers M_i, $1 \le i \le m$, which serve as the endpoints of these intervals. Corresponding to G_1 we let $M_1 = T+n_1-\nu_1 = 4n-5m+2$, thereby satisfying the requirement that $M_1 \ge T+n_1-\nu_1$. Corresponding to the next 3m-n factors isomorphic to Q_2, we take the M_i in such a way that each interval $I_i = [M_{i-1},M_i]$ contains the required $\nu_i = 4$ integers; that is we let $M_i = M_1+3(i-1) = 4n-5m-1+3i$ for $2 \le i \le 3m-n+1$. Corresponding to the next n-2m-2 factors isomorphic to Q_3 we choose the M_i so that the resulting intervals $[M_{i-1},M_i]$, $3m-n+2 \le i \le m-1$, contain the required $\nu_i = 8$ integers; that is, we let $M_i = M_{3n-m+1}+7(i-(3m-n+1)) = 8n-17m-5+7i$ for these i. Notice that all our choices for the M_i so far, that is for $1 \le i \le m-1$, satisfy $M_i \ge M_1 = T+n_1-\nu_1 \ge T+n_i-\nu_i$, showing that the requirement $M_i \ge T+n_i-\nu_i$ is also satisfied. As for M_m, the Lemma requires only $M_m \ge T+n_m-\nu_m = 4n-5m+2$, so we take $M_m = 4n-5m+2$. Therefore the dimension of the pancake network into which Q_n is embedded by Lemma 4 is

$$\begin{aligned} m+M &= m + \sum_{i=1}^{3m-n+1} M_i + \sum_{i=3m-n+2}^{m-1} M_i + M_m \\ &= m + \sum_{i=1}^{3m-n+1} (4n-5m-1+3i) + \sum_{i=3m-n+2}^{m-1} (8n-17m-5+7i) + M_m \\ &= \frac{9}{2} m^2 + 2n^2 - 4mn + \frac{21}{2} m - 6n + 7, \end{aligned}$$

giving us the expression for the dimension k claimed in part a).

Now minimizing this expression over all integers m in the relevant interval $[\frac{n}{3}, \frac{n}{2}]$, we find by using the first derivative that the minimum occurs at the integer m nearest $\frac{8n - 21}{18}$, as long as $n \ge \frac{21}{2}$ so that this minimum lies in $[\frac{n}{3}, \frac{n}{2}]$. Substituting this into the expression for k we obtain $k = \frac{10}{9} n^2 + O(n)$, completing the proof of a).

For part b), in Lemma 4 use $G_i = K_n$ for each of the m factors (so each ν_i=n and

T=mn-m), and let each f_i be the B-embedding from observation c) (in the discussion immediately following the definition of B-embeddings), so that each n_i = 2n-2. Let M_m = mn-m+n-2, and for each i<m let M_i = mn-m-1+i(n-1), so that m+M = (n-1)($3m^2$-m+2)/2, giving the pancake network dimension as in b). This time the intervals I_i determined are composed of ν_i=n many positive integers, with $M_i \geq T+n_i-\nu_i$, proving b). ∎

In devising Theorem 4a we used the flexibility available in factoring Q_n as a product of hypercubes of smaller dimensions. The goal is naturally to find a factorization which, given n, optimizes (i.e. minimizes) the pancake dimension m+M (denoted k in the theorem) resulting from the application of Lemma 4. The factorization we used in the proof had the following features.

i) All factors are isomorphic to Q_2 or Q_3.

ii) There are at least two factors isomorphic to Q_3.

iii) The ordering of the factors is first Q_3, followed by all the Q_2 factors, and finally followed by the remaining Q_3 factors.

iv) There are $m = \frac{8n - 21}{18}$ (rounded to the nearest integer) factors, from which it follows using i) that the fraction of them which are isomorphic to Q_2 and Q_3 is $\frac{3m\text{-}n}{m} = \frac{3}{4} + o(1)$ and $\frac{1}{4} + o(1)$ respectively.

We briefly outline here a proof that this factorization is the optimum over all factorizations of Q_n satisfying i). The proof of Theorem 4a) above already shows that among factorizations satisfying i), ii), and iii), the optimum also satisfies iv). This, together with the following two facts, yield the optimality of our factorization over those satisfying i).

(I) Among the factorizations satisfying i), the optimum one also satisfies ii).

(II) Among the factorizations satisfying i) and ii), the optimum one also satisfies iii).

The proof of (I) comes, when n is even, from comparing the value of m+M resulting from a factorization into all Q_2's with the value resulting when three of the Q_2's are exchanged for two Q_3's, with one of the latter placed first in the ordering of factors and the other last. The smaller value of m+M occurs with the factorization involving two Q_3's. For n odd we use three Q_3's in place of one Q_3 and three Q_2's. The proof of (II), given a fixed number m of factors, comes in two stages. First we show that among all factorizations satisfying i) and ii) an optimum is attained under the ordering in which the factors $G_2, G_3, ..., G_{m-1}$ are arranged with all the Q_2's coming first and all the Q_3's coming next. This is easily seen by analyzing the effect on m+M of interchanging the order of successive factors, one of which is Q_2 and the other is Q_3. The proof is then reduced to comparing the values of m+M resulting from the four possible orderings in which the G_i, $2 \leq i \leq m-1$, are so arranged but where G_1 and G_m can be Q_2 or Q_3 independently. A messy calculation, omitted here for brevity, shows that the optimum ordering satisfies iii) for $n \geq 8$.

While we offer no proof, we believe that application of Lemma 4 using factors of dimensions other than 2 or 3 would lead to no improvement. In any case, no claim is made concerning the optimality of the embeddings behind Theorem 4a. These embeddings are simply the best we know how to produce for large n via the restrictive but convenient sort of embedding employed in Lemma 4.

Note that Theorem 4b shows that $(K_4)^n \overset{1}{\Rightarrow} P_{(9n^2-3n+6)/2}$. Thus $(K_4)^n$, considerably more dense than its subgraph Q_{2n}, is embedded into a pancake network of nearly the same dimension as the one into which Q_{2n} is embedded by Theorem 4a.

3. Dilation 2 embeddings

We next give dilation 2 embeddings of hypercubes into pancake networks. The embedding given in Theorem 5a) is one-to-one, but is otherwise inferior to the one-to-many embedding given next in Theorem 5b) in that the ratio of host network dimension to guest network dimension in the first result is considerably larger than that ratio in the second.

Theorem 5:

a) For $n \geq 4$, $Q_n \overset{2}{\rightarrow} P_{2n-2}$.

b) $Q_n \overset{2}{\Rightarrow} P_{2n}$, with an edge routing of congestion 4.

Proof: Figure 5 shows a dilation 2 embedding of Q_4 into P_4, proving a) when n=4. Straightforward iteration of Lemma 1b) completes the proof of a).

