

# A Cayley-Prüfer Type Formula for the Number of Hamming-Labeled Trees

Laurence E. LaForge

The Right Stuff of Tahoe, Incorporated  
 The Right Place, 3341 Adler Court  
 Reno, NV 89503-1263 USA  
 +1.775.322.5186  
 Larry.LaForge@The-Right-Stuff.com

**Abstract.** Prüfer's coding scheme yields a classic proof of Cayley's formula: using  $n$  distinct labels, the number of free-labeled trees on  $n$  vertices equals  $n^{n-2}$ . We introduce an analogous result: using labels of shortest length  $n - 1$ , the number of Hamming-labeled trees on  $n$  vertices equals  $2^{n-1} n^{n-3}$ . These shortest labels have least radix 2 and maximum weight 1. Imposing the proviso that each tree contain some fixed but arbitrary label, we further show that the number of such Hamming labelings reduces to the number  $n^{n-2}$  of free-labeled trees. As a warmup, we enumerate the free labelings of hypermeshes and hypercubes metrized on the Lee, Manhattan, and Hamming distances, and compare with the count of Hamming labelings in each case. We sample pertinent applications, related work, key algorithms, and open problems.

**Keywords and phrases.** Hamming graph; Hamming labeling; tree enumeration; tree labeling; hypercube.

## TABLE OF CONTENTS

<b>1.</b>	<b>LABELS, LABELINGS, AND HAMMING GRAPHS ...</b>	<b>1</b>
<b>2.</b>	<b>BINARY HAMMING GRAPHS .....</b>	<b>7</b>
<b>3.</b>	<b>WHIRLWIND TOUR: APPLICATIONS, RELATED WORK, ALGORITHMS, AND OPEN PROBLEMS ...</b>	<b>20</b>
	<b>REFERENCES .....</b>	<b>28</b>

Text: 16¼ pages. 13 Figures: 5½ pages. 8 Tables: 6½ pages.

## 1. LABELS, LABELINGS, AND HAMMING GRAPHS

### 1.1 Basic Vocabulary

Tables 1, 2, 3, and 5 index definitions and notation. For the sake of exposition, we elaborate certain of these in the text.

Consider the complete set  $L_j$  of all labels on  $d$  ordered digits radix  $j$ . For  $L_j$  or any of its nonempty subsets  $\lambda$ , the  $q^{\text{th}}$  digit takes its value from  $j_q \geq 2$  symbols. Each of  $j_q$  symbols appears in the  $q^{\text{th}}$  digit of at least one label of  $\lambda$ ; *i.e.*,  $\lambda$  is not padded. Without loss of generality, we deem digit  $q$  as taking on integer values from 0 to  $j_q - 1$ . Integer  $j_q$  is the *extent* of digit  $q$ . The ordered  $d$ -tuple  $\mathbf{j} = (j_{d-1}, \dots, j_0)$  of extents is the *mesh* (or *mixed*) *radix vector* of  $L_j$ .

Unless specified otherwise, our radices are *extent-majorized*. For extent positions  $q$  and  $i$  in  $\mathbf{j}$ , that is,  $q \geq i$  implies  $j_q \geq j_i$ . The position of a label digit is the same as that of its extent. For reasons we illuminate in Section 1.2, the maximum extent  $j_{d-1}$  of  $\mathbf{j}$  is the *cube radix* of  $L_j$ . The radix is *j-ary*, or *uniform*, if all the extents of  $\mathbf{j}$  equal  $j$ ; we may emphasize this fact by replacing vector  $\mathbf{j}$  with the scalar extent  $j$ .

The (unweighted) *Hamming distance*  $|x, y|_H$  equals the number of digits where the  $d$ -digit labels  $x$  and  $y$  differ. In Section 3.2 we touch on weighted Hamming distances. The *edge distance*  $|x, y|_G$  equals the minimum pathlength between vertices labeled  $x$  and  $y$  in a connected graph.

Graph  $G$  is *Hamming-labeled* if the edge distance between every two vertices equals the Hamming distance between the respective labels:  $|x, y|_G = |x, y|_H$ , for all  $x, y \in V(G)$ . A graph is *Hamming* if it can be Hamming-labeled. Unless specified otherwise, our labelings have minimum dimension, and their majorized radices are extent- or cube-minimum. The *minimum Hamming dimension* (*dimension*, for short) of graph  $G$  is the fewest number  $d$  of digits that suffice to Hamming-label  $G$ , using some (unpadded) subset  $\lambda \subseteq L_j$  of the complete set of  $d$ -digit labels, for some radix  $j$ .

Suppose that graph  $G$  is Hamming-labeled with majorized radix  $\mathbf{j}$  and dimension  $d$ . Vector  $\mathbf{j}$  is *extent-minimum* if, for any other majorized  $d$ -tuple  $\phi$  that Hamming-labels  $G$ ,  $j_q \leq \phi_q$ , for integers  $0 \leq q < d$ . Vector  $\mathbf{j}$  is *cube-minimum* if it minimizes the cube radix  $j_{d-1}$ . If the labeling is  $j$ -ary uniform then extent and cube minima are equivalent.

While many graphs are Hamming, many are not. So, and as a function of the order  $n$ , just how many Hamming-labeled graphs are there? Alas, and as summarized in Table 8, this question falls among those to which sharp answers remain an open challenge. However, we can (and do) explicate the number of Hamming-labeled *trees*. Table 6 and Table 7 catalog the contributions served up by this paper.

$ x, y _G$	<i>Edge distance</i> , a.k.a. <i>graph distance</i> : minimum pathlength between vertices $x$ and $y$ .	page 1
$ x, y _H$	<i>Hamming distance</i> : the number of digits where $d$ -digit labels $x$ and $y$ differ.	1
$ x, y _L$	<i>Lee</i> , a.k.a. <i>modulo</i> , a.k.a. <i>cyclic distance</i> : $\sum_{0 \leq q < d} \min( x_q - y_q , j_q -  x_q - y_q )$	4, 8
$ x, y _M$	<i>Manhattan</i> , a.k.a. <i>city-block distance</i> : $\sum_{0 \leq q < d}  x_q - y_q $	4, 10
<i>connected graph G</i>	Graph $G$ is <i>connected</i> if there is a path between every two vertices of $G$ .	1
<i>cube-minimum radix</i>	Mesh radix vector $\mathbf{j}$ is <i>cube-minimum</i> if it minimizes the cube radix $j_{d-1}$ .	1
<i>cube radix</i>	Maximum extent $j_{d-1}$ in majorized radix vector $\mathbf{j}$ of dimension $d$ .	1
<i>cycle; C<sub>q</sub></i>	A path of length $q - 1 \geq 2$ , together with an edge joining the endpoints of the path.	2
<i>digit</i>	Positionally distinct variable in a set of labels. If the <i>extent</i> of the $q^{\text{th}}$ digit is $j_q$ , then the $q^{\text{th}}$ digit assumes any of $j_q$ integer values from 0 to $j_q - 1$ inclusive.	1
<i>extent of a digit</i>	The number of symbols that a digit can take on as its value.	1
<i>extent-minimum radix</i>	Minimizes all the extents over all majorized radices that Hamming-label a graph.	1
$G(E, V); V(G); E(G)$	( <i>Simple, undirected</i> ) <i>graph G</i> comprising <i>edges E</i> that prescribe the image of a symmetric irreflexive relation on a finite set $V$ of <i>vertices</i> .	1
<i>(minimum) dimension</i>	Fewest digits sufficient to Hamming-label (or, generally, <i>metrize</i> ) a graph.	1
<i>Hamming graph</i>	A graph is <i>Hamming</i> if it can be Hamming-labeled.	1
<i>Hamming labeling</i>	Graph $G$ is <i>Hamming-labeled</i> if the edge distance between every two vertices equals the Hamming distance between the respective labels: $ x, y _G =  x, y _H, \forall x, y \in V(G)$ .	1
<i>j-ary; j-ary radix</i>	Labeling with respect to vector radix $\mathbf{j}$ ; <i>uniform</i> in scalar extent $j$ : $\mathbf{j} = (j, \dots, j)$ .	1
$\mathbf{j}^d$	Shorthand for $\prod_{0 \leq q < d} j_q$ . Equal to $ V(K_{\mathbf{j}}) $ and $ L_{\mathbf{j}} $ . Reduces to $j^d$ in the $j$ -ary case.	4
<i>K-cube; K<sub>j</sub><sup>d</sup></i>	<i>Clique-based hypercube; complete j-ary d-dimensional Hamming graph</i> . For $d = 1$ we may write $K_j$ for a <i>clique</i> , a.k.a. <i>hyperedge</i> , a.k.a. <i>complete graph of order j</i> .	4, 4
<i>K-mesh; K<sub>j</sub></i>	<i>Clique-based hypermesh; complete Hamming graph radix j</i> .	4
<i>label x</i>	Ordered tuple of <i>digit values</i> , the $q^{\text{th}}$ of which we denote $x_q$ .	1
$L_{\mathbf{j}}; L_{\mathbf{j}}^d; \boldsymbol{\lambda}$	Unpadded set $L_{\mathbf{j}}$ of labels radix $\mathbf{j}$ resp. $j$ -ary radix dimension $d$ ; $\boldsymbol{\lambda}$ is a subset of $L$	1, 1
<i>majorized</i>	(Radix) vector $\mathbf{j}$ is <i>majorized</i> if $d > q \geq i \geq 0$ implies $j_q \in \mathbf{j} \geq j_i \in \mathbf{j}$ .	1
<i>order</i>	Vertex cardinality. If the vertex set of a graph is $V$ , we write its <i>order</i> as $n =  V $ .	4
<i>path; P(x ... z)</i>	A single <i>endpoint</i> vertex $x = P(x \dots x)$ ; else union $P(x \dots z)$ of path $P(x \dots y)$ with edge $(y, z)$ , such that $y$ and $z \notin P(x \dots y)$ are endpoints of $P(x \dots y)$ resp. $P(x \dots z)$ .	2
<i>pathlength</i>	Size of a path.	2
<i>(mixed) mesh radix j</i>	Majorized vector $\mathbf{j} = (j_{d-1}, \dots, j_0)$ ; $j_q$ is the <i>extent</i> of digit $q$ ; $d$ is the <i>dimension</i> .	1
<i>size</i>	Edge cardinality. If the edge set of a graph is $E$ , we write its <i>size</i> as $ E $ .	2
<i>subgraph of G</i>	Any subset of the edges of $G$ , unioned with any subset of the vertices of $G$ .	5
<i>tree</i>	Connected graph with minimum size. Equivalently, a cycle-free connected graph.	1
$ X ;  x $	<i>Cardinality</i> , or number of elements in the set $X$ . <i>Absolute value</i> of real number $x$ .	2, 2

Table 1: Nomenclature and notation introduced, directly or indirectly, through the first part of Section 1.2. Also see most any textbook on graphs; e.g., [Bollobás 1998], [Chartrand and Lesniak 1986]. Also see Tables 2, 3, and 5.

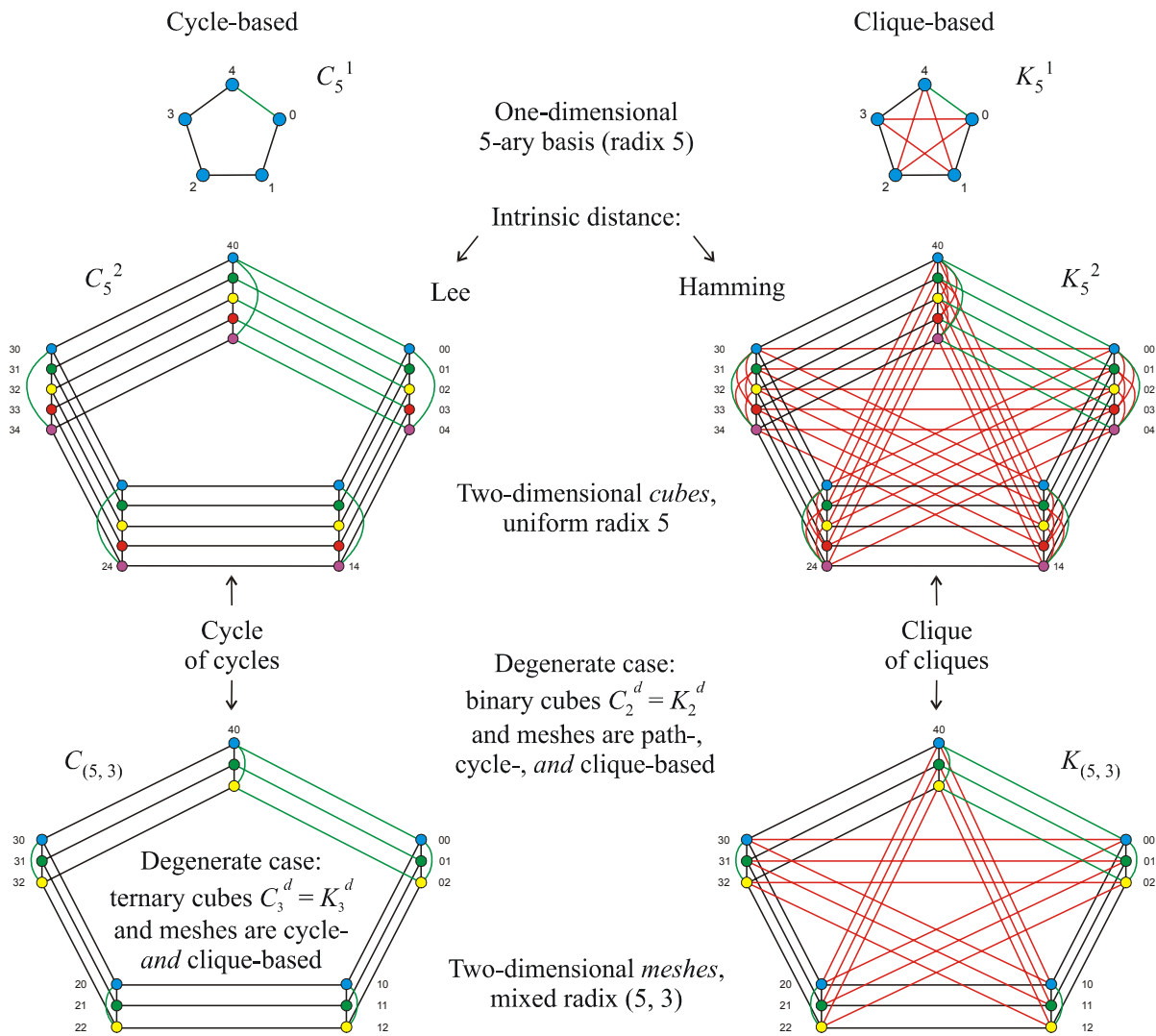


Figure 1: Dimension, radix, and labelings of cubes and meshes. Deleting the green edges from the cycle-based constructions on the left yields a path, or path-based cube or mesh, whose intrinsic distance is Manhattan.