For b), let f' be the dilation 1, one-to-many embedding of Q_n into DCP_{2n} as given in Theorem 2a). Network DCP_{2n} embeds as a subdigraph into CP_{2n} via the identity embedding function id, and in turn CP_{2n} has a dilation 2 embedding f" into P_{2n}, as shown in Theorem 1a). Therefore the composition $f'' \circ id \circ f'$ is a one-to-many dilation 2 embedding.

Concerning congestion, for any $xy \in E(Q_n)$ and $x' \in f(x)$ let $y'_{x'} \in f(y)$ be the designated point in f(y) at distance 2 from x' in P_{2n}. The toggling corresponding to f' tells us that $y'_{x'}$ is obtained by performing a right cyclic shift by one symbol of some prefix of x'. Thus wlog we can let x' = AqB and $y'_{x'}$ = qAB, with q a single symbol and A a length i-1 string for some i. We define the edge routing for the map $Q_n \overset{2}{\Rightarrow} P_{2n}$ by letting $P(x',y'_{x'})$ be AqB — A^rqB — qAB, a length 2 path in P_{2n}. We refer to the edge AqB — A^rqB of $P(x',y'_{x'})$ generated by the shorter prefix reversal as the "first" edge of $P(x',y'_{x'})$, and to the other edge as the "second" edge of $P(x',y'_{x'})$.

Now take any edge $e \in E(P_{2n})$, where wlog we can let e = AqB — A^rqB with A still a length i-1 string for some i. There are two possible routing paths that could use e as a first edge, the one in the above paragraph and also A^rqB — AqB — qA^rB. Now write $A = \alpha_1 a_1 = a_2 \alpha_2$, where a_i is a single symbol for i = 1,2 and α_1 (resp. α_2) is the length i-2 prefix (resp. suffix) of A. Then there are also two possible routing paths that could use e as a second edge, namely $\alpha_2 a_2$qB

— A^rqB — $AqB = a_2\alpha_2qB$ and $\alpha_1{}^r a_1qB$ — AqB — $A^rqB = a_1\alpha_1{}^rqB$. Hence e is contained in at most 4 routing paths defined for the map $Q_n \overset{2}{\Rightarrow} P_{2n}$, as required. ∎

As should be no surprise, we can do considerably better embedding into burnt pancake networks.

Theorem 6: $Q_n \overset{2}{\Rightarrow} BP_n$.

Proof: As usual let $V(BP_n)$ be the set of all arrangements of n of the 2n burnt symbols $Y_1,Y_2,...Y_n,Y_1{}^r,Y_2{}^r,...Y_n{}^r$ in which exactly one symbol appears from each of the symbol pairs $\{Y_i,Y_i{}^r\}$. Let $R = V(BP_n)$, and for a string $S \in R$ define the co-embedding g by $g(S) = b_1b_2...b_n$ where $b_i = 1$ if Y_i appears in reversed orientation, $b_i = 0$ if Y_i appears in ordinary orientation. To toggle the i-th bit of S when burnt symbol Y_i or its reverse appears as the j-th symbol of S where j>1, apply the generator rev[1,j-1] and then rev[1,j]. This reverses the orientation of Y_i without altering the orientations of any other burnt symbols. In the case j=1 simply apply rev[1,1]. ∎

In one instance we know of a dilation 2 embedding of an (n+1)-dimensional hypercube into an n-dimensional burnt pancake network, an improvement over Theorem 6. This is in the case n=2, where BP_2 happens to be isomorphic to an 8-cycle, and there exists a dilation 2 embedding of Q_3 into BP_2.

As we approach our next theorem it is useful to observe that the efficiency of our embeddings improves in the following sense. Prior to Theorem 6, our embeddings associated only one or a small constant number of hypercube bits per block. In Theorem 6 we succeeded in associating one hypercube bit per symbol. For the duration of the paper, we continue this trend by associating to each A symbol a bit string. In Theorem 7b) below, each A symbol accounts for a nonconstant number k many bits of the hypercube being embedded. Hence the efficiency of our embeddings is improving as we succeed in associating progressively longer bit strings with each A symbol, in that the ratio of the dimension of the hypercube embedded to the dimension of the pancake host is increasing. This is to be expected as we allow the dilation in our embeddings to increase.

Theorem 7:

a) For $n \geq m$, $(K_m)^{n-m+1} \overset{2}{\to} CP_n$, where $(K_m)^{n-m+1} = K_m \times K_m \times ... \times K_m$ (n-m+1 factors). Thus $G_1 \times G_2 \times ... \times G_{n-m+1} \overset{2}{\to} CP_n$ where each G_i is an arbitrary graph of order m.

b) For $n \geq 2^k$, $Q_{k(n-2^k+1)} \overset{2}{\to} CP_n$.

Proof: Let Σ_k be on the symbols $A_1,A_2,...,A_{n-m+1},D_1,D_2,...,D_{m-1}$. Let R be the set of those permutations of the form $\alpha_0;D_1;\alpha_1;D_2;\alpha_2;D_3;...;\alpha_{m-2};D_{m-1};\alpha_{m-1}$ in which within each substring α_i it is the case that the A's appear in lexicographic order, i.e. A_c appearing to the left of A_d within substring α_i implies that c < d. For an arbitrary string $S = \alpha_0;D_1;\alpha_1;D_2;...;\alpha_{m-2};D_{m-1};\alpha_{m-1}$ in R and integer $1 \le i \le n-m+1$, let S(i) be the index such that A_i appears within the substring $\alpha_{S(i)}$ of S. Then we define the co-embedding g by g(S) = S(1);S(2);...;S(n-m+1). To toggle S(i) we can apply a single prefix cycle to bring symbol A_i to the front of the string and then another prefix cycle to move symbol A_i into any α_h of our choosing, making sure that it is filed within α_h lexicographically. Thus S(i) is rendered equal to any integer from 0 to m-1 without affecting any other coordinates S(j), proving a). Part b) and the second part of a) follow from a) since $G_i \subseteq K_m$ and $Q_k \subseteq K_{2k}$ respectively. ∎

4. Dilation 3 embeddings

Continuing with the theme of using divider symbols, we present our dilation 3 result. We will use the following special notation here and in the next section. Given strings $s_m, s_{m+1},\ldots,s_n$ over some alphabet, we let <u>Con{i=m,n;s_i}</u> denote the string $s = s_m;s_{m+1};\ldots;s_n$ obtained by concatenating the s_i in the order of increasing i proceeding along s from left to right.

<u>**Theorem 8**</u>**:**

a) $(K_n)^{k-2n} \overset{3}{\Rightarrow} P_k$, where $(K_n)^{k-2n} = K_n \times K_n \times \ldots \times K_n$ (k-2n factors), with $k \ge 2n$.
Thus $G_1 \times G_2 \times \ldots \times G_{k-2n} \overset{3}{\Rightarrow} P_k$ where each G_i is an arbitrary graph of order n.

b) Let m and k be positive integers satisfying $k \ge 2$ and $1 \le m \le \lfloor \log_2(k) \rfloor -1$.
Then $Q_{m(k-2m+1)} \overset{3}{\Rightarrow} P_k$.