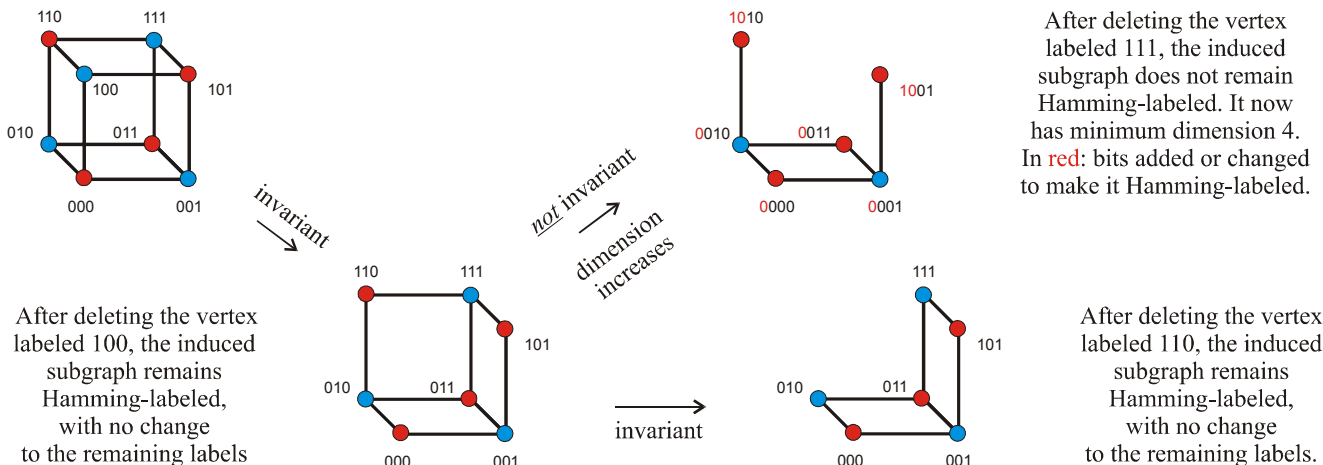


Figure 2: Graphs induced from the 3-dimensional Hamming-labeled binary cube: invariant versus dimension-increasing radix.

### 1.2 K-Meshes and K-Cubes are Hamming Ideals

To count the number of Hamming graphs, it pays to know something about them. Table 4 chronicles a selection of key properties and results. For the sake of exposition, we elaborate certain of these in the text. For example, what does the anatomical blueprint of Hamming graphs look like?

To answer this question, first consider the *complete Hamming-labeled graph* radix  $j$ , also known as a *clique-based hypermesh*  $K_j$ , or *K-mesh*. To construct  $K_j$ , attach a vertex to each element in the complete set  $\mathbf{L}_j$  of all labels whose mesh radix is  $j$ ; join two vertices with an edge if and only if the Hamming distance between their labels equals one. The number  $|\mathbf{L}_j|$  of such labels equals

$$\prod_{0 \leq q < d} j_q \stackrel{\text{def}}{=} j^d \tag{1}$$

... whence (1) also expresses  $|V(K_j)|$ , the order of  $K_j$ . The shorthand on the righthand side will prove quite convenient.

If  $\mathbf{L}$  is  $j$ -ary then the uniform K-mesh so constructed specializes to a *clique-based hypercube*,  $K_j^d$ , a.k.a.  $j$ -ary *K-cube* of dimension  $d$ . In this case (1) reduces to

$$j^d = |V(K_j^d)| = |\mathbf{L}_j^d| \tag{2}$$

Equivalently, and as illustrated in Figure 1, we can Hamming-label K-meshes (hence K-cubes) by induction on the dimension. The basis at  $d = 0$  is an unlabeled vertex. At  $d = 1$  we have a *clique*, or *complete graph*, of order  $j_0$  and size  $\frac{1}{2}(j_0^2 - j_0)$ ; i.e.; a  $j_0$ -*hyperedge*. To get a K-mesh of radix  $(j_{d-1}, \dots, j_0)$ , Hamming-label  $j_{d-1}$  copies of a K-mesh whose radix is  $(j_{d-2}, \dots, j_0)$ . For integer values of  $q$  ranging from 0 to  $(j_{d-1} - 1)$ , prepend each label on the  $q^{\text{th}}$  copy with  $q$ ; join two vertices if and only if they are in different copies and their labels are identical in digits 0 through  $d - 2$ .

If the extent of each digit equals 2 then we have the familiar binary hypercube [Armstrong and Gray 1981]; e.g., Figure 2. If any Hamming extent exceeds three, however, then clique-based meshes and cubes differ from cycle-based C-meshes  $C_j$  and C-cubes  $C_j^d$  popularized by the application-oriented literature [Bose et al 1995], [LaForge et al 2003].

The preceding algorithm correctly constructs a complete Hamming-labeled K-mesh or K-cube, regardless of the order by which we pivot on the digits ([LaForge 2004] p. 4). It generalizes to procedures that generate meshes and cubes from other bases, such as paths and cycles (see Figure 1). Of interest are the corresponding intrinsic distances; e.g., the Manhattan distance, for path-based P-meshes  $P_j$  and P-cubes  $P_j^d$ , and the Lee or modulo distance, in the case of cycle-based C-meshes and C-cubes ([LaForge et al 2003] Thm 9). With respect to any such distance, what graphs are *homomorphically metrizable* on the set  $\mathbf{L}_j$  of labels radix  $j$ ?

More precisely, graph  $G$  is  $\phi$ -*metrizable* with respect to metric space  $\lambda$  on the distance function  $|\cdot, \cdot|_{\Delta}$  if there exists a one-to-one mapping  $\phi$ , or *metrized labeling*, between  $V(G)$  and  $\lambda$ , such that  $|x, y|_G = |\phi(x), \phi(y)|_{\Delta}$  for all  $\phi$ -labeled vertices  $x, y \in V$ . When is there such a metrization  $\phi$ ? How many metrizations  $\phi$  and inverse metrizations  $\phi^{-1}$  are there? Though independent, these notions are similar to many of those considered by [Graham and Winkler 1985].

For the particular case of Hamming labelings,  $|\cdot, \cdot|_{\Delta} = |\cdot, \cdot|_{\mathbf{H}}$ , and  $\lambda$  is a subset of the complete set  $\mathbf{L}_j$  of labels radix  $j$ , or, in the  $j$ -ary uniform case, a subset of the complete set of  $d$ -dimensional  $j$ -ary labels. Without loss of generality, and for reasons set forth below, we stipulate that  $\mathbf{L}_j$  is the smallest such complete set containing  $\lambda$ .

To see that  $\lambda \subseteq \mathbf{L}_j$  is a metric space, note that  $K_j$  isomorphically preserves Hamming distance, that the edge distance on any connected graph induces a metric space ([Chartrand and Lesniak 1986] Chap. 2), and that  $K_j$  is connected. Check that  $|\cdot, \cdot|_{\mathbf{H}}$  on  $\lambda$  satisfies the requisite conditions for a metric space. For all  $x, y, z \in \lambda$ , that is: i)  $|x, y|_{\mathbf{H}} \geq 0$ ; ii)  $|x, y|_{\mathbf{H}} = |y, x|_{\mathbf{H}}$ ; iii)  $|x, y|_{\mathbf{H}} = 0$  if and only if  $x = y$ ; iv)  $|x, z|_{\mathbf{H}} \leq |x, y|_{\mathbf{H}} + |y, z|_{\mathbf{H}}$ . The latter follows by our  $\mathbf{L}_j \leftrightarrow K_j$  isomorphism, notwithstanding any jumps out of, and back into, the image of  $\lambda$ .

Any  $\lambda \subseteq \mathbf{L}_j$  has one or zero Hamming metrizations. The Hamming distance maps  $\lambda$  into a unique graph  $G$ , such that each component of  $G$  is Hamming-labeled by one of the Gray-code-transitive blocks that partition  $\lambda$  (Table 4.a);  $\lambda$  is *Hamming graphic* if and only if there is but one such block. [LaForge 2004] introduces Algorithm  $A_{\text{Construct-Hamming}}$ , an  $O(d|\lambda|^2)$  running-time procedure for determining if  $\lambda$  is Hamming graphic. Since the input size is  $\Omega(d \cdot |\lambda|)$ , this leaves us with the open challenge of bridging the  $\Theta(|\lambda|)$  multiplicative gap between problem lower bound and algorithmic upper bound. (Cf. open problems, Table 8.h, i.)

It is somewhat more formidable to characterize metrization in the forward direction: for given unlabeled graph  $G$ , what  $\lambda$ , if any, Hamming-labels  $G$ , and how? To this end, our own recent discoveries about Hamming graphs come in handy. K-meshes are *Hamming ideals* in the sense that every Hamming-labeled graph radix  $j$  is induced by successively deleting vertices from  $K_j$  (Table 4.b). Such a sequence of induced graphs is *radix-invariant*, *invariant* for short (see Figure 2). In other words, if  $G$  is Hamming, then it has one and only one mesh ideal  $K_j$ . This K-mesh is the complete Hamming graph, of minimum dimension and extent, that contains  $G$ . In terms of order and size, therefore,  $K_j$  is the smallest K-mesh that invariantly induces  $G$ , and  $\mathbf{L}_j$  is the smallest complete set of labels whose elements Hamming-label  $G$ . These results specialize to K-cubes as Hamming ideals on a cube radix. (Cf. research avenue, Table 8.b.)

$\lceil x \rceil$	<i>Ceiling</i> , or least integer not less than the real number $x$ .	page 7
$O(g(x))$	Set of functions no greater than $c_1 \cdot g(x)$ , for constants $c_1, x_1$ and all $x > x_1$	4
$\Theta(g(x))$	<i>Exact order of magnitude</i> : intersection of $O(g(x))$ and $\Omega(g(x))$	4
$\Omega(g(x))$	Set of functions no less than $c_2 \cdot g(x)$ , for constants $c_2, x_2$ and all $x > x_2$	4
$C$ -cube; $C_j^d$	<i>Cycle-based hypercube</i> ; <i>complete <math>j</math>-ary <math>d</math>-dimensional graph on the Lee distance</i> . Basis at $d = 1$ is a the cycle $C_j$ .	4, 8
$C$ -mesh; $C_j$	<i>Cycle-based hypermesh</i> ; <i>complete graph on the radix <math>j</math> Lee distance</i> .	4, 8
<i>component of graph <math>G</math></i>	Maximal connected subgraph $Q$ of $G$ ; there is no edge between $Q$ and $G \setminus Q$ .	4
<i>cover of set <math>S</math></i>	Set of sets whose union contains $S$ ; the cover is <i>exact</i> if the union equals $S$ .	5
<i>EDJ sets</i>	Sets are ( <i>pairwise</i> ) <i>edge-disjoint</i> : if no (two) sets share a common edge.	7
<i>Gray-code adjacent</i>	Labels $x \neq y$ are <i>Gray-code adjacent</i> if $ x, y _{\mathbb{H}} = 1$ , else <i>Gray-code nonadjacent</i> .	6
<i>Gray-code transitive set of labels</i>	The set $\{x\}$ comprising label $x$ is Gray-code transitive. If $y$ is Gray-code adjacent to some element of a Gray-code transitive set $\lambda$ then $\lambda \cup \{y\}$ is Gray-code transitive.	4
<i>edge-induced</i>	A subgraph $Q$ of $G$ is <i>edge-induced</i> if $Q$ spans $G$ .	5
<i>factorization <math>E</math> of graph <math>G</math>; weak; partial; incomplete; primality; prime</i>	Partition $E = \{E_0, \dots, E_{d-1}\}$ of $E(G)$ such that each <i>factor</i> $E_q$ spans $V(G)$ . If $E_q$ need not span $V(G)$ , then $E_q$ is a <i>weak factor</i> of $G$ , and $E$ is a <i>weak factorization</i> of $G$ . If $E$ need not cover $G$ , then the matching $E$ is a <i>partial</i> , a.k.a. <i>incomplete factorization</i> of $G$ . The <i>primality</i> of $E$ is the number $ E $ of factors. If $ E $ is minimum then $E$ is <i>prime</i> .	7, 7, 7
<i>Hamming graphic</i>	Label set $\lambda$ is <i>Hamming graphic</i> if the Hamming graph that metrizes $\lambda$ is connected.	4
<i>hyperseparator, hyperedge <math>q</math>-separator</i>	Subset $H$ of the hyperedges (which may be edges) of a connected graph $G$ whose removal from $G$ edge-induces $q > 1$ components. We don't remove vertex $x$ in the edge-induced subgraph, even if all of $x$ 's neighbors belong to a hyperedge in $H$ .	7
<i>independent sets</i>	Set of pairwise disjoint sets: the intersection of every two sets is empty. Vertices with no edges in common ( <i>i.e.</i> , edge-complementary to a clique) exemplify independence.	5
<i>induced subgraph of <math>G</math></i>	Subgraph obtained by deleting vertices of $G$ (along with all edges impinging on them)	4
<i>matching of set <math>S</math></i>	An independent packing of $S$ .	5
<i>metric space <math>(S, \Delta)</math></i>	Set $S$ with real-valued <i>distance</i> $ \cdot, \cdot _{\Delta}$ such that, for all $x, y, z \in S$ : i) $ x, y _{\Delta} \geq 0$ ; ii) $ x, y _{\Delta} =  y, x _{\Delta}$ ; iii) $ x, y _{\Delta} = 0$ if and only if $x = y$ ; iv) $ x, z _{\Delta} \leq  x, y _{\Delta} +  y, z _{\Delta}$ .	4
<i>metrization <math>\phi</math>; metrized labeling; inverse metrization <math>\phi^{-1}</math></i>	One-to-one mapping $\phi$ , or <i>labeling</i> , between the vertices of graph $G$ and the metric space of labels $\lambda$ on distance $ \cdot, \cdot _{\Delta}$ , such that $ x, y _G =  \phi(x), \phi(y) _{\Delta}$ for all $\phi$ -labeled vertices $x, y \in V$ . The inverse $\phi^{-1}$ isomorphically maps labels to graphs.	4
<i>packing of set <math>S</math></i>	Set of sets whose union is contained in $S$ ; <i>complete</i> if the union equals $S$ .	5
<i>partition of set <math>S</math></i>	Complete matching of $S$ . The independent sets in the matching are called <i>blocks</i> .	5
<i>pivot (verb)</i>	To remove a digit or hyperedge factor, thus reducing the dimension of an instance.	4
$P$ -cube; $P_j^d$	<i>Path-based hypercube</i> ; <i>complete <math>j</math>-ary <math>d</math>-dimensional graph on the Manhattan distance</i> . Basis at $d = 1$ is the path $P_j$ .	4, 10
$P$ -mesh; $P_j$	<i>Path-based hypermesh</i> ; <i>complete graph on the radix <math>j</math> Manhattan distance</i> .	4, 10
<i>radix invariant graph with <math>K</math>-mesh ideal <math>K_j</math></i>	Element of a sequence of induced subgraphs of $K_j$ , each containing its successor, and each of which remains Hamming-labeled on the identical mesh radix $j$ as $K_j$ .	4
<i>cube-monotone graph with <math>K</math>-cube ideal <math>K_j^d</math></i>	Element of a sequence of induced subgraphs of $K_j^d$ , each containing its successor, each with cube radix $j$ , and each of whose mesh radix is no less than its successor's	4
<i>spanning graph of <math>G</math></i>	Set of edges that covers the vertices of graph $G$ . Such a set of edges is said to <i>span</i> $G$ .	5

Table 2: Nomenclature and notation introduced, directly or indirectly, beginning with the latter part of Section 1.2.

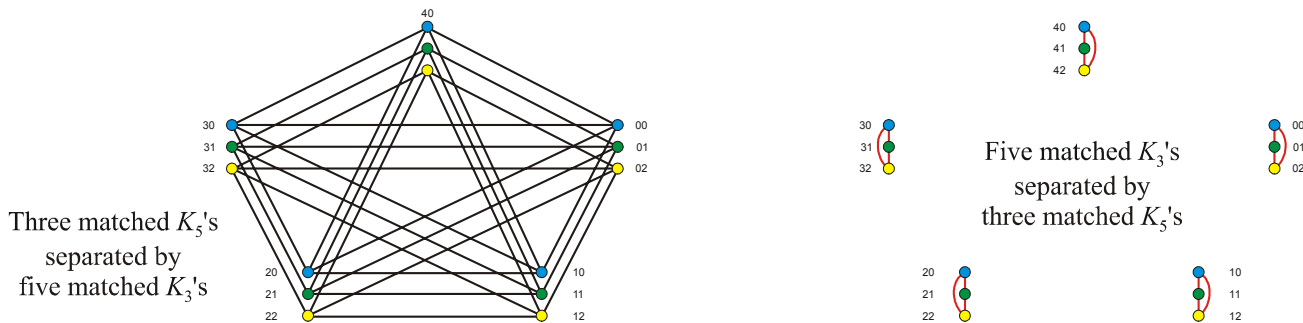


Figure 3: Prime factorization of  $K_{(5,3)}$ , the K-mesh of Figure 1, illustrating the anatomical blueprint of Table 4.d. Each of  $d = 2$  factors is a matching of hyperedges that separate Gray-code adjacent vertices.

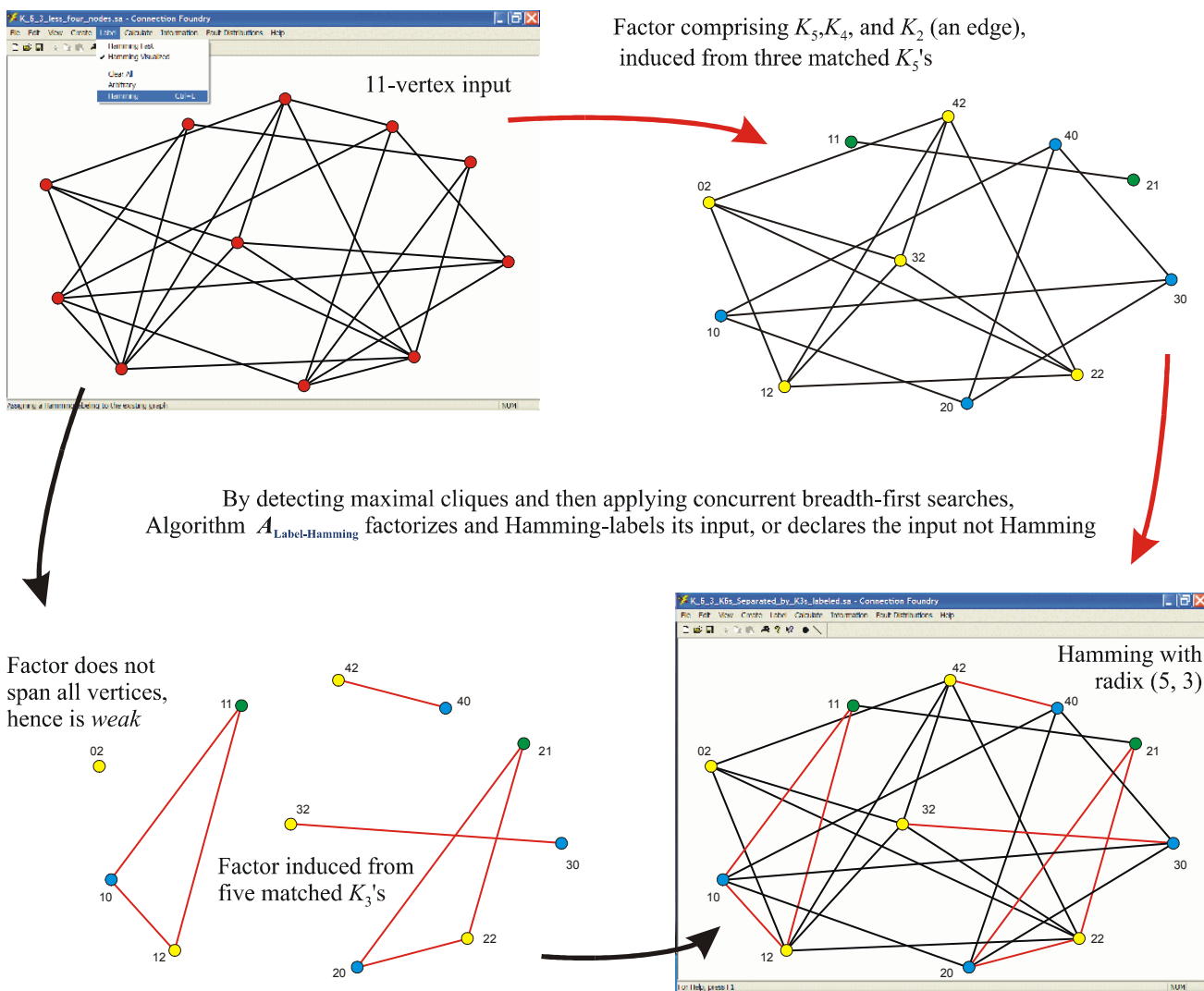


Figure 4: Prime factorization of an 11-vertex radix-invariant subgraph of the 15-vertex K-mesh  $K_{(5,3)}$  of Figure 1. By the anatomical blueprint of Table 4.d, the ideal factorization shown in Figure 3 is weakly preserved. The practical upshot: an efficient algorithm,  $A_{Label-Hamming}$ , that correctly Hamming-labels any graph, or declares it not Hamming. Screenshots shown here illustrate  $A_{Label-Hamming}$  as implemented in The Right Stuff of Tahoe's Connection Foundry software.

### 1.3 Hamming Graphs Factorize as Hyperseparators

Picking up where Section 1.2 left off, let us dissect Hamming graphs in a bit more detail. Hamming graph  $G$  is necessarily induced from its  $K$ -mesh ideal, in a radix-invariant fashion. Conversely, an invariantly-induced graph preserves the Hamming metrization of its ideal (Table 4.b), so this condition is also sufficient for  $G$  to be Hamming.

The recursive definition set forth in Section 1.2 tells us that growing a  $(d-1)$ -dimensional  $K$ -mesh to dimension  $d$  introduces exactly  $j^{d-1}$  hyperedges, each of which spans exactly  $j_{d-1}$  vertices. These  $j^{d-1}$  hyperedges are edge-disjoint (EDJ). As Figure 3 illustrates, they separate  $j_{d-1}$  copies of the underlying  $(d-1)$ -dimensional  $K$ -mesh in a strong, pairwise fashion. Each  $K$ -mesh edge belongs to one such hyperedge. Therefore, a  $K$ -mesh *factorizes* into *separators*, with each factor itself a *matching* of hyperedges. The factorization has minimum cardinality  $d$ , hence is *prime*.

As a complement to (1), we have the Hamming ideal *size*:

$$|E(K_j)| = \frac{1}{2} d(\underline{j}_d - 1) j^d ; |E(K_j^d)| = \frac{1}{2} d(j-1) j^d \quad (3)$$

... where we have applied the result and statistics-flavored notation of Table 4.c. We may drop the subscript on  $\underline{j}_d$  when the averaging range is clear. Summing over all  $j^d$  degrees  $d(\underline{j}_d - 1)$ , the factor  $\frac{1}{2}$  adjusts for our having counted each edge twice. Because the integer  $d(\underline{j}_d - 1) j^d$  is even, we can omit the ceiling, or least-integer-not-less-than function  $\lceil \cdot \rceil$ , from (3). To wit: either some factor of  $j^d$  is even, or all of the factors of  $j^d$  are odd; in the latter case all summands of  $d(\underline{j}_d - 1)$  are even.

Table 4.d strengthens the preceding from  $K$ -meshes to *all* Hamming graphs  $G$ . As Figure 4 exemplifies, any radix-invariant sequence of induced graphs preserves the hyperseparator prime factorization of its  $K$ -mesh ideal. Radix-invariant deletion permits all but one of the  $j_q$ -hyperedges corresponding to digit  $q$  to be eroded. Unlike the  $K$ -mesh ideal, each hyperseparator matching need not span  $G$ , and the factorization is *weak*. Since the radix is invariant, however, there must remain at least one such  $j_q$ -hyperedge. This result yields Algorithm  $A_{\text{Label-Hamming}}$ , illustrated by Figures 4 and 8, and discussed in Section 3.3.

Analogous hyperfactorization carries over in the case of  $K$ -cubes.  $G$  has Hamming dimension  $d$  and cube radix  $j$  if and only if  $G$  is induced from its  $j$ -ary  $d$ -dimensional  $K$ -cube ideal  $K_j^d$  in a *cube-monotone* (as opposed to radix-invariant) fashion. To expound on Figure 2, cube-monotone deletion of vertices never increases the extent, and the cube radix remains constant at  $j$ . The low-order extents in the radix vector can (and generally will) decrease to match the mesh radix of  $G$ . One consequence of this (Table 4.f) is that the cube radix of a Hamming graph equals its clique number.

## 2. BINARY HAMMING GRAPHS

Hamming labelings radix 2 are necessarily extent-minimum, and render majorization moot. For certain binary Hamming graphs, moreover, the relation between the order  $n$  and the minimum Hamming dimension  $d$  is well-understood. For example, even cycles  $n = 2d$  are binary Hamming, and have minimum dimension  $d$  (Table 4.i);  $n$ -vertex trees are binary Hamming with minimum dimension  $n-1$ , a result which specializes to paths (Table 4.m; also Lemma 2 herein).

Attractive facts such as these spur the chief combinatorial contributions of this paper. Table 6 summarizes formulae we derive for the number of Hamming-labeled  $C$ -meshes (including cycles and  $C$ -cubes); the number of Hamming-labeled  $P$ -meshes (including paths and  $P$ -cubes); and, as advertised in the title, the number of Hamming-labeled trees.

### 2.1 Labelings: Distinct, Free, and Metrized

Labeled graphs  $G$  and  $Q$  are the *same*, or *identical*, if their vertices and edges are equal:  $V(G) = V(Q)$  and  $E(G) = E(Q)$ . Otherwise,  $G$  and  $Q$  are *distinct*. By the number of labeled graphs, we mean the maximum cardinality of a set of (pairwise) distinct labeled graphs. Section 2.4 revisits graph representation from a computational viewpoint.

The challenge of finding an expression for the number of labeled graphs with given properties is of longstanding interest. In its classic version, we parameterize the order  $n$ , and seek to enumerate graphs whose vertex labels are a set of  $n$  successive integers. Without loss of generality we take that set as  $L_n^1$ , *i.e.*, the integers from 0 to  $n-1$  inclusive. Here there is no *a priori* requirement for metrization, and we christen such a labeling as *free*.

Let  $\Phi(\mathbf{G})$  denote the set of free labelings of a set  $\mathbf{G}$  of graphs; write  $\Phi(\mathbf{G})_\Delta$  for the shortest labelings of  $\mathbf{G}$ , metrized on distance  $|\cdot, \cdot|_\Delta$ . It is interesting to contrast the cardinality of free labelings with that of their metrized counterparts.

For example, suppose that  $\mathbf{G}$  is the  $n$ -cycle  $C_n$ . Then

$$\text{For } n \geq 3: \quad |\Phi(C_n)| = \frac{1}{2} (n-1)! \quad (4)$$

To see this, assign the label 0 to an arbitrary vertex. Choose either of two neighboring vertices of 0, and assign it any of the  $n-1$  remaining labels. Continue in this direction, now determined, with  $n-q$  choices for labeling the  $q^{\text{th}}$  vertex,  $0 < q < n$ . The result (4) follows by the product rule of counting ([Comtet 1974] pp. 4, 231), and by noting that we have counted each labeling twice. By contrast:

$$\begin{aligned} \text{Theorem 1. } |\Phi(C_n)_H| &= 1 && \text{for } n = 3 \\ &= 0 && \text{for odd } n \geq 5 \\ &= \frac{1}{2} (d!) = \frac{1}{2} ([n/2]!) && \text{for } n = 2d \geq 4 \end{aligned} \quad (5)$$

**Proof.**  $C_3 = K_3$ , whence there is but a single one-dimensional Hamming labeling radix three. Otherwise,  $n \geq 4$ , and the requirement that the order  $n$  of the cycle be even follows from Table 4.k or, albeit less directly, from Table 4.e.

It remains to consider even  $n = 2d \geq 4$ . By the result of Table 4.l, not only is  $C_{2d}$  binary Hamming on fewest bits  $d$ , but any  $d$ -bit Hamming labeling of  $C_{2d}$  is determined by any  $d$  successive vertices: vertex  $x$  corresponds to its ones-complement antipode  $\bar{x}$ , at diametric distance  $d$ , halfway around the cycle. By dimension-minimality,  $\mathbf{0}$ , the label with all  $d$  bits clear, *must* appear, as must its antipode, the label  $\mathbf{1}$  with each bit set. The label on a vertex at distance  $q$  from  $\mathbf{0}$  has  $q$  bits set. Drawing from the theory of codes,  $q$  is the *weight* of the vertex ([MacWilliams and Sloane 1998] p. 8).

Assign the label  $\mathbf{0}$  to any vertex. Both of  $\mathbf{0}$ 's neighbors must have weight one. There are  $\frac{1}{2}(d^2 - d)$  ways to accomplish this, leaving  $d - 2$  bits clear in the partial labeling so far. The two interior-disjoint (IDJ) paths between  $\mathbf{0}$  and  $\mathbf{1}$  are directional and distinguishable, say, as *high*, for the path whose label on the neighbor of  $\mathbf{0}$  has a bit set in a higher order position than that of the other, *low* path. If  $d = 2$  then we are done: the number of binary Hamming labelings of  $C_4$  equals one, which is the exact value of (5). If  $d = 3$  then we are done as well: the number of binary Hamming labelings of a  $C_6$  equals three, again the exact value of (5).