<u>Proof</u>: Let $V(K_n) = \{1,2,\ldots,n\}$. We view $V(P_k)$ as the set of permutations on the symbols $B_1,B_2,\ldots,B_n$, $D_1,D_2,\ldots,D_n$, $A_1,A_2,\ldots,A_{k-2n}$.

We will define a set $R \subseteq P_k$ and a co-embedding $g:R \rightarrow (K_n)^{k-2n}$. For convenience, in the following description we will omit the indices of the D-symbols, and just write each of them as D. We then let R be the set of all strings in P_k of the form

$$D;\alpha_1;B_{i_1};\alpha'_1;D;\alpha_2;B_{i_2};\alpha'_2;D;\ldots D;\alpha_n;B_{i_n};\alpha'_n, \qquad (*)$$

where the induced string $\alpha_1;\alpha'_1;\alpha_2;\alpha'_2\cdots;\alpha_n;\alpha'_n$ is a permutation of $A_1, A_2,\ldots, A_{k-2n}$. In particular, i_t denotes the index of the t-th B symbol from left to right.

For a given $S \in R$ expressed as in (*), define g(S) as follows. For each r, $1 \le r \le k-2n$, we let $\beta(r,S)$ be the index i_t such that the symbol A_r is in the substring $\alpha_t;B_{i_t};\alpha'_t$. Thus $\beta(r,S)$ indicates the subscript of the B symbol belonging to the block containing A_r. We then let g(S) = Con{r=1,k-2n;β(r,S)}.

To see how to toggle the i-th coordinate $\beta(i,S)$ of string b = g(S) for a given $S \in R$, let b' be the string obtained from b by altering the i-th coordinate of b to a desired element e of $V(K_n)$. Let e = i_t and let x denote the substring $\alpha_t;B_{i_t};\alpha'_t$ of S. We can then write $S = D\alpha DxD\gamma A_i\delta$ for suitable substrings α, γ, and δ of S, where a similar argument can be given if instead x follows A_i in the string S. Now let $S' = D\alpha^r DA_i xD\gamma\delta$, so that $\beta(i,S')=b'$ and $\beta(j,S')=\beta(j,S)$ for any $j\neq i$, and hence S' is a string obtained from S by toggling the given bit. Finally we observe that dist(S,S') ≤ 3 via the pancake steps $S\to\gamma^r Dx^r D\alpha^r DA_i\delta \to A_i D\alpha DxD\gamma\delta \to S'$, thereby completing the proof of the first part of a), the second part following since the graph product in question is a subgraph of $(K_n)^{k-2n}$. Part b) then follows from the second part of a) by letting each G_i equal Q_m and $n = 2^m$. ∎

We now find a value of m which at least comes close to maximizing the hypercube dimension $u(k,m) = m(k-2^{m+1})$ of Theorem 8b for a fixed k and $1 \leq m \leq \lfloor \log_2(k) \rfloor -1$. Letting u(m) denote u(k,m) with k fixed, we find that its derivative is

$$u'(m) = k - 2^{m+1}(1 + m\ln(2)).$$

Observe that the equation $u'(m) = 0$ must have a unique solution since the second term on the right side of the equation for $u'(m)$ is an increasing function of m. Also for large enough k we have $u'(1) > 0$ while $u'(\lfloor \log_2(k) \rfloor -1) < 0$. Hence u(m) has a unique maximum m* for positive m, and this maximum does indeed occur somewhere in the interval $[1, \lfloor \log_2(k) \rfloor -1]$ specified by Theorem 8b.

Unfortunately a simple closed form solution to the equation $u'(m) = 0$ seems unavailable, so we cannot specify the maximum point m* in a simple way. However the following closer analysis pins down m* within a reasonably narrow subinterval of $[1, \lfloor \log_2(k) \rfloor -1]$. For convenience denote $\log_2(k)$ by log(k), and let C be any constant.

Now observe that if we evaluate $u'(m)$ at $m = \log(k) - C\log(\log(k))$ we get

$$u'(\log(k) - C\log(\log(k))) = k - 2\frac{k}{[\log(k)]^C}(1 + \ln(2)(\log(k) - C\log(\log(k)))).$$

It follows for any constant $C > 1$ that $u'(\log(k) - C\log(\log(k))) > 0$ for large k, while for $C < 1$ $u'(\log(k) - C\log(\log(k))) < 0$ for large k. Hence for any two constants α and β satisfying $\alpha<1<\beta$ we have $\log(k) - \beta\log(\log(k)) < m^* < \log(k) - \alpha\log(\log(k))$ for k sufficiently large. Thus a reasonable approximation to m* is $\log(k) - \log(\log(k))$, leading to the choice of m used in the following corollary to Theorem 8b.

Corollary 8.1: $Q_{w(k)} \overset{3}{\Rightarrow} P_k$, where w(k) is given by

$w(k) = k[\log_2(k) - \log_2(\log_2(k))] - (2+ o(1))k.$

Proof: By the theorem, we can obtain a map $Q_{u(k,m^{**})} \overset{3}{\Rightarrow} P_k$, where m** is asymptotically $\log_2(k) - \log_2(\log_2(k))$, and by the discussion preceding the corollary this m** is the near optimum choice for m in the theorem. Plugging m** in for m in u(k,m) we get

$$u(k,m^{**}) \sim u(k, \log_2(k) - \log_2(\log_2(k))) = [\log_2(k) - \log_2(\log_2(k))]\cdot[k - 2\frac{k}{\log_2(k)}]$$

$$= k[\log_2(k) - \log_2(\log_2(k))] - (2+ o(1))k.$$ ∎

5. Dilation 4 embeddings

In our next theorem giving dilation 4 embeddings, we combine the method of divider symbols with recursion. It is interesting to observe that the dimension of the hypercube we can so embed is asymptotically the same as the best known result for the dimension of the hypercube that can be embedded with dilation 4 into the star network [NSK] (see the concluding remarks). Part i) of the theorem handles special pancake network dimensions, while part ii) interpolates between these values.

Theorem 9:

(i) $Q_{(k-2)2^k+2} \overset{4}{\Rightarrow} P_{2^k-1}$

(ii) If $0 \le z < 2^k$ then $Q_{(k-2)2^k+2+z(k-1)} \overset{4}{\Rightarrow} P_{2^k-1+z}$.

Proof: Let $T(k) = (k-2)2^k+2$. We start by showing that $Q_{T(k)} \overset{4}{\Rightarrow} P_{2^k-1}$.

First we let the symbols of P_{2^k-1} be B(1), B(2), ..., B(2^{k-1} -1), A(1), A(2), ..., A(2^{k-1}). Now express each string $S \in P_n$ in the form $S = \alpha(0);B(i_1);\alpha(1);B(i_2);\alpha(2)... ;B(i_m);\alpha(m)$, with $m = 2^{k-1} - 1$, where each $\alpha(i)$ is a string over the A's such that the letter sets of the $\alpha(i)$ partition the set {A(1), A(2), ..., A(2^{k-1})}; that is, the concatenation $\alpha(1);\alpha(2);...;\alpha(m)$ is a permutation of the string A(1)A(2)...A(2^{k-1}).