Otherwise,  $d \geq 4$ ; let  $q = 2$  and proceed as follows. Assign to the next unlabeled vertex in the high path the same label as its predecessor; then set one of the  $d - 2$  bits which have so far remained clear, and increment  $q$ . Continue in this fashion, assigning a total of  $d - 1$  such labels, not counting  $\mathbf{0}$  and  $\mathbf{1}$ , but including the high and low neighbors of  $\mathbf{0}$  initially set with weight one. Alternate between high and low paths, assigning to the next unlabeled vertex the same label as its predecessor on the same path, then setting a bit which has so far remained clear. At the  $q^{\text{th}}$  assignment there remain  $d - q + 1$  clear bits from which to choose. Lest we inconsistently overwrite the antipode of a vertex, the last assignment occurs at  $q = d - 1$ . The result (5) follows by the product rule of counting.  $\square$

## 2.2 Labelings of C-Meshes and C-Cubes

To generalize Theorem 1, let us first specify exactly what we mean by a *Lee-labeled* cycle-based mesh, or C-mesh  $C_j$  with mixed radix  $j$ . Assign each label of  $L_j$  to one of  $j^d$  vertices, where  $d$  is both the Lee dimension of  $C_j$  and the number of Lee digits in  $L_j$ . Join with an edge two vertices labeled  $x$  and  $y$  if and only if  $x$  and  $y$  differ in one digit, say, the  $q^{\text{th}}$ , and the modulo- $j_q$  difference between the respective values in digit  $q$  equals  $\pm 1$ . That is, vertices share an edge if and only if they are unit neighbors with respect to the *Lee distance* [Bose et al 1995], [LaForge et al 2003].

Equivalently, and as illustrated in Figure 1, we can Lee-label C-meshes by induction on the dimension. The basis at  $d = 0$  is an unlabeled vertex. At  $d = 1$  we have a cycle of order and size  $j_0$ . To get a C-mesh of radix  $(j_{d-1}, \dots, j_0)$ , Lee-label  $j_{d-1}$  copies of a C-mesh whose radix is  $(j_{d-2}, \dots, j_0)$ . For integer values of  $q$  ranging from 0 to  $(j_{d-1} - 1)$ , prepend each label on the  $q^{\text{th}}$  copy with  $q$ . Join two vertices with an edge if and only if their labels are adjacent with respect to the Lee distance. As with K-meshes, this works correctly regardless of the order by which we pivot on the digits. If  $L$  is  $j$ -ary then the extent-uniform C-mesh specializes to a *cycle-based hypercube*,  $C_j^d$ , a.k.a.  $j$ -ary *C-cube* of dimension  $d$ . At  $j = 2$  we have  $C_2^d = K_2^d$ . Since  $C_4^d = K_2^{2d}$ , a 4-ary Lee digit is equivalent to two bits [LaForge et al 2003].

While the orders  $j^d$  of  $C_j$  and  $K_j$  are equal (1), the sizes are, in general, not equal. Verify and contrast with (3):

$$|E(C_j)| = \frac{1}{2}(2d - b)j^d, \quad b = |\{\text{binary extents of } j\}| \quad (6)$$

**Lemma 1.**  $C_j$  is binary Hamming if and only if no Lee extent is odd, whence  $C_j$  has minimum Hamming dimension

$$\frac{1}{2}d[\underline{l}_d], \quad \text{and where } \underline{l}_d \text{ is as defined in Table 4.c.} \quad (7)$$

**Proof.** If some Lee extent equals three then  $C_j$  contains a triangle, in which case some Hamming extent must equal three, and  $C_j$  is not binary Hamming. The requirement that any extent greater than three be even follows from Table 4.k or, albeit less directly, from Table 4.e.

Supposing that no Lee extent is odd, we Hamming-label  $C_j$  by induction on the Lee dimension  $d$ . For a basis at  $d = 1$  we either have an edge whose endpoints are labeled 0 and 1, or a cycle of order and size  $j_0$ . In the latter case we apply Theorem 1. Either way, we obtain a binary Hamming labeling of minimum dimension  $\frac{1}{2}[j_0] = \frac{1}{2}d[\underline{l}_d]$ . To Hamming-label a C-mesh with Lee radix  $(j_{d-1}, \dots, j_0)$ , inductively Hamming-label  $j_{d-1}$  copies of a C-mesh whose Lee radix is  $(j_{d-2}, \dots, j_0)$ . For integer values of  $q$  ranging from 0 to  $(j_{d-1} - 1)$ , prepend each Hamming label on the  $q^{\text{th}}$  copy with the  $\frac{1}{2}[j_{d-1}]$ -bit Hamming label corresponding to Lee label  $q$  in the base case, applying Theorem 1 if  $j_{d-1} \geq 4$ , else assigning complementary bit values if  $j_{d-1} = 2$ . The edge distance between vertices equals the Hamming distance between their respective labels. The number of bits equals  $\text{DIAM}(C_j)$ ; by the result of Table 4.g, the labeling has minimum Hamming dimension as given by (7).  $\square$

**Remark 1.** The preceding generalizes to C-meshes whose Lee extents are either three or even: ternary extents are identically Hamming, and the remaining sub-mesh – with even extents only – is binary Hamming per Lemma 1.

Lemma 1 corrects Theorem 6 of [LaForge et al 2006]. The latter should have referred to P-meshes, not to C-meshes.

**Theorem 2.** Let  $C_j$  be the  $d$ -dimensional C-mesh on Lee radix  $j$  having  $b$  binary,  $t$  ternary, and  $f$  4-ary extents, and such that all other extents are even. With  $\mathbf{j}_d$  as in Table 4.c:

$$|\Phi(C_j)_H| = 1 \quad \text{if } b + f = d - t \quad (8)$$

$$\text{else} \quad = 2^{b+t+f-d} [\frac{1}{2}(d\mathbf{j}_d - 3t)]! \quad (9)$$

**Proof.** At  $d = 1$ : if  $b + t + f = 1$  then we have (8); else (9) pertains, and gives equality by (5), Theorem 1.

For  $d \geq 2$ , and recalling that radices are majorized, the  $t$  ternary digits of Lee radix  $j$  are also the high-order  $t$  digits of a minimum Hamming labeling of  $C_j$ . Since labels are not padded, these high-order digits prescribe a  $t$ -dimensional ternary C-cube (equivalently, a ternary K-cube), wherein the digits are indistinguishable. Thus, ternary digits contribute a factor 1 to the number of Hamming labelings. By Remark 1, it suffices to enumerate all  $[\frac{1}{2}(d\mathbf{j}_d - 3t)]$ -bit labelings of the  $(d - t)$ -dimensional sub-mesh of  $C_j$  whose extents are even.

As first noted on page 8,  $C_4^d = K_2^{2d}$ , whence we replace any 4-ary Lee digit with two bits. Analogous to the case with ternary digits, the  $b$  binary and  $f$  4-ary digits of the Lee radix  $j$  prescribe a  $(b + 2f)$ -dimensional binary cube, wherein the corresponding  $b + 2f$  bits are indistinguishable. Any Lee digit whose extent is 2 or 4 contributes a factor 1 to the number of Hamming labelings, and  $b + f = d - t$  implies (8).

Otherwise, the remaining  $d - b - t - f > 0$  Lee digits have even extents greater than or equal to six. Dimension-minimality implies that  $\mathbf{0}$ , the label with all  $\frac{1}{2}(d\mathbf{j}_d - 3t)$  bits clear, *must* appear. Invoking Lemma 1, we choose  $\frac{1}{2}[j_q]$  bits corresponding to the  $q^{\text{th}}$  Lee digit, and use Theorem 1 to Hamming-label the associated cycle as it emanates from  $\mathbf{0}$ . The number of ways of choosing these bits times the number of Hamming labelings of the cycle is given by the  $q$ -indexed factors in (10). Cycles so-labeled and emanating from  $\mathbf{0}$  completely determine all bit values on each vertex of  $C_j$ . Each such labeling is Hamming, hence the number of minimum Hamming labelings of  $C_j$  is given by the product:

$$|\Phi(C_j)_H| = \prod_{b+t+f < q \leq d} \left( \frac{1}{2} [q\mathbf{j}_q - 3t] \right) \frac{1}{2} \binom{j_{q-1}}{2}! \quad (10)$$

Expanding and canceling factors, (10) simplifies to (9).  $\square$

**Remark 2.** To count the number of un-majorized minimum Hamming labelings of  $C_j$ , replace  $t$  by 0 in (8) and  $3t$  by  $t$  in (9). Either way, a Hamming labeling of  $C_j$  need not preserve the order of its Lee labeling. For Lee digit values in extent-majorized positions  $q$  and  $i$ , that is,  $q \geq i$  implies  $j_q \geq j_i$ . However, binary values in bits  $p$  and  $r$  of a minimum Hamming labeling of  $C_j$ , and that correspond to Lee-digit positions  $q$  resp.  $i$ , may be such that  $p < r$ .

To illustrate Lemma 1 and Theorem 2: a) the cycle, C-cube, and C-mesh of Figure 1 all have an extent equal to five, hence there are no Hamming labelings. b) For the three-dimensional binary hypercube of Figure 2, (8) correctly tells us that there is only one Hamming labeling. c)  $C_{(8,6,4)}$  has  $(\frac{1}{2})^2 \cdot 9! = 90720$  Hamming labelings on 9 bits, the fewest possible. d) The C-cube  $C_6^4$  has  $(\frac{1}{2})^4 \cdot 12! = 29937600$  Hamming labelings on 12 bits, the fewest possible.

We round out our comparative mini-study of C-meshes with a multi-dimensional extension of (4).

**Theorem 3.** Let  $C_j$  be the  $d$ -dimensional C-mesh on Lee radix  $j$  having  $b$  binary extents. Then

$$|\Phi(C_j)| = 2^{\lceil b/2 \rceil - d} (j^d - 1)! \quad (11)$$

**Proof.** By induction on the dimension  $d$ , with (4) reducing to (11) at  $d = 1, j_0 > 2, b = 0$ . At  $d = 1, j_0 = 2, b = 1$  we have an edge; there is only one way to free-label it, so (11) holds.

Assuming the theorem holds for dimensions through  $d - 1$ , recursively Lee-label  $C_j$  according to the construction on page 8. Pivoting on the high-order digit, if  $j_{d-1} = 2$  then, by majorization,  $C_j$  is a binary cube. Since  $C_4^q = C_2^{2q} = K_2^{2q}$ , the case of even  $d = b$  reduces to a 4-ary C-mesh with dimension  $\frac{1}{2}[d]$ , and (11) follows by induction.

Otherwise, either  $j_{d-1} > 2$  or  $j_{d-1} = 2$  with  $d = b$  odd.  $C_j$  is a cycle or edge of  $j_{d-1}$   $(d - 1)$ -dimensional sub-C-meshes, each of which has order  $j^{d-1}$ . Assigning the label 0 to an arbitrary vertex distinguishes the  $0^{\text{th}}$  such sub-C-mesh, with

$$\binom{j^d - 1}{j^{d-1} - 1} = \binom{j^d - 1}{j^d - j^{d-1}} \quad (12)$$

remaining choices for labels therein, and where we have invoked the symmetry of the binomial coefficient. If  $j_{d-1} > 2$  then, by induction, there are  $2^{\lceil b/2 \rceil - d + 1} (j^{d-1} - 1)!$  ways of free-labeling the  $0^{\text{th}}$  such sub-C-mesh. If  $j_{d-1} = 2$  then, since  $\lceil b/2 \rceil - d = -\lfloor b/2 \rfloor$ , there are  $2^{\lceil b/2 \rceil - d + 1} (j^{d-1} - 1)! = 2^{\lceil b/2 \rceil - d + 1} (2^{d-1} - 1)!$  ways to free-label it. The edges imposed by the (recursively defined) C-mesh transitively induce a matching between the vertices of the  $0^{\text{th}}$  sub-C-mesh and those of the  $q^{\text{th}}$  sub-C-mesh,  $1 \leq q < j_{d-1}$ , as we traverse the cycle or edge with respect to the  $d^{\text{th}}$  digit of  $j$ . Having labeled the  $0^{\text{th}}$  sub-C-mesh, and for any set of labels chosen for the  $q^{\text{th}}$  sub-C-mesh, the number of such matchings is just the number  $(j^{d-1})!$  of one-to-one mappings between two sets of cardinality  $j^{d-1}$ . At the  $q^{\text{th}}$  sub-C-mesh, we pick  $j^{d-1}$  labels from the  $j^d - (qj^{d-1})$  as-yet unchosen labels, leaving  $j^d - (q + 1)j^{d-1}$  labels from which to choose for the  $(q + 1)^{\text{st}}$  sub-C-mesh. If  $C_j$  is a cycle of sub-C-meshes, then this decision sequence counts each labeling twice. Thus, the number of free labelings of  $C_j$  is the product of these  $q$  factors,  $0 \leq q < j_{d-1}$ , divided by two if  $j_{d-1} > 2$ .

Codifying the preceding,  $|\Phi(C_j)|$  equals (12) times

$$2^{\lceil b/2 \rceil - d} \prod_{1 \leq q < j_{d-1}} \binom{j^d - qj^{d-1}}{j^d - (q+1)j^{d-1}} (j^{d-1})! \quad (13)$$

Canceling factors, (12) times (13) simplifies to (11).  $\square$

To illustrate: a) (11) pegs the number of free labelings of the cycle, C-mesh, and C-cube of Figure 1 at  $\frac{1}{2}(4!) = 12$ ,  $(\frac{1}{2})^2(14!) = 21794572800$ , and, respectively,  $(\frac{1}{2})^2(24!) = 155112100433310000000000$ . b) There are  $\frac{1}{2}(7!) = 2520$  free labelings of the 3-dimensional binary cube of Figure 2. c) The number of free labelings of the ternary cube  $C_3^3 = K_3^3$  equals  $(\frac{1}{2})^3(26!) = 50411432640825700000000000$ .

To recap Section 2.2, a C-mesh is Hamming if and only if it has no odd extent greater than four. If  $C_j$  is Hamming, then a minimum labeling has dimension  $\lceil \frac{1}{2}(d\mathbf{j}_d - 3t) \rceil + t$ , where  $t$  is the number of ternary digits in both the Lee and Hamming labels, and the remaining  $\frac{1}{2}(d\mathbf{j}_d - 3t)$  Hamming digits are binary. While any C-mesh has but one Lee labeling, a C-mesh that is Hamming can be Hamming-labeled in many ways (9), as long as it has (even) extents at least six. Comparing (9) with (11), however, we see that a C-mesh which does have Hamming labelings has considerably fewer of them than it has free labelings. Curious, but why?

Since the available labels are less numerous in the free case ( $j^d$ ) than for binary Hamming labelings ( $\geq 2^{\lceil d\mathbf{j}_d \rceil}$ ), we might anticipate fewer free labelings than Hamming labelings. On the other hand, metrization constrains the choice of Hamming labels. So which of these opposing forces prevails? Do we have more free labelings, or more Hamming labelings? With C-meshes the answer is, "More free labelings". As Section 2.3 reveals, this trend holds for P-meshes as well. Alas, the balance does not always tip this way. As we uncover in Section 2.5, Hamming-labeled trees are more numerous than free-labeled trees of the same order.

### 2.3 Labelings of P-Meshes and P-Cubes

Theorem 2's enumeration of C-mesh Hamming labelings takes but implicit advantage of Hamming ideals and hyperfactorization, as reviewed in Sections 1.2 and 1.3. To count the Hamming labelings of P-meshes, we begin to exploit these properties in earnest.