We proceed as usual by defining a co-embedding $g:P_{2^k-1} \rightarrow Q_{T(k)}$. Let $S^{(0)}=S \in P_{2^k-1}$ be expressed as above, and define binary strings $b(r,S^{(0)})$ of length k-1 for each $1 \le r \le 2^{k-1}$ by letting $b(r,S^{(0)})$ be the binary equivalent of the integer i_s such that A(r) is a symbol in the substring $\alpha(s)$ when $i_s>0$, and letting it be the all 0's string of length k-1 when A(r) is a symbol in the substring $\alpha(0)$. Thus $b(r,S^{(0)})$ indicates the subscript of the nearest B symbol to the left of A(r).

Proceeding inductively suppose that for each i, $0 \le i \le k-2$, we have constructed a string $S^{(i)} \in P_{2^{k-i}-1}$ together with binary strings $b(r,S^{(i)})$ of length k-i-1 for each $1 \le r \le 2^{k-i-1}$. In analogy with the above, we suppose further that $P_{2^{k-i}-1}$ is defined over the symbols $B^{(i)}(1)$, $B^{(i)}(2)$, ..., $B^{(i)}(2^{k-i-1}-1)$, $A^{(i)}(1)$, $A^{(i)}(2)$, ..., $A^{(i)}(2^{k-i-1})$, and we express each $S^{(i)}$ in the form

$$S^{(i)} = \alpha^{(i)}(0);B^{(i)}(i_1);\alpha^{(i)}(1);B^{(i)}(i_2);\alpha^{(i)}(2);...;B^{(i)}(i_m);\alpha^{(i)}(m),$$

where $m = 2^{k-i-1}-1$, and where the $\alpha^{(i)}$'s are strings over the $A^{(i)}$'s. We call the substring $B^{(i)}(i_t);\alpha^{(i)}(t)$ of $S^{(i)}$ a <u>block</u> of $S^{(i)}$ with <u>index</u> i_t, and also a block (of S) at <u>level i</u> (when there is no need to reference the value of the index i_t). Now let $S^{(i+1)} \in P_{2^{k-i-1}-1}$ be the string obtained from $S^{(i)}$ by replacing block $B^{(i)}(c_d);\alpha^{(i)}(d)$ by a symbol $B^{(i+1)}(r)$ when $1 \le r \le 2^{k-i-2}-1$ and $c_d=2r$, and replacing such a block by the symbol $A^{(i+1)}(r)$ when $1 \le r \le 2^{k-i-2}$ and $c_d=2r-1$. Thus $S^{(i+1)}$ can be expressed in a form similar to the one given for $S^{(i)}$ above, except with i replaced by i+1 and $m = 2^{k-i-2}-1$. We then define the length k-i-2 binary string $b(r,S^{(i+1)})$ to be the binary equivalent of the index c_d such that $A^{(i+1)}(r)$ is in the block $B^{(i+1)}(c_d);\alpha^{(i+1)}(d)$ of $S^{(i+1)}$.

Figure 6 illustrates the inductive construction of the $S^{(i)}$.

Finally we define co-embeddings $g_k : P_{2^k-1} \to Q_{T(k)}$ by letting

$$g_k(S) = Con\{i=0,k-2;Con\{r=1,2^{k-i-1};b(r,S^{(i)})\}\},$$

that is, $g_k(S)$ is the concatenation of the various bit strings $b(r,S^{(i)})$ over all possible r and i. The cube dimension T(k) given in the codomain of g_k is seen to be correct by the following induction. In base case k = 2 we have P_3 defined on the letters {A(1),A(2),B(1)}. The construction associates a bit b(1,S) and b(2,S) to A(1) and A(2) respectively, this being 0 (resp. 1) if the symbol is to the left (resp. right) of B(1) in S. Hence $g_k(S)$ has length 2, in agreement with T(2) = 2. Proceeding inductively observe that there is a length k-1 binary string associated to each of the 2^{k-1} many A-symbols in the strings of P_{2^k-1}. Hence level 0 contributes $(k-1)2^{k-1}$ bits to $g_k(S)$ for any $S \in P_{2^k-1}$. Assuming inductively that the codomain dimension of g_{k-1} is $T(k-1) = (k-3)2^{k-1}+2$, the codomain dimension of g_k is therefore $T(k-1)+(k-1)2^{k-1} = (k-2)2^k+2$. This completes the proof of i).

For the proof of ii), we now consider pancakes of dimension $n = 2^k -1+z$, z>0, i.e. dimension not of the form $2^k -1$. The basic idea is to pad the number of A-symbols, keeping the same number of B-symbols as when $n = 2^k -1$. We now regard P_n as being defined on the symbols {B(1), B(2), ..., $B(2^{k-1} -1)$, A(1), A(2), ..., $A(n-2^{k-1}+1)$}, and starting with any $S \in P_n$ we form the analogous set of binary strings $b(r,S^{(0)})$, $1 \le r \le n-2^{k-1}+1$, and $b(r,S^{(i)})$, $1 \le r \le 2^{k-i-1}$, when i >0. These strings describe the positions of symbols $A^{(i)}$ in blocks at level i, just as above. We omit the remaining nearly identical details. Though this construction applies to any integer $n > 2^k -1$, the dimension of the hypercube embedded in P_n is maximized when we choose k as large as possible, i.e. $k = \lfloor \log_2(n+1) \rfloor$. This explains the restriction on z in the statement of ii).

We now describe how to toggle any bit σ of S (as usual, σ is a bit in the string $g_k(S)$). Let σ be contained in the substring $b = b(r,S^{(i)})$ of g(S), and let b' be the binary string obtained from b by flipping σ. Recall that b denotes the position within $S^{(i)}$ of the symbol $A^{(i)}(r)$ [as indicated by the index of the block at level i to which $A^{(i)}(r)$ belongs]. But $A^{(i)}(r)$ is in turn obtained by a sequence of coalescing operations applied to a nested sequence of substrings of S [expressed as blocks at various levels]. Thus $A^{(i)}(r)$ corresponds naturally to a substring x of S. Hence toggling this bit amounts to moving x to a different [and in general arbitrary] location in the string S, the latter position being determined by the string b'. Letting S' be the string obtained from S in this way, we can write S as yzxt and S' as yxzt for suitable substrings y,z,and t. We must then show that

(a) S' satisfies dist(S,S') ≤ 4, and

(b) The strings g(S) and g(S') agree in all bits excepting σ.

Statement (a) follows from the path of length 4 from S to S' in P_n given by

$yzxt \to z^R y^R xt \to x^R yzt \to y^R xzt \to yxzt$.

Consider now statement (b). Clearly any two strings $b(t,S^{(i)})$ and $b(t,S'^{(i)})$ for t ≠ r are identical since no A-symbol at level i other than $A^{(i)}(r)$ was moved in S to obtain S'. So we must now compare any two strings $B = b(t,S^{(j)})$ and $B' = b(t,S'^{(j)})$ with $j \neq i$ and t arbitrary, and show that they are identical.

Suppose first that $j<i$. Now B is the index of the block in $S^{(j)}$ containing $A^{(j)}(t)$, and this block remains a single symbol or is coalesced into other symbols at higher levels. It follows that $A^{(j)}(t)$ is contained in this block under any movements of symbols at the higher level i required to obtain S'. Hence B and B' are identical.

Now suppose that $j>i$. Thinking of B as the block index of the substring x of S corresponding to $A^{(j)}(t)$, we see that movements of A-symbols at lower levels [resulting in S'] possibly change the composition of x but do not change the index. Hence B and B' are identical. Thus (b) is proved.

We now determine the cube dimension associated with the embedding for arbitrary $n = 2^k-1+z$. Observe that the co-embedding g associates a length k-1 bit string with each of the A-symbols (at level 0) used to increase the number of A-symbols (that number now being $n-2^{k-1}+1$). The rest of the construction is identical, so now the cube dimension is $(k-2)2^k+2+z(k-1)$, as required. ∎

6. Concluding Remarks

The focus of this paper has been the construction of very low dilation maps of hypercubes into pancake (or related substring reversal) networks as a step in the simulation of the former by the latter. A subsidiary goal has been to show how the technique of one-to-many embeddings can be useful in such constructions.

For a fixed pancake network, there is a tradeoff between the size of the hypercube embedded and the dilation for such maps. We therefore list below the largest dimension d (as a function of n) for which we have succeeded in embedding Q_d into P_n, indicating the relevant result. The notation $\Theta(f(n))$ stands as usual for a function whose absolute value is bounded above and below by a constant multiple of f(n) for n large.

dilation D	Largest dimension d for which our constructions give $Q_d \overset{D}{\Rightarrow} P_n$	Reference
1	$\frac{3}{\sqrt{10}}\sqrt{n} + O(1)$, (congestion 1)	Theorem 4
2	$\frac{1}{2}n$, for n even (congestion 4)	Theorem 5
3	$n\big(\log(n) - \log(\log(n))\big) - (2 + o(1))n$	Corollary 8.1

4	nlog(n) - (2 + o(1))n	Theorem 9

The usefulness of the one-to-many embedding technique is clear from our ability to embed a hypercube of higher dimension in P_n than we are able to via a one to one embedding at these low dilations. For D = 1 we obtained embeddings $Q_d \overset{D}{\rightarrow} SSR_n$, where d = Θ(n), from Theorem 3c. While this shows that for dilation 1 we were able to embed a larger hypercube into SSR_n than into P_n, this is no surprise because of the higher degree $\Theta(n^2)$ in SSR_n as compared to Θ(n) in P_n.

The threshold dilation D = 4 seems like a natural place to stop when investigating very low dilation maps of hypercubes into pancake networks. First, the dimension d = nlog(n) - (2 + o(1))n of the embedded hypercube matches that obtained (implicitly) in [NSK] for one-to-one and dilation at most 4 embeddings $Q_d \rightarrow S_n$ of a hypercube into the star network S_n of dimension n. The latter is the Cayley graph $G(\Sigma_n,T_n)$ on Σ_n with generator set T_n consisting of the n-1 transpositions through the first position, that is, $T_n = \{(1,i): 2\le i\le n\}$. Extending the result of [NSK], maps $Q_d \rightarrow S_n$ and $Q_d \rightarrow P_n$ were constructed [MPS1] with the larger hypercube dimension $d = n\log(n) - (\frac{3}{2} + o(1))n$ and dilation bounded above by a constant K, but where K is considerably larger than 4. Thus dilation 4 seems a natural "largest" very low dilation to investigate in this context. Of course the interesting problem remains of investigating how large d can be in a map $Q_d \rightarrow P_n$ or $Q_d \overset{D}{\Rightarrow} P_n$ with reasonably small dilation (say bounded by some constant or slowly growing function of n), and what tradeoff exists between dilation D and hypercube dimension d in such a map.

We conclude with the reference table below of the substring reversal networks studied in this paper.

Network	Vertices	Edges AB
P_n	all n! strings of length n obtained by permuting the symbols {1,2,...,n}	B is obtained from A by reversing any initial substring of A; that is, A = S;T, B = S^r;T for suitable S and T.
BP_n	all $2^n n!$ strings of length n over the alphabet $\{1,2,..,n,1^r,2^r,..n^r\}$, where for each i exactly one of $\{i,i^r\}$ appears in the string	$A = x_1x_2...x_n$, $B = x_i^r x_{i-1}^r..x_1^r x_{i+1}x_{i+2}...x_n$ for some i, where if $x_t = m^r$ for some m, then $x_t^r = m$.
SSR_n	the same as for P_n	B is obtained from A by reversing any substring of A; that is, A = R;S;T, B = R;S^r;T for substrings R, S, T with $\|S\| \geq 2$, with R or T possibly empty.
$BSSR_n$	the same as for BP_n	B is obtained from A by reversing any substring of A, and in the process reversing each symbol in the reversed substring; that is, $A = x_1x_2...x_n$, and $B = x_1x_2..x_{i-1}x_j^r x_{j-1}^r..x_i^r x_{j+1}x_{j+2}...x_n$ for some i and j. Again if $x_t = m^r$ then $x_t^r = m$.
CP_n	the same as for P_n	B is obtained from A by a cyclic rotation, in either direction, of an initial substring of A; that is, A = q;S;R and B = S;q;R, or A = S;q;R and B = q;S;R, where q is a single symbol.
DCP_n (directed)	the same as for P_n	B is obtained from A by a right cyclic shift of an initial substring of A; that is, A = S;q;R and B = q;S;R, where q is a single symbol. Thus AB is a directed edge from A to B.

References

[AK] S. Akers and B. Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," IEEE Trans. Comput., vol. 38 (4)(1989), 555-566.

[AKM] S.G. Akl, Q. Ke, H. Meyer "On some properties and algorithms for the star and pancake interconnection networks", J. Parallel and Distributed Computing, Vol. 22, No. 1 (1994) 16-25.

[AQS] S.G. Akl, K. Qiu, I. Stojmenovic, "Fundamental algorithms for star and pancake interconnection networks with application to computational geometry", Networks Vol. 23, No. 4 (1993) 215-225.

[BCHLR] S.N. Bhatt, F.R.K. Chung, J.W. Hong, F.T. Leighton, A.L. Rosenberg, "Optimal simulations by butterfly-like networks", Journal of the ACM 43, No. 2 (1996) 293-330.

[BCLR] S.N. Bhatt, F.R.K. Chung, F.T. Leighton and A.L. Rosenberg, "Efficient embeddings of trees in hypercubes," SIAM J. Computing (1992) 151-162.

[BFP] P. Berthome, A. Ferreira, S. Perennes, "Optimal information dissemination in star and pancake networks", *Proc of 1993 Fifth IEEE Symposium on Parallel and Distributed Processing* (1995) (IEEE Comput. Soc. Press, Los Alamitos, Ca.).