For the Manhattan-labeled path-based mesh  $P_j$ , or P-mesh with radix  $\mathbf{j}$ , we assign each label of  $\mathbf{L}_j$  to one of  $\mathbf{j}^d$  vertices; the dimension  $d$  equals the number of digits. Join with an edge two vertices labeled  $x$  and  $y$  if and only if  $x$  and  $y$  differ in one digit, say, the  $q^{\text{th}}$ , and the arithmetic difference between the respective values in digit  $q$  equals  $\pm 1$ . That is, vertices share an edge if and only if they are unit neighbors with respect to the Manhattan distance.

Equivalently, we can Manhattan-label P-meshes by induction on the dimension. The basis at  $d=0$  is an unlabeled vertex. At  $d=1$  we have a path of order and size  $j_0$ . To get a P-mesh of radix  $(j_{d-1}, \dots, j_0)$ , Manhattan-label  $j_{d-1}$  copies of a P-mesh whose radix is  $(j_{d-2}, \dots, j_0)$ . For integers  $0 \leq q < j_{d-1}$ , prepend each label on the  $q^{\text{th}}$  copy with  $q$ . Join two vertices with an edge if and only if their labels are adjacent with respect to the Manhattan distance. Similar to the case with K- and C-meshes, this works correctly regardless of the order by which we pivot on the digits. If  $\mathbf{L}$  is  $j$ -ary then the extent-uniform P-mesh so constructed specializes to a path-based hypercube,  $P_j^d$ , a.k.a.  $j$ -ary P-cube of dimension  $d$ . At  $j=2$  we have  $P_2^d = K_2^d = C_2^d$ .

The orders  $\mathbf{j}^d$  of  $P_j$ ,  $K_j$  and  $C_j$  are all equal (1). Since the vertices of  $K_j$  and  $C_j$  are degree-regular at  $d(\mathbf{j}_d - 1)$  resp.  $(2d - |\{\text{binary extents of } \mathbf{j}\}|)$ , it is straightforward to calculate their sizes (3), (6). By contrast, P-meshes are irregular. To see just how irregular they are, let  $b$  be the number of bits in the Manhattan radix  $\mathbf{j}$ ; denote by  $\mathbf{J}[q, b]$  the set of all subsets  $\mathbf{j}[q, b]$  of  $q$  extents drawn from the subvector  $(j_{d-1}, \dots, j_b)$  on the  $d-b$  high-order extents; write  $\Gamma(x)$  for the adjacency of vertex  $x$ . Subject to the convention that the summation in (14) equals one if  $q=0$ , verify that the number of vertices of  $P_j$  with degree  $d+q$  is:

$$\begin{aligned} &|\{x \in V(P_j) : |\Gamma(x)| = d+q\}| \quad 0 \leq q \leq d-b \\ &= 2^{d-q} \sum_{\mathbf{j}[q, b] \in \mathbf{J}[q, b]} \prod_{i \in \mathbf{j}[q, b]} (i-2) \quad (14) \end{aligned}$$

To illustrate: a)  $b=0$  for  $P_{(5,4,3)}$ ; it contains  $2^0 \cdot 3 \cdot 2 \cdot 1 = 6$  vertices of maximum degree  $2d=6$ ;  $2^1 \cdot (3 \cdot 2 + 3 \cdot 1 + 2 \cdot 1) = 22$  vertices of degree 5;  $2^2 \cdot (3+2+1) = 24$  vertices of degree 4; and  $2^3 \cdot 1 = 8$  vertices of minimum degree 3. As a check:  $6 + 22 + 24 + 8 = 60 = 5 \cdot 4 \cdot 3 = \mathbf{j}^d$ . b)  $b=2$  for  $P_{(5,4,3,2,2)}$ ; it contains  $2^2 \cdot 3 \cdot 2 \cdot 1 = 24$  vertices of degree  $2d-2=8$ ;  $2^3 \cdot (3 \cdot 2 + 3 \cdot 1 + 2 \cdot 1) = 88$  vertices of degree 7;  $2^4 \cdot (3+2+1) = 96$  vertices of degree 6; and  $2^5 \cdot 1 = 32$  vertices of degree 5. As a check:  $24 + 88 + 96 + 32 = 240 = 5 \cdot 4 \cdot 3 \cdot 2 \cdot 2 = \mathbf{j}^d$ . c)  $P_3^d$  has 1 vertex of degree  $2d$ , and  $3^d - 1$  of degree less than  $2d$ .

The rather messy exercise above underscores why, instead of attempting to wrangle (14), we derive the P-mesh size by appealing to the inductive definition. To avoid subscripted subscripts in the radix vector  $\mathbf{j}$ , we write  $e_P(j_{d-1}, \dots, j_0)$  and  $n_P(j_{d-1}, \dots, j_0)$  in place of  $|E(P_j)|$  resp.  $|V(P_j)|$ . This yields the recurrence

$$e_P(j_{d-1}, \dots, j_0) = j_{d-1} \cdot e_P(j_{d-2}, \dots, j_0) + (j_{d-1} - 1) \cdot n_P(j_{d-2}, \dots, j_0) \quad (15)$$

Invoking (1), replace  $n_P(j_{d-1}, \dots, j_0)$  by  $\mathbf{j}^d$ . Apply the initial condition  $e_P(j_0) = j_0 - 1$ ; by substitution, verify that

$$e_P(j_{d-1}, \dots, j_0) = \sum_{0 \leq q < d} (j_q - 1) \prod_{0 \leq i \neq q < d} j_i \quad (16)$$

solves (15) for  $e_P = |E(P_j)|$ . To illustrate: the P-mesh with radix  $\mathbf{j} = (5, 4, 3)$  has  $4 \cdot 4 \cdot 3 + 3 \cdot 5 \cdot 3 + 2 \cdot 5 \cdot 4 = 133$  edges.

For P-cubes on a scalar radix  $j$ , the erstwhile daunting equation (16) reduces to the refreshingly simple

$$|E(P_j^d)| = d(j-1)j^{d-1} \tag{17}$$

Comparing (3), (6) and (17) at identical dimension  $d$  and scalar radix  $j$ , we see that the K-cube's size is  $j/2$  that of the corresponding P-cube, and  $(j-1)/2$  that of a  $d$ -dimensional non-ternary  $j$ -ary C-cube. To illustrate at  $d = 3$  and  $j = 4$ : the K-cube has  $\frac{1}{2}(3 \cdot 3 \cdot 4^3) = 288$  edges; the P-cube has size  $3 \cdot 3 \cdot 4^2 = 144$ ; while that for the C-cube is  $3 \cdot 4^3 = 192$ .

In contrast with C-meshes, P-meshes are *always* Hamming:

**Theorem 4.** With Manhattan radix  $\mathbf{j}$ , and  $\mathbf{j}_d$  as in Table 4.c:

$$P_j \text{ is binary Hamming with dimension } d(\mathbf{j}_d - 1) \tag{18}$$

$$|\Phi(P_j)_H| = 2^{d(\mathbf{j}_d - 1)} [d(\mathbf{j}_d - 1)]! \tag{19}$$

**Proof.** By induction on the Manhattan dimension  $d$ . For a basis at  $d = 1$ , the results of Table 4.b, g, and m imply that the Hamming ideal  $K_2^{j-1}$  invariantly induces  $P_j^1$ , whence (18). Thus, the number of  $(j-1)$ -bit Hamming labelings of  $P_j^1$  equals the number of diametric geodesics in  $K_2^{j-1}$ . From an arbitrary vertex  $x$  of  $K_2^{j-1}$  to its ones-complement antipode  $\bar{x}$ , there are  $j-q$  distance-monotone choices for the  $q^{\text{th}}$  edge. By the product rule of counting, therefore, we have  $(j-1)!$  diametric geodesics from any starting vertex. Summing over all  $2^{j-1}$  starting vertices counts each diametric geodesic twice. The number  $2^{j-2}(j-1)!$  of such geodesics equals the number (19) of Hamming labelings on  $j-1$  bits; and  $j-1$  is the minimum dimension (18).

Assuming the theorem to be true for Manhattan dimensions 1 through  $d-1$ , apply the result of Table 4.d to the  $d$ -dimensional  $P_j$ . Each of the  $j_{d-1}-1$  sets of edges that (hyper)separate vertices between two  $(d-1)$ -dimensional P-meshes increments the minimum Hamming dimension; *i.e.*, each such pair adds one to the fewest number of bits that Hamming-label the P-mesh. Constructively achieve this bound by prepending (say)  $j_{d-1}-1$  clear bits to labels on  $j_{d-1}$  copies of the Hamming-labeled  $(d-1)$ -dimensional P-mesh. For  $1 \leq q \leq j_{d-1}-1$ , set the low-order (say)  $q-1$  bits prepended to each label on the  $q^{\text{th}}$  copy.

The labeling so produced is Hamming on a number  $d(\mathbf{j}-1)$  of bits equal to  $\text{DIAM}(P_j)$ . By the result of Table 4.g, this labeling has minimum Hamming dimension (18).

In contrast with C-cubes, the label  $\mathbf{0}$  with all bits clear need *not* appear in a minimum P-mesh Hamming labeling. Therefore, we cannot exactly mirror the counting argument at the heart of our proof of Theorem 2.

In lieu of  $\mathbf{0}$ , however, we can start counting at  $x$ , one of the  $2^d > 0$  corner points (14) of  $P_j$  with minimum degree  $d$ . Choose  $j_q-1$  bits corresponding to the  $q^{\text{th}}$  Manhattan digit, and use the base case at  $d = 1$  to Hamming-label the associated path as it emanates from  $x$ . The number of ways of choosing these bits times the number of Hamming labelings of the path is given by the  $q$ -indexed factors in (20). Paths so-labeled and emanating from  $x$  completely determine all bit values on each vertex of  $P_j$ . Each such labeling is Hamming, hence the number of minimum Hamming labelings of  $P_j$  is given by the product:

$$|\Phi(P_j)_H| = \prod_{0 \leq q < d} \binom{[q+1][\mathbf{j}_{q+1}-1]}{q[\mathbf{j}_q-1]} 2^{j_q-2} (j_q-1)! \tag{20}$$

Expanding and canceling factors, (20) simplifies to (19).  $\square$

To illustrate Theorem 4: a) There are  $2^4 \cdot 5! = 1920$  five-bit Hamming labelings of a 6-vertex path. b)  $P_{(5,4,3)}$  has  $2^6 \cdot 9! = 23224320$  Hamming labelings on 9 bits, the fewest possible. c) Using the least number 8 of bits, the P-cube  $P_3^4$  can be Hamming-labeled in  $2^6 \cdot 8! = 2580480$  different ways.

We conclude our comparative mini-study of P-meshes with

**Theorem 5.** Let  $P_j$  be the  $d$ -dimensional P-mesh on Manhattan radix  $\mathbf{j}$  with  $b$  binary extents. Then

$$|\Phi(P_j)| = (\frac{1}{2})^{\lfloor b/2 \rfloor + d} (\mathbf{j}^d)! \tag{21}$$

**Proof.** By induction on the Manhattan dimension  $d$ . At  $d = 1$ , we have a path. Start at either endpoint, moving to the other, with  $j_0-q$  choices for labeling the  $q^{\text{th}}$  vertex,  $0 \leq q < j_0$ . The result (21) follows by the product rule, and by noting that we have counted each labeling twice.

Assuming the theorem holds for dimensions through  $d-1$ , recursively Manhattan-label  $P_j$  according to the construction on page 10. Pivoting on the high-order digit, if  $j_{d-1} = 2$  then, by majorization,  $P_j$  is a binary cube. Since  $P_2^d = C_2^d = K_2^d$ , the case of  $d = b$  reduces to (11), routine manipulation of which establishes equality with (21).

Otherwise,  $j_{d-1} > 2$ .  $P_j$  is a path of  $j_{d-1}$   $(d-1)$ -dimensional sub-P-meshes, each of order  $\mathbf{j}^{d-1}$ , with ends distinguished by vertices having only one neighbor with respect to the  $(d-1)^{\text{st}}$  digit. Move from one to the other of these end sub-P-meshes. Similar to the proof of Theorem 3, it follows that

$$|\Phi(P_j)| = \left(\frac{1}{2}\right)^{\lfloor b/2 \rfloor + d} \prod_{0 \leq q < j_{d-1}} \binom{\mathbf{j}^d - q\mathbf{j}^{d-1}}{\mathbf{j}^d - (q+1)\mathbf{j}^{d-1}} (\mathbf{j}^{d-1})! \tag{22}$$

... by induction, the product rule, and noting that we have counted each labeling twice. Expanding and canceling factors, (22) simplifies to (21).  $\square$

To illustrate **Theorem 5**: a) There are  $(\frac{1}{2}) \cdot 6! = 360$  free labelings of a 6-vertex path. b)  $P_{(5,4,3)}$  has  $(\frac{1}{2})^3 \cdot 60!$  free labelings. c)  $P_3^4$  can be free-labeled in  $(\frac{1}{2})^4 \cdot 81!$  ways.

### 2.4 Free Labelings of K-meshes

**Table 8.a** suggests potential investigations of mesh labeling that would expand considerably on **Sections 2.2** and **2.3**. To save stage room for our featured act in **Section 2.5**, we punctuate our comparative mini-study of meshes by enumerating the free labelings (23) of Hamming ideals.

To elucidate the proof of **Theorem 6**, recall one of the standard representations of a graph  $G$ : the *adjacency matrix*  $\Gamma(G)$ . Entry  $(q, r)$  of  $\Gamma(G)$  equals one (or is nonempty) if  $(q, r)$  is an edge of  $G$ ; else entry  $(q, r)$  equals zero (or is empty). Algebraic graph theory has quite a lot to say about binary adjacency matrices, especially their eigenvalues (cf. **Table 4.h**; also [Biggs 1996] prob. 21a, p. 169). Although adjacency matrices are frequently convenient, their  $\Theta(n^2)$  storage space is suboptimal when the average vertex degree scales sub-linearly in the graph order  $n$  ([Corman et al 1993] p. 467).

**Theorem 6.** Let  $K_j$  be the  $d$ -dimensional K-mesh on Hamming radix  $\mathbf{j}$  having  $b$  binary extents. Then

$$|\Phi(K_j)| = \left(\frac{1}{2}\right)^{\lfloor b/2 \rfloor} \frac{(\mathbf{j}^d)!}{\prod_{0 \leq q < d} (j_q)!} \quad (23)$$

**Proof.** By induction on the dimension  $d$ . At  $d = 1$  equation (23) equals one. To see that this is correct, represent a  $j$ -vertex clique  $K_j^1$  by its binary adjacency matrix  $\Gamma(K_j^1)$ , in accordance with the definition above. Permuting  $\Gamma$ 's rows or columns does not change  $G$ 's adjacency. Moreover, sorting  $\Gamma$ 's  $n$  rows and  $n$  columns (say, by increasing label) results in diagonal entries  $(q, q)$  equal to zero, with  $(q, r)$  equal to one for off-diagonal entries  $q \neq r$ . Thus,  $|\Phi(K_j^1)| = 1$ .