[BHOS] A. Bouabdallah, M.C. Heydemann, J. Opatrny, and D. Sotteau, "Embedding complete binary trees into star and pancake graphs," manuscript, Comp. Sci. Sept., Concordia University, Montreal, Canada (1993).

[BMPS] S. Bettayeb, Z. Miller, T. Peng, and I.H. Sudborough, "Embedding d-D meshes into optimum hypercubes with dilation 2d-1," Parallel and Distributed Computing - Theory and Practice, *First Canada-France Conf. Proc.* (Cosnard, Ferreira, Peters eds.) in Lecture Notes in Computer Science 805 Springer Verlag (1994), 73-80.

[C] M.Y. Chan, "Embedding of grids into optimal hypercubes," SIAM J. Computing, (1991), 834-864.

[CB] D.S. Cohen and M. Blum, "Improved bounds for sorting pancakes under a conjecture," to appear in Algorithms, available from Comp. Sci. Division, U.C. Berkeley, May 1992.

[F] M. Fellows, "Encoding Graphs in Graphs", Ph.D. Dissertation, Dept. of Computer Science, University of California, San Diego (1985).

[FMC] V. Faber, J.W. Moore, and W.Y.C. Chen, "Cycle prefix digraphs for symmetric interconnection networks," Networks 23 (1993) 641-649.

[GP] W. H. Gates and C. H. Papadimitriou, "Bounds for sorting by prefix reversal," Discrete Math, Vol. 27 (1979), 47-57.

[GVV] L. Gargano, U. Vaccaro, A. Vozella, "Fault tolerant routing in the star and pancake interconnection networks", Information Processing Letters, Vol. 45 No. 6 (1993) 315-320.

[HS] M.H. Heydari and I.H. Sudborough, "On sorting by prefix reversals and the diameter of pancake networks," in *Proc. First Heinz Nixdorf Symp., November 1992*, Lec. Notes in Comp. Sci., vol. 678, Springer-Verlag (1993), 218-227.

[K] A. Kanevsky, "On the embedding of cycles in pancake graphs", Parallel Computing 21, No. 6 (1995) 923-936.

[KLMRRS] R. Koch, F.T. Leighton, B.M. Maggs, S.B. Rao, A.L. Rosenberg, E.J. Schwabe, "Work preserving emulations of fixed-connection networks", Journal of the ACM 44 No.1 (1997) 104-147.

[L] F.T. Leighton, Introduction to Parallel Algorithms and Architectures: Arrays, Trees, Hypercubes, Morgan Kaufmann Publishers (1992).

[LMR] F.T. Leighton, B. Maggs, and S. Rao, "Universal packet routing algorithms", in Proceedings of the 29'th Annual Symposium on Foundations of Computer Science (Oct. 1988) 256-271, IEEE, New York.

[M1] F. Meyer Auf Der Heide, "Efficiency of universal parallel computers", Acta Inf. 19 (1983) 269-296.

[M2] F. Meyer Auf Der Heide, "Efficient simulations among several models of parallel computers", SIAM J. Computing 19 (1986) 106-119.

[MPS] Z. Miller, D. Pritikin, and I.H. Sudborough, "Bounded dilation maps of hypercubes into Cayley graphs on the symmetric group", Math. Systems Theory 29 (1996) 551-572.

[MPS2] Z. Miller, D. Pritikin, and I.H. Sudborough, "Near embeddings of hypercubes into Cayley graphs on the symmetric group", IEEE Trans. Comput., 43 (1) (1994), 13-22.

[MS1] B. Monien and I.H. Sudborough, "Embedding one interconnection network into another," Computing, *Computing Suppl. 7*, Springer-Verlag (1990) 257-282.

[MS2] B. Monien and I.H. Sudborough, "Simulating binary trees on hypercubes", *Proc. of 1988 AWOC Conference on VLSI Algorithms and Architectures* (July 1988), in Lecture Notes in Computer Science, Vol. 319, Springer Verlag (1994) 170-180.

[MW] F. Meyer Auf Der Heide, R. Wanka "Time-optimal simulations of networks by universal parallel computers", in Proceedings of the 6'th Symposium on Theoretical Aspects of Computer Science (1989) 120-131.

[NSK] M. Nigam, S. Sahni, and B. Krishnamurthy, "Embedding hamiltonians and hypercubes in star interconnection graphs", in Proc. Int'l Conf. Parallel Processing Vol. III (1990) 340-343.

0000	1234	1000	4321
0001	2134	1001	4312
0010	1243	1010	3421
0011	2143	1011	3412
0100	1324	1100	4231
0101	2314	1101	4132
0110	1423	1110	3241
0111	2413	1111	3142

Table 1: A dilation 1 embedding of Q_4 into SSR_4, showing $Q_4 \subseteq SSR_4$.

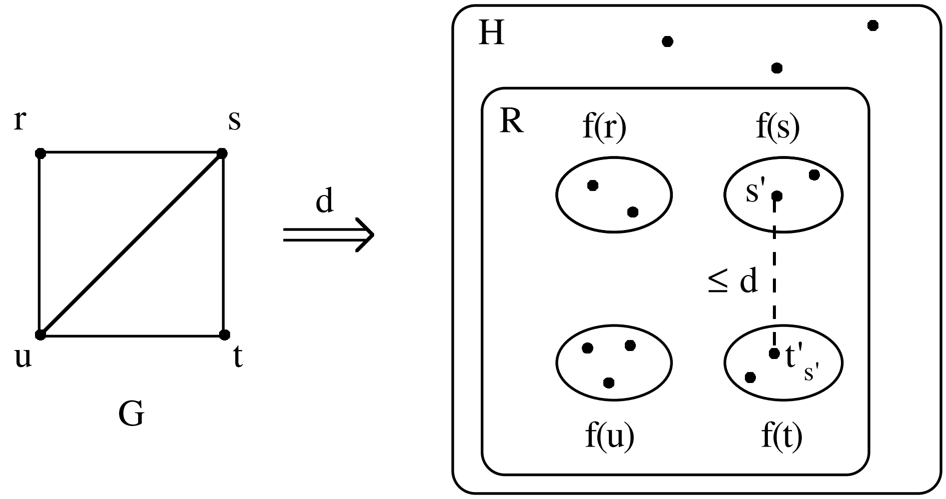
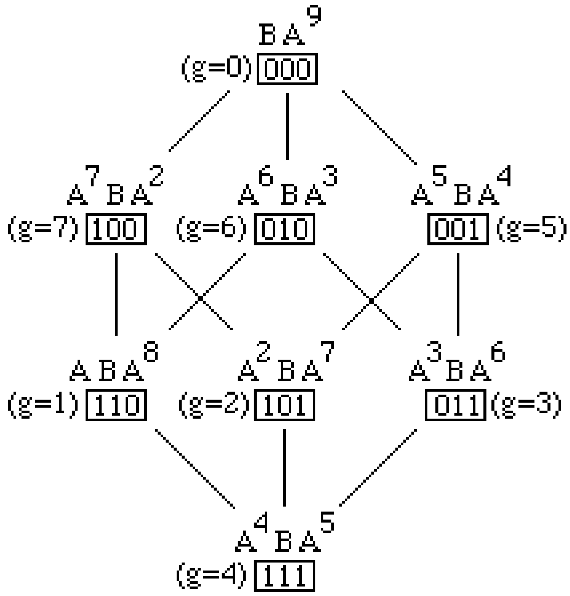
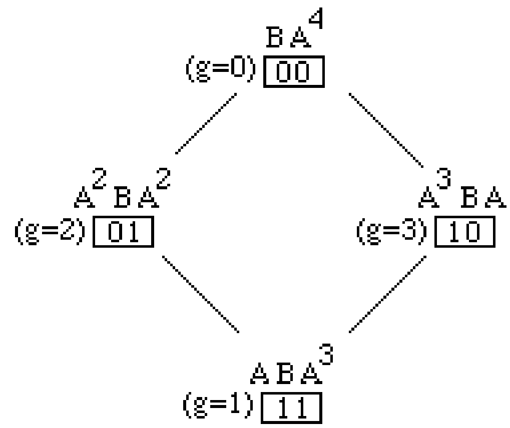