Assuming the theorem holds for dimensions through  $d - 1$ , recursively Hamming-label  $K_j$  according to the construction on **page 4**. Pivoting on the high-order digit, if  $j_{d-1} = 2$  then, by majorization,  $K_j$  is a binary cube. Since  $P_2^d = C_2^d = K_2^d$ , the case of  $d = b$  reduces to (11), straightforward manipulation of which establishes equality with (23).

Otherwise,  $j_{d-1} > 2$ , and  $K_j$  is a clique of  $j_{d-1}$   $(d - 1)$ -dimensional sub-K-meshes, each of order  $\mathbf{j}^{d-1}$ . Similar to the decision-sequence argument in the proof of **Theorem 3**, distribute  $\mathbf{j}^{d-1}$  labels into any such sub-K-mesh, the  $0^{\text{th}}$ . To avoid subscripted subscripts in the explicated radix  $\mathbf{j}$ , write  $\Phi_K(j_{d-1}, \dots, j_0)$  in place of  $|\Phi(K_j)|$ . Having labeled the  $0^{\text{th}}$  sub-K-mesh in one of  $\Phi_K(j_{d-2}, \dots, j_0)$  ways, the edges between it and the  $q^{\text{th}}$  sub-K-mesh prescribe a matching.

At the  $q^{\text{th}}$  sub-K-mesh, we pick  $\mathbf{j}^{d-1}$  labels from the  $\mathbf{j}^d - (q\mathbf{j}^{d-1})$  as-yet unchosen labels, leaving  $\mathbf{j}^d - (q+1)\mathbf{j}^{d-1}$  labels from which to choose for the  $(q+1)^{\text{st}}$  sub-K-mesh. Choosing  $\mathbf{j}^{d-1}$  labels for the  $q^{\text{th}}$  sub-K-mesh induces  $(\mathbf{j}^{d-1})!$  matchings.  $|\Phi(K_j)|$  is therefore  $\Phi_K(j_{d-2}, \dots, j_0)$  times

$$\left(\frac{1}{2}\right)^{\lfloor b/2 \rfloor} \prod_{0 \leq q < j_{d-1}} \binom{\mathbf{j}^d - q\mathbf{j}^{d-1}}{\mathbf{j}^d - (q+1)\mathbf{j}^{d-1}} (\mathbf{j}^{d-1})! \quad (24)$$

... all divided by the number  $(j_{d-1})!$  of times that (24) overcounts labelings. The latter follows by noting that each  $(d - 1)$ -dimensional sub-K-mesh chosen amounts to a label in a one-dimensional clique of order  $j_{d-1}$ . As with the base case  $d = 1$ , no two orderings are distinguishable. Since (24) times  $\Phi_K(j_{d-2}, \dots, j_0)$  covers all  $(j_{d-1})!$  orderings of sub-K-meshes, we must divide by  $(j_{d-1})!$ . Contrast this with the proofs of **Theorem 3** and **Theorem 5**, where our factor-of-two adjustment was a consequence of reverse traversals of cycles or paths. Divide  $(j_{d-1})!$  into the inductive value of  $\Phi_K(j_{d-2}, \dots, j_0)$  and multiply by (24). Canceling like factors yields (23).  $\square$

To illustrate **Theorem 6**: a) the K-cube  $K_5^2$  of **Figure 1** has  $25!/(5! \cdot 5!) = 1077167364120210000000$  free labelings. b) There are  $15!/(5! \cdot 3!) = 1816214400$  free labelings of the K-mesh  $K_{(5,3)}$  of **Figure 1**. c) The number of free labelings of  $K_{(3,2,2)}$  equals  $\frac{1}{2}[(12!/(3! \cdot 2! \cdot 2!))] = 9979200$ , the same value given by (11); this is proper since  $K_{(3,2,2)} = C_{(3,2,2)}$ .

### 2.5 Free vs Hamming Labelings of Trees

Let  $T_n$  be a tree of order  $n$ ; write  $\mathcal{T}_n$  for the set of all  $T_n$ . **Sections 2.2** through **2.4** give expressions for the number of labelings – either Hamming or free – of meshes based on cycles, paths, and cliques. The structured,  $d$ -dimensional adjacency of any such mesh is uniquely prescribed by a radix of  $d$  extents. By contrast,  $\mathcal{T}_n$  is both simpler and more complicated. Simpler, since we have but one parameter: the order  $n$ . More complicated since, for  $n \geq 4$ ,  $\mathcal{T}_n$  is a set with more than one tree. We begin with the free labelings of  $\mathcal{T}_n$ :

**Theorem 7.**  $|\Phi(\mathcal{T}_n)| = n^{n-2}$  [Cayley 1889] (25)

We then prove the result advertised in the title of this paper:

**Theorem 8.**  $|\Phi(\mathcal{T}_n)_{\text{H}}| = 2^{n-1} n^{n-3}$  (26)

By a *Cayley-Prüfer type formula*, we mean one which subjectively resembles (25), and which counts something about labelings and trees. In addition to (26), for example,

$$q^{n-q-1} (n-q)^{q-1} \quad (27)$$

is a Cayley-Prüfer type formula; (27) expresses the number of spanning trees of a  $q \times (n - q)$  bipartite clique whose labels are fixed ([Comtet 1974] p. 92).

$\lfloor x \rfloor$	<i>Floor</i> , or greatest integer not greater than the real number $x$ .	page 9
$\square$ ; <i>iff</i>	End of proof; if and only if	8, 22
$\Gamma(x)$	<i>Adjacency</i> of vertex $x$ : all vertices with which $x$ shares an edge.	10
$\Phi(\mathbf{G})_\Delta$	Set of distinct labelings of $\mathbf{G}$ , metrized on distance $ \cdot, \cdot _\Delta$ .	7
$\Phi_K(j_{d-1}, \dots, j_0)$	Number $ \Phi(K_j) $ of free K-mesh labelings, explicated Hamming radix $(j_{d-1}, \dots, j_0)$ .	12
$\mathbf{0}; \mathbf{1}$	(Binary) label with all digits zero (all bits clear); resp. all digits one (all bits set).	8, 8
<i>adjacency matrix</i> $\Gamma(G)$	Entry $(q, r)$ is one (or nonempty) if $(q, r)$ is an edge; else $(q, r)$ is zero (or empty).	12
<i>antipodal, diametric</i>	$x \in S, y \in S$ are <i>antipodal</i> , or <i>diametric</i> , in metric space $(S, \Delta)$ if $ x, y _\Delta = \text{DIAM}(S)_\Delta$ .	8, 8
$b$	Number of binary extents in a radix; number of 2-valued digits in a set of labels.	8, 9
<i>bipartite clique</i>	Bipartite graph on blocks $V_1$ and $V_2$ , complete with all possible $ V_1  \times  V_2 $ edges.	12
<i>bipartite graph</i>	Graph whose vertices can be bipartitioned into blocks, with all edges between blocks.	12
<i>bit</i>	Binary ( <i>i.e.</i> , radix 2) digit. A bit is <i>clear</i> if it equals zero, <i>set</i> if it equals one.	8
<i>Cayley-Prüfer type</i>	Formula subjectively resembling $n^{n-2}$ , and counting something about labels and trees.	12
<i>clique number of G</i>	Order of the largest clique that graph $G$ contains.	7
<i>connectivity, graph G</i>	<i>Vertex (edge) connectivity</i> : $G$ 's minimum $q$ - $r$ vertex (resp. $q$ - $r$ edge) connectivity.	14
<i>connectivity, q-r vertex (resp. edge)</i>	Maximum number of pairwise-EDJ (resp. -IDJ) paths between vertices $q$ and $r$ . $q$ - $r$ -connected is shorthand for $q$ - $r$ -vertex-connected	13, 13
<i>degree of vertex x</i>	Number of vertices $ \Gamma(x) $ with which $x$ shares an edge. $\Gamma(x)$ is the <i>adjacency</i> of $x$ .	14
<i>degree-regular graph</i>	Graph whose vertex degrees are all equal. A graph that is not regular is <i>irregular</i> .	10, 10
<i>diameter, DIAM(S)<math>_\Delta</math></i>	Maximum eccentricity, $\max_{x \in S} \text{ECC}(x)_{S, \Delta}$ , of the metric space $(S, \Delta)$ .	8
<i>distinct vs identical labeled graphs</i>	Labeled graphs $G$ and $Q$ are the <i>same</i> , or <i>identical</i> , or <i>equal</i> , if their vertices and edges are equal: $V(G) = V(Q)$ and $E(G) = E(Q)$ . Otherwise, $G$ and $Q$ are <i>distinct</i> .	7, 7
<i>eccentricity, ECC(x)<math>_{S, \Delta}</math></i>	Maximum distance between $x$ and some other element of the metric space $(S, \Delta)$ .	13
$e_P(j_{d-1}, \dots, j_0)$	Size $ E(P_j) $ of P-mesh $P_j$ with explicated Manhattan radix $\mathbf{j} = (j_{d-1}, \dots, j_0)$ .	10
$f$	Number of 4-ary extents in a radix; number of 4-valued digits in a set of labels. Worst-case number of faults tolerated = one less than the vertex connectivity.	9, 20
<i>free labeling</i>	$n$ -vertex graph $G$ is <i>free-labeled</i> by any matching between $V(G)$ and $\mathbf{L}_n^1$ . $\Phi(\mathbf{G})$ denotes the set of distinct free labelings of the set $\mathbf{G}$ of graphs.	7, 7
<i>geodesic</i>	Minimum length path.	11
<i>Gray-coded graph G; G is Gray-codable if it can be Gray-coded</i>	$G$ is <i>Gray-coded</i> if i) labels on all edge-adjacent vertices are Gray-code adjacent, and ii) all vertex pairs with Gray-code adjacent labels are edge-adjacent. Imposing (i) only, we have a <i>weak Gray coding</i> , which some authors take as the definition of Gray coding (e.g., [Wakerly 1994] pp. 47, 53). While a path or cycle subgraph of a Hamming-labeled K-mesh may not be Gray-coded, it is always weakly Gray-coded.	14, 14
<i>IDJ paths</i>	Paths are <i>interior-disjoint</i> if, apart from their endpoints, the paths do not intersect.	8
$\mathbf{j}[q, b]$ ; $\mathbf{J}[q, b]$	Subset of $q$ extents drawn from the subvector $(j_{d-1}, \dots, j_b)$ on the $d - b$ high-order extents of a $d$ -dimensional Manhattan radix with $b$ bits; $\mathbf{J}[q, b]$ = set of all $\mathbf{j}[q, b]$ .	10, 10
$n_P(j_{d-1}, \dots, j_0)$	Order $ V(P_j) $ of P-mesh $P_j$ with explicated Manhattan radix $\mathbf{j} = (j_{d-1}, \dots, j_0)$ .	10
<i>ones complement</i> $\underline{x}$	Result of setting the clear bits of binary label $x$ , and clearing the bits of $x$ that are set.	8
$t$	Number of ternary extents in a radix; number of 3-valued digits in a set of labels.	9
<i>weight of label x; <math>\mu</math></i>	Number of nonzero digits in $x$ . Vector $\mu$ for taking inner product with an indicator.	8, 26

Table 3: Nomenclature and notation introduced, directly or indirectly, beginning with Section 2.

a.	( <b>Graphs: Gray-codable vs Hamming</b> ). Let $L_j$ be all labels radix $j$ . The Hamming distance maps each $\lambda \subseteq L_j$ into a unique graph $G$ , such that each component of $G$ is Hamming-labeled by one of the Gray-code-transitive blocks that partition $\lambda$ . Any Gray-coded graph comprises Hamming-labeled components. A Hamming graph is just a connected Gray-codable graph.	[LaForge 2004] Thm. 2	
b.	( <b>Existence and uniqueness of Hamming ideal</b> ). Graph $G$ is Hamming if and only if $G$ is invariantly (resp. cube-monotonically) induced from its K-mesh (resp. K-cube) ideal. To wit: suppose $G$ has Hamming dimension $d$ , mesh radix $j$ , and cube radix $j$ . Then $K_j$ (resp. $K_j^d$ ) is the unique <i>mesh</i> (resp. <i>cube</i> ) <i>ideal</i> of $G$ ; that is, the smallest complete Hamming graph from which we can delete vertices in sequence, such that each of the induced subgraphs preserves the edge $\leftrightarrow$ Hamming metrization of its ideal, with invariant (resp. cube-monotone) radix.	[LaForge 2004] Cor. 2.4	
c.	Hamming ideals achieve equality in vertex connectivity $\leq$ edge connectivity $\leq$ min vertex degree $\leq$ max vertex degree [Chartrand and Lesniak 1986] Thm 5.1	$= \begin{cases} d(j-1) & \text{for } K_j; \\ d(j-1) & \text{for } K_j^d \end{cases} \quad \text{where } j_q \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } q = 0 \\ \frac{1}{q} \sum_{0 \leq i < q} j_i & \text{else} \end{cases}$	[LaForge 1999] straightforward extension of Thms 8 and 9
d.	( <b>Matched hyperseparators comprise a weak factorization of any Hamming graph</b> ). Connected graph $G$ is Hamming with dimension $d$ and radix $\mathbf{j} = (j_{d-1}, \dots, j_0)$ , if and only if $G$ prime-factorizes into $d$ matched hyperseparators $\mathbf{H} = \{H_{d-1}, \dots, H_0\}$ such that i) for all vertices $x$ and $y$ in $G$ , ii) $(x, y)$ is an edge of $G$ whenever iii) $(x, y)$ is an edge in $K_j$ .	[LaForge et al 2006] Thm 5	
e.	A triangle-free graph ( <i>i.e.</i> , a graph that contains no $K_3$ ) is Hamming only if it is bipartite.	[LaForge et al 2006] Rem. (11)	
f.	The cube radix of a Hamming graph equals its clique number.	[LaForge et al 2006] Rem. (13)	
g.	The dimension of a Hamming graph is at least its diameter, and at most one less than its order. These bounds are best possible: for any order, there exist graphs which achieve them.	[LaForge 2004] Thm. 3	
h.	Number of vertices at edge distance $q$ in $K_j^d$	$= (j-1)^q \binom{d}{q} =$ multiplicity of $q^{\text{th}}$ eigenvalue $d(j-1) - jq$ of the adjacency matrix of $K_j^d$ , $0 \leq q \leq d$	[LaForge 1999] [Brouwer et al 1989]
i.	Deleting $q < j_0$ vertices from a Hamming-labeled K-mesh $K_j$ of dimension greater than one induces a graph that is Hamming on the remaining $j^d - q$ vertices, with labels unchanged.	[LaForge 2004] Thm. 4	
j.	If $G$ is Hamming then we can Hamming-label $G$ with origin $\mathbf{0}$ , all digits zero, at any vertex.	[LaForge 2004] Lemma 2	
k.	For integer $q \geq 2$ , an odd cycle $C_{2q+1}$ is not Hamming (in the unweighted sense used throughout most of this paper; for uniform weight $\frac{1}{2}$ , cf. Section 3.2, $C_{2q+1}$ <u>is</u> Hamming.)	[LaForge 2004] Thm. 5	
l.	For integer $q \geq 2$ , an even cycle $C_{2q}$ is Hamming with dimension $q$ and radix 2.	[LaForge 2004] Thm. 6	
m.	Any $n$ -vertex tree is Hamming with minimum dimension $n - 1$ and radix 2.	[LaForge 2004] Thm. 7	
n.	Two triangles of a Hamming graph either belong to the same hyperedge, or are edge-disjoint.	[LaForge et al 2006] Lemma 1	
o.	Any two hyperedges of a Hamming graph are edge-disjoint.	[LaForge et al 2006] Cor 2	
p.	A binary Hamming graph is bipartite.	[Aurenhammer and Haguier 1995]	
q.	The Petersen graph is not Hamming (in the unweighted sense used throughout most of this paper; for uniform weight $\frac{1}{2}$ , cf. Section 3.2 and Figure 13, the Petersen graph <u>is</u> Hamming.)	[LaForge et al 2006] Rem. (15)	

Table 4: Hamming graph facts. Unless specified otherwise, properties are with respect to the unweighted Hamming distance.

**Proof of Theorem 7.** Cayley's formula (25) follows from the classic coding scheme of [Prüfer 1918] (see also [Comtet 1974] p. 63 *et seq*). The code records systematic deletion of a tree, from the leaves inward. For completeness, and in keeping with the computational emphasis of contemporary graph theory ([Corman et al 1993] Sec VI), we adapt a proof from [LaForge 2004 MANETs]. Algorithm  $A^{-1}_{\text{Prüfer}}$  below fleshes out the "as the reader should check" part of the proof sketched on page 277 of [Bollobás 1998].

**Algorithm  $A_{\text{Prüfer}}(T_n)$**       % Input: free-labeled tree  
 1) If  $n \leq 2$                       % Output: left-to-right,  
     then output the empty        % least-to-most Prüfer code  
         label and STOP            % Only one tree of order 2  
 2) Let  $q$  be the leaf whose      %  $q \in L_n^1$  denotes both  
     label  $q$  is minimum        % the vertex *and* its label  
 3)     $r$  be  $q$ 's neighbor        % Output a label  $\in L_n^{n-2}$   
 4) Output the label  $r$             % of  $n-2$  labels  $\in L_n^1$   
 5)  $A_{\text{Prüfer}}(T_n - q)$             % Recursive invocation

**Algorithm  $A^{-1}_{\text{Prüfer}}(\lambda_n^{n-2})$**     % Input: label  $\in L_n^{n-2}$   
 1) If  $n = 2$  (*i.e.*,  $\lambda_n^{n-2}$  is the empty label)  
     then output the edge  $(0, 1)$   
         and STOP                  % Only one tree of order 2  
 2) Let  $q$  be the least value of  
      $L_n^1$  that is not a digit of  $\lambda_n^{n-2}$   
 3) Let  $r$  be the value of the leftmost  
     (*i.e.*, high-order) digit of  $\lambda_n^{n-2}$   
 4) Let  $\lambda_{n-1}^{n-3}$  be the rightmost  
     (*i.e.*, low-order)  $n-1$  digits of  $\lambda_n^{n-2}$ , any such  
     digit decremented by one if it is greater than  $q$   
 5) Let  $T_{n-1}$  be the result of incrementing by one every  
     label, whose value is greater than or equal to  $q$ ,  
     on the tree output by  $A^{-1}_{\text{Prüfer}}(\lambda_{n-1}^{n-3})$   
 6) Output  $T_n$ , the result of adding  
     edge  $(q, r)$  to  $T_{n-1}$       % Graft leaf onto tree

By inspection, and for any  $T_n$ , Algorithm  $A_{\text{Prüfer}}$  outputs a unique  $n$ -ary label  $\lambda_n^{n-2}$  of length  $n-2$ : the *left-to-right, least-to-most Prüfer code* of  $T_n$ . It remains to prove, by induction on  $n$ , that Algorithm  $A^{-1}_{\text{Prüfer}}$  (re-)constructs  $T_n$  from  $\lambda_n^{n-2}$ . For a basis, there is but one free-labeled tree on  $n=2$  vertices. The empty label therefore prescribes the tree on the free-labeled edge  $(0, 1)$ , whence line 1 of  $A^{-1}_{\text{Prüfer}}$ .

Assume that  $A^{-1}_{\text{Prüfer}}$  (re-)constructs trees for labels of length  $0, \dots, (n-3)$ , having uniform extents  $2, \dots$  resp.  $(n-1)$ , and input an arbitrary  $n$ -ary label  $\lambda_n^{n-2}$  of length  $n-2$ . At Line 2 we let  $q$  be the least value between  $0$  and  $n-1$  inclusive that is missing (there must be at least two) from  $\lambda_n^{n-2}$ . Lines 3 and 4 remove the leftmost digit  $r$  of  $\lambda_n^{n-2}$ . In the label  $\lambda_{n-1}^{n-3}$  so truncated, Line 4 decrements any digit whose value exceeds  $q$ . Since  $0 \leq q \leq n-1$ , this guarantees that the extent of the truncated label is  $(n-1)$ -ary.

By induction, the recursion at Line 5, with the digit-decremented  $\lambda_{n-1}^{n-3}$  as argument, outputs a unique  $(n-1)$ -vertex tree, free-labeled on  $L_{n-1}^1$ . Also by induction, applying Algorithm  $A_{\text{Prüfer}}$  to this free-labeled tree outputs  $\lambda_{n-1}^{n-3}$ . Returning to the top-level thread of execution, note that the Prüfer code explicates the labels of the *neighbors* of the (outermost) leaves. In particular,  $A_{\text{Prüfer}}$  initially outputs the leftmost label  $r$  if and only if the tree on which  $A_{\text{Prüfer}}$  operates contains a vertex  $r$  that is the neighbor of a leaf whose label  $q$  is less than that of any such leaf. The recursively-constructed  $T_{n-1}$ , with labels incremented in accordance with  $n$ -ary uniform extents, satisfies these conditions. By Lines 2 and 5, any leaf added by the recursive invocation to Algorithm  $A^{-1}_{\text{Prüfer}}$  has a label that exceeds  $q$ . Thus, joining leaf  $q$  to vertex  $r$  at Line 6 completes the construction of  $T_n$ , and Algorithm  $A_{\text{Prüfer}}$ , acting on  $T_n$ , outputs  $\lambda_n^{n-2}$ . (*Cf.* exercise, Table 8.c.)

Each  $n$ -ary label of length  $n-2$  (interpreted as a Prüfer code) corresponds to a unique free-labeled tree of order  $n \geq 2$ , and conversely. Together with the trivial case at  $n=1$ , this one-to-one correspondence implies (25).  $\square$

To illustrate Theorem 7, there are  $7^5 = 16807$  free-labeled trees of order 7. Figure 5 depicts one of these, along with its Prüfer code 11313, as output by Algorithm  $A_{\text{Prüfer}}$ . Figure 6 traces the execution of Algorithm  $A^{-1}_{\text{Prüfer}}$ , as it re-constructs the same tree from its Prüfer code 11313.

At last to the number of Hamming labelings of  $T_n$ . The result of Table 4.m predates that of Table 4.d, and so was proved without the benefit of insights about how all Hamming graphs hyperfactorize. Exploiting these since-discovered facts, we streamline Theorem 7 of [LaForge 2004] as

**Lemma 2.**  $T_n$  is binary Hamming with dimension  $d = n - 1$ .

**Proof.** There is only one path between two vertices, else  $T_n$  is not a minimum-size connected graph, contradicting the definition of tree in Table 1. Thus, any edge is a hyperseparator of  $T_n$ . By the result of Table 4.d, the minimum Hamming dimension is  $T_n$ 's size  $n-1$  ([Bollobás 1998] p. 11). Since an edge is a  $K_2^1$ , the extent-minimum radix is uniformly two.  $\square$

**Remark 3.** Notwithstanding the general-purpose Algorithm  $A_{\text{Label-Hamming}}$ , illustrated in Figure 4 and Figure 8, the result of Table 4.m gives a useful algorithm for Hamming-labeling any tree  $T_n$ . As Figure 7 exemplifies: For  $n=1$ , assign a lone vertex the empty label. For  $n > 2$ : pick any leaf  $q$  of  $T_n$ . Inductively Hamming-label  $T_n - q$ , a tree of dimension  $n-2$  and radix 2. Prepend a cleared bit to each label of  $T_n - q$ . Make the label  $q$  the same as that of its unique neighbor, only set the high-order bit on  $q$ . Since there is only one path between two vertices, this  $(n-1)$ -bit labeling is Hamming.

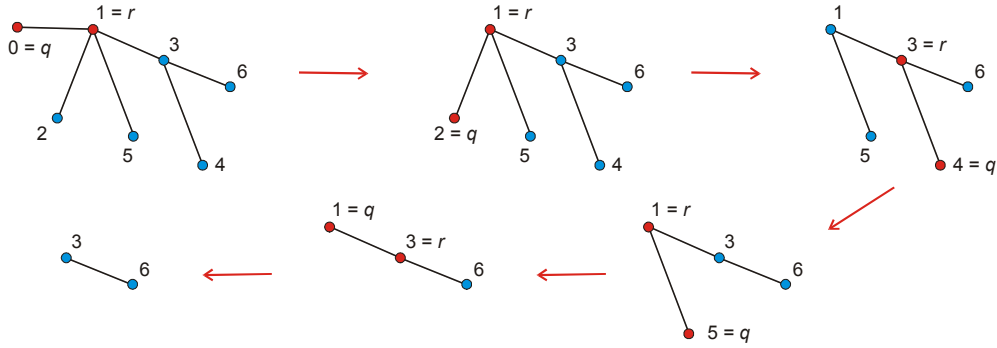


Figure 5: Algorithm  $A_{\text{Prüfer}}$  codes a free-labeled tree of order  $n$ . Shown above: tree with order 7 and Prüfer code 11313.

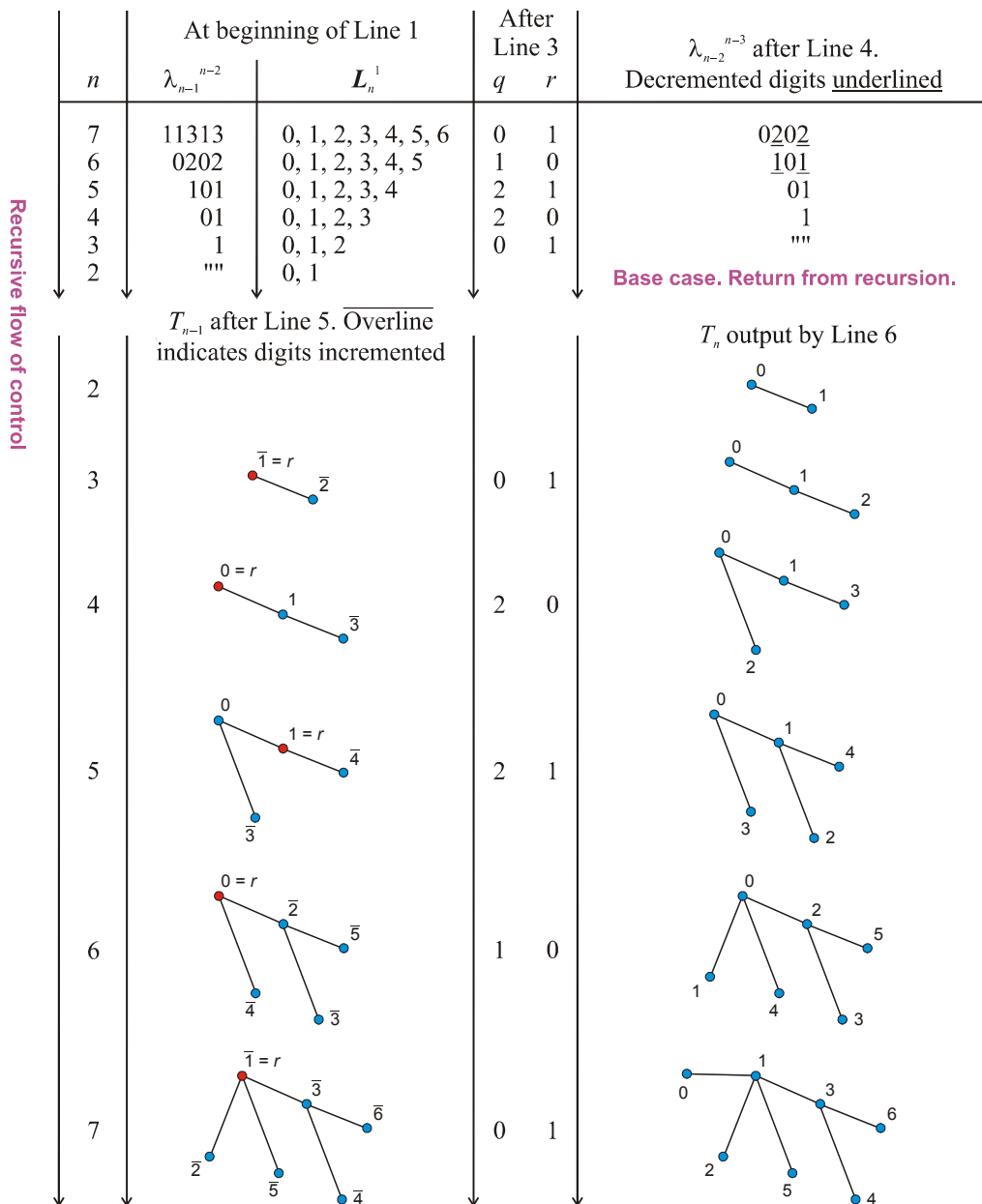


Figure 6: Execution trace of Algorithm  $A_{\text{Prüfer}}^{-1}$ , as it re-constructs the tree of Figure 5 from the Prüfer code 11313.