Figure 1: A one-to-many embedding $G \overset{d}{\Rightarrow} H$ seen as a map $f : V(G) \rightarrow 2^{V(H)} - \varnothing$. It is required for each edge xy in G and each x' in f(x) that there exist at least one vertex $y'_{x'}$ in f(y) at distance at most d from x' in H. This is illustrated for the edge st, with $s' \in f(s)$, $t'_{s'} \in f(t)$.



(a)



(b)

Figure 2:

(a) A one-to-many dilation 1 embedding of Q_3 into P_{10} (a B-map).

(b) A one-to-many dilation 1 embedding of Q_2 into P_5 (a B-map).

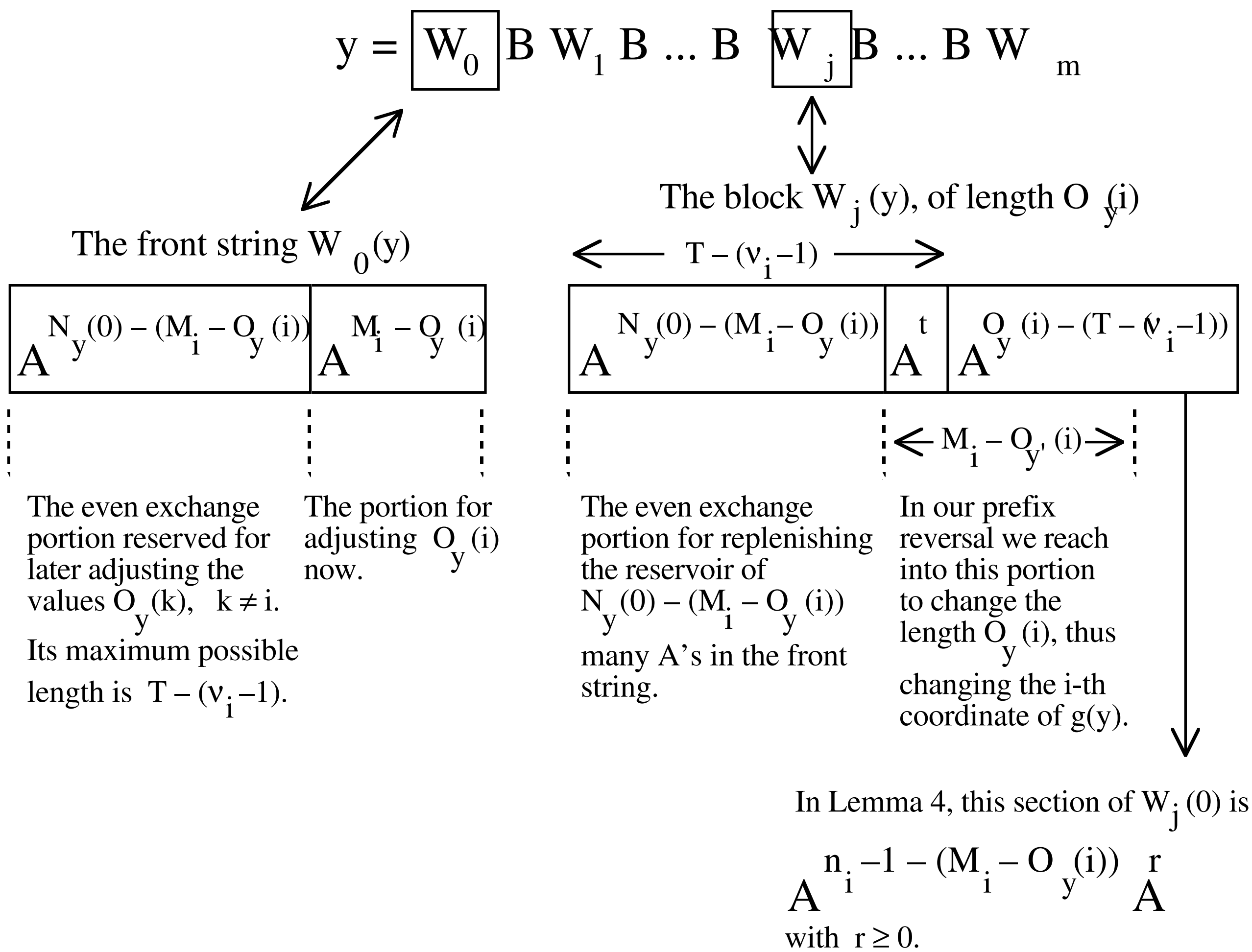


Figure 3: How the front string acts as a reservoir of A's used for adjusting any block to any suitable desired length. By this action a string y for which the i-th coordinate is $M_i - O_y(i)$ is converted to the string y' for which the i-th coordinate is $M_i - O_{y'}(i)$.