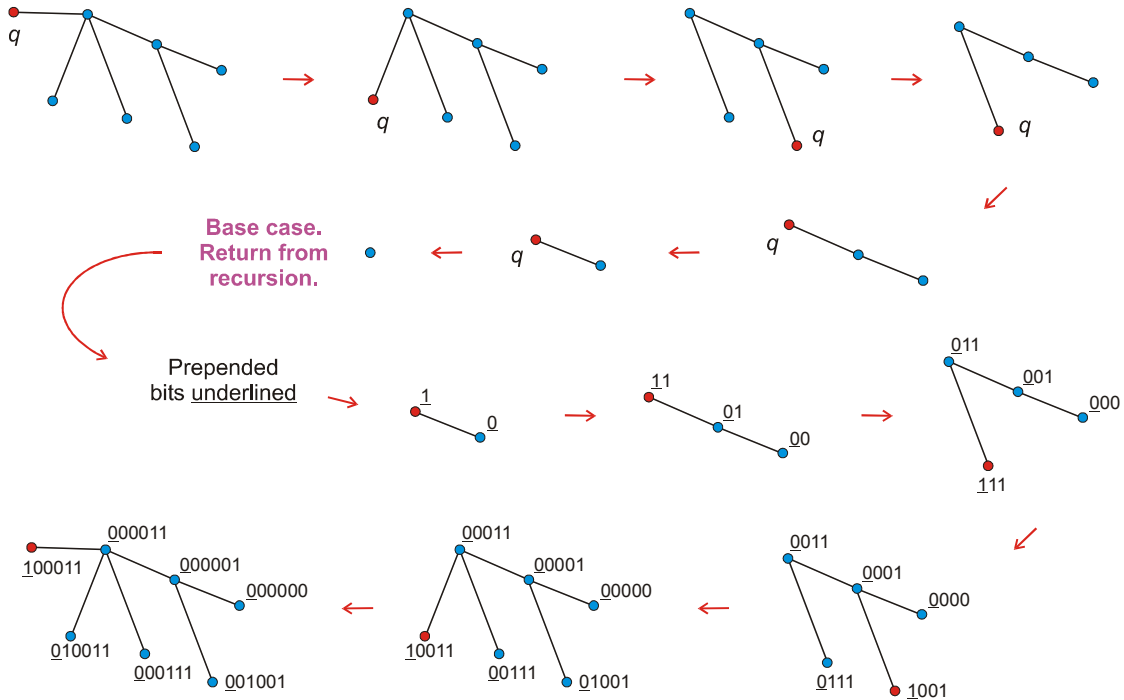


Figure 7: The algorithm of Remark 3 Hamming-labels any tree. Shown here: execution trace for same input as in Figure 5. We can pivot the recursion on the edge of any leaf  $q$ . By the result of Table 4.d, in fact, any edge will do.

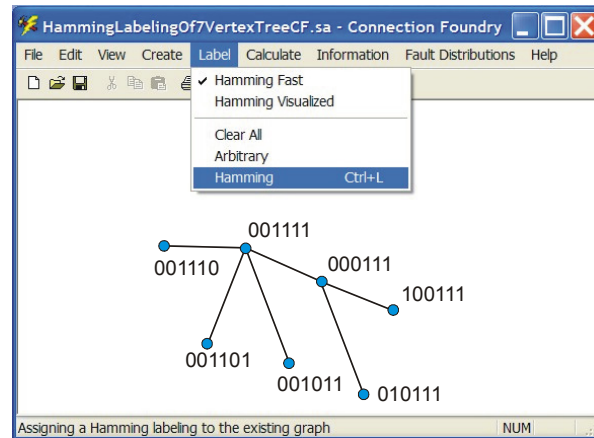


Figure 8: Introduced in [LaForge et al 2006], general-purpose Algorithm  $\mathcal{A}_{\text{Label-Hamming}}$  does not necessarily give the same result as the tree-labeling algorithm of Remark 3. Together with the instance of Figure 7, the above-shown solution, computed by Connection Foundry's implementation of  $\mathcal{A}_{\text{Label-Hamming}}$ , comprise but two of the 153664 seven-vertex Hamming-labeled trees having minimum dimension 6 and extent-minimum radix 2. Compare with Figure 4.



Figure 9: The set  $\Phi(\mathcal{T}_n)_{\text{II}}$  at  $n = 3$ . All  $2^{n-1} n^{n-3} = 4$  extent-minimum Hamming-labeled trees of minimum dimension 2.

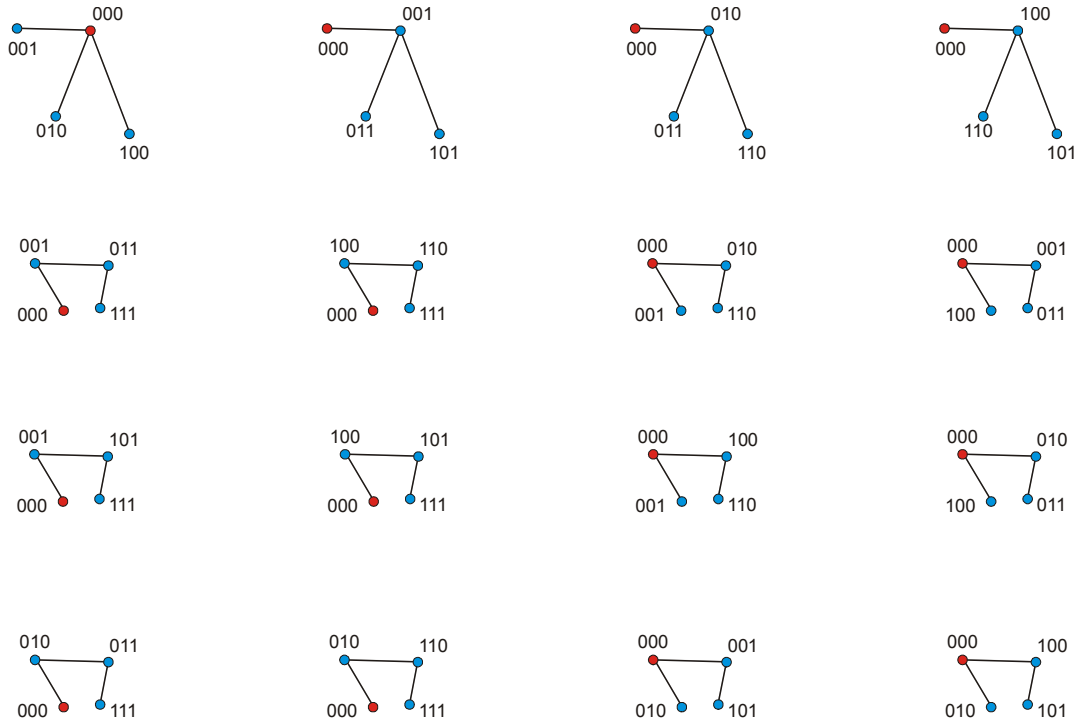


Figure 10: The set  $\Phi_{\mathbf{0}}(\mathcal{T}_n)_{\mathbb{H}}$  at  $n = 4$ . All  $n^{n-2} = 16$  binary-Hamming-labeled trees, of minimum dimension 3, that contain  $\mathbf{0}$ . By Corollary 1, the number of Hamming-labeled trees with a fixed but arbitrary label equals the number of free-labeled trees.

$\lambda ; \lambda_n^{n-2}$	$n$ -ary label of length $n - 2$ . Not to be confused with the label set $\lambda$ (in bold).	page 15
$\Phi_{\mathbf{0}}(\mathcal{T}_{d+1})_{\mathbb{H}}, \Phi_x(\mathcal{T}_{d+1})_{\mathbb{H}}$	Set of $d$ -dimensional Hamming-labeled trees containing $\mathbf{0}$ resp. arbitrary fixed label.	19
$\rho_{\text{MOL}}^-(f, n)$	Lower bound (38) on radius, any minimum-size $(f + 1)$ -connected graph of order $n$ .	20
$\tau(d)$	Number $ \Phi(\mathcal{T}_{d+1})_{\mathbb{H}} $ of $d$ -dimensional Hamming-labeled trees.	19
$\tau_x(d)$	$ \Phi_x(\mathcal{T}_{d+1})_{\mathbb{H}} $ . That is, $\tau(d)$ , such that each tree contains a fixed but arbitrary label.	19
<i>in-, out- degree of <math>x</math></i>	Number of vertices $y$ such that $(y, x)$ resp. $(x, y)$ is an arc.	25, 25
<i>digraph</i>	<i>Directed graph</i> comprising <i>arcs</i> prescribing an irreflexive relation on its vertex set $V$ .	25
<i>fault</i>	Nonfunctioning or untrustworthy processing node, herein modeled by vertex deletion	20
<i><math>(f + 1, g)</math> cage</i>	$(f + 1)$ -regular graph with girth $g$ and minimum order.	20
<i>girth of graph <math>G</math></i>	Minimum size (= minimum order) of any cycle that is a subgraph of $G$ .	20
<i>gracefully degradable set of graphs</i>	Sequence of $(f + 1)$ -connected graphs, whose order grows without bound, such that the diameter of any component induced by deleting up to $f$ vertices is $\Theta(\rho_{\text{MOL}}^-)$ .	20, 21
<i>interior vertex; leaf</i>	Vertex other than a leaf; resp. vertex with degree one, usually in a tree.	20, 15
<b>lg</b>	Logarithm with respect to base 2	25
$n_{\text{Moore}}(f, h)$	Upper bound on the order of any $(f + 1)$ -regular graph, for given $f > 2$ and radius $h$ .	20
$\mathcal{o}(g(q))$	Set of functions $h(q)$ such that $\lim_{q \rightarrow \infty} (h / g) = 0$ .	20
<i>Ore tree</i>	(A.k.a. <i>distance</i> or <i>breadth-first</i> ) spanning tree preserving edge distance from its root.	20
<i>radius, <math>\text{RAD}(S)_{\Delta}</math></i>	Minimum eccentricity, $\min_{x \in S} \text{ECC}(x)_{S, \Delta}$ , of the metric space $(S, \Delta)$ .	20
<i>strongly <math>q</math>-connected</i>	Digraph with $2q$ IDJ paths between $x$ and $y$ , $q$ in each direction, $\forall$ vertex pairs $x, y$ .	25
$T_n, \mathcal{T}_n$	Tree of order $n$ , set of all trees of order $n$ .	12

Table 5: Nomenclature and notation introduced, directly or indirectly, beginning with first proof in Section 2.5.

Continuing in the spirit of Lemma 2, we showcase the Hamming dimension  $d$ . With respect to the result of Table 4.b, for example, each tree of  $T_n$ , Hamming-labeled on  $d$  bits, spans  $d+1$  vertices of its Hamming ideal  $K_2^d$ . Similarly, we will prefer *dimension* to *size*, mindful that Lemma 2 tells us that these two quantities are equal.

**Proof of Theorem 8.** By the result of Table 4.j, we can replace any of  $d+1$  vertices in a Hamming-labeled tree with  $\mathbf{0}$  (the label of  $d$  cleared bits), then adjust the remaining labels to assure that the modified tree is also Hamming-labeled. Conversely, consider the set  $\Phi_0(\mathcal{T}_{d+1})_{\mathbb{H}}$  of  $d$ -dimensional Hamming-labeled trees that contain  $\mathbf{0}$ . Transform each tree in this set into another Hamming-labeled tree by replacing  $\mathbf{0}$  with an arbitrary label  $x$  from  $\mathcal{L}_2^d$ , maintaining bitwise-even parity with any other label  $y$  in the tree. That is, if the  $q^{\text{th}}$  bit of  $y$  was clear, then make the  $q^{\text{th}}$  bit of  $y$  equal to the  $q^{\text{th}}$  bit of  $x$ ; otherwise, the  $q^{\text{th}}$  bit of  $y$  was set, so make the  $q^{\text{th}}$  bit of  $y$  equal to the complement of the  $q^{\text{th}}$  bit of  $x$ . Each vertex  $x$  appears in the same number of trees, so, over all  $2^d$  vertices  $x$  in  $K_2^d$ , the sum of the cardinalities of the sets  $\Phi_x(\mathcal{T}_{d+1})_{\mathbb{H}}$  of  $d$ -dimensional trees that contain  $x$  equals  $2^d |\Phi_0(\mathcal{T}_{d+1})_{\mathbb{H}}| = 2^d |\Phi_x(\mathcal{T}_{d+1})_{\mathbb{H}}|$ . But this counts each tree once for every vertex it contains; *i.e.*, it counts each Hamming-labeled tree  $d+1$  times. Letting  $|\Phi(\mathcal{T}_{d+1})_{\mathbb{H}}| = \tau(d)$  and  $|\Phi_x(\mathcal{T}_{d+1})_{\mathbb{H}}| = \tau_x(d)$ , it follows that

$$(d+1) \tau(d) = 2^d \tau_x(d) \tag{28}$$

Our shorthand  $\tau(d)$  reads: "The number of  $d$ -dimensional Hamming-labeled trees". In the paragraph preceding (28),  $x$  is a dummy, or formal, variable; thus,  $\tau_x(d)$  reads: "The number of  $d$ -dimensional Hamming-labeled trees, such that a fixed but arbitrary label appears somewhere in each tree". In a fashion reminiscent of the Catalan recurrence ([Comtet 1974] p. 53), we pinpoint  $\tau(d)$  and  $\tau_x(d)$  by establishing:

$$\tau(d) = 2^{d-1} \sum_{0 \leq q < d} \binom{d-1}{q} \tau_x(q) \tau_x(d-1-q) \tag{29}$$

Subject to the initial conditions  $\tau(1) = 1 = \tau_x(1)$  (30)

By Lemma 2,  $d$  bits are necessary and sufficient to Hamming-label  $T_{d+1}$ , wherein there is a unique edge  $(y, z)$  such that labels  $y$  and  $z$  differ only in the  $(d-1)^{\text{st}}$  bit. By the result of Table 4.d,  $(y, z)$  separates  $T_{d+1}$  into two Hamming-labeled sub-trees. Within either sub-tree, the  $(d-1)^{\text{st}}$  bit value remains unchanged. For  $0 \leq q < d$ , the dimensions of these sub-trees are, say,  $q$  resp.  $d-1-q$ , corresponding to a  $q \times (d-1-q)$  partition of the low-order  $d-1$  bits. Within a sub-tree, only the bits in its partition change, and each bit toggles only once as we traverse some edge. The set of possible sub-trees on each side of edge  $(y, z)$  is determined by the bit partition. For given  $(y, z)$  and bit partition, the number of Hamming-labeled trees equals  $\tau_x(q)$  times  $\tau_x(d-1-q)$ , whence the rightmost two factors in (29).

The number of  $q \times (d-1-q)$  partitions of the low-order  $d-1$  bits of  $y$  (or  $z$ ) is just the binomial coefficient in (29). By the result of Table 4.b, summing over all edges  $(y, z)$  is equivalent to summing over all vertices in the  $(d-1)$ -dimensional binary cube  $K_2^{d-1}$ , hence the coefficient  $2^{d-1}$  on the righthand side of (29). To put a fine point on this, note that including clear-set *and* set-clear ordered pairs for the leftmost bit of  $y$  and  $z$  double-counts the possibilities. Therefore, only the low-order  $d-1$  bits contribute to  $\tau(d)$ . All such summands are the same, hence the coefficient  $2^{d-1}$  on the righthand side of (29).

Having established (29), let us prove that its solution is

$$\tau_x(d) = (d+1)^{d-1}; \quad \tau(d) = 2^d (d+1)^{d-2} \tag{31}$$

Verify that (31) satisfies (30) at  $d=1$ . Applying (28) and the lefthand side of (31) to (29), it suffices to show that

$$2(d+1)^{d-2} = \sum_{0 \leq q < d} \binom{d-1}{q} (q+1)^{q-1} (d-q)^{d-2-q} \tag{32}$$

Start with Abel's generalization of the binomial identity:

$$(u+v)^m = \sum_{0 \leq q \leq m} \binom{m}{q} u (u-qw)^{q-1} (v+qw)^{m-q} \