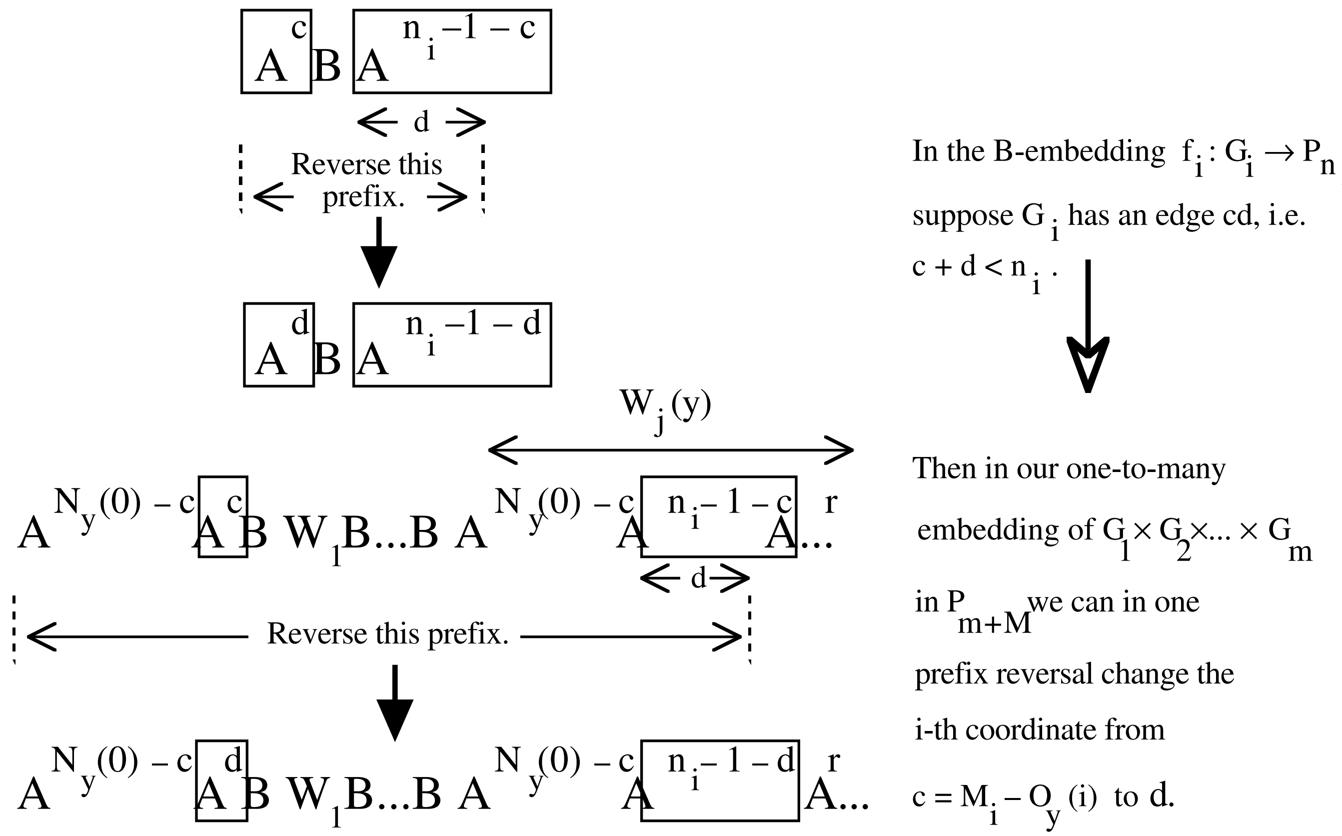


Figure 4: How a B-embedding of each graph G_i is used in constructing a dilation 1 co-embedding of $G_1 \times G_2 \times ... \times G_m$ into P_{m+M} .

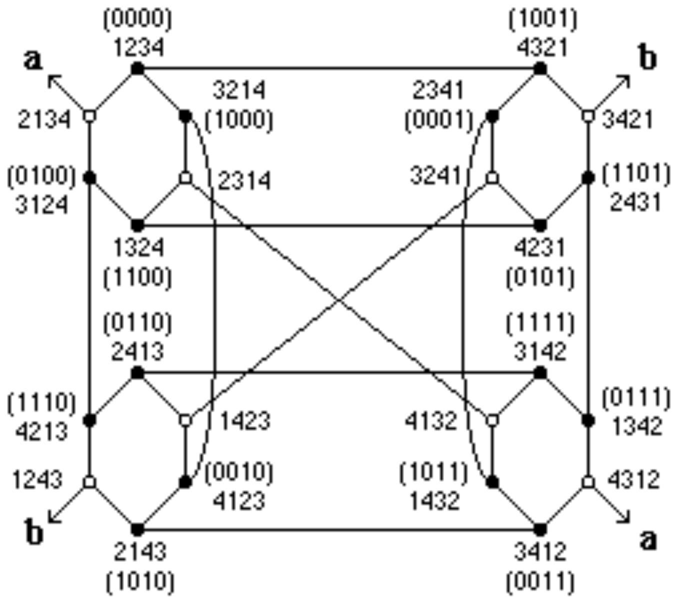
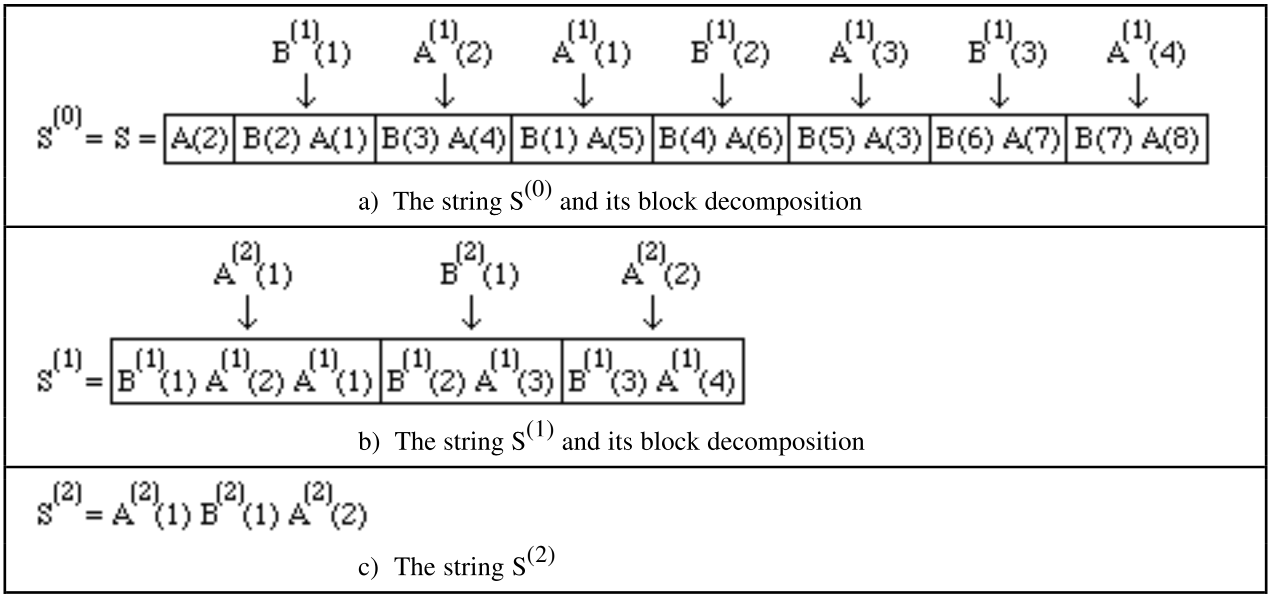


Figure 5: A dilation 2 embedding of Q_4 into P_4.



a) The string $S^{(0)}$ and its block decomposition

b) The string $S^{(1)}$ and its block decomposition

c) The string $S^{(2)}$

Figure 6: Strings $S^{(i)} \in P(2^{k-i}-1)$ derived from a string $S \in P(2^k-1)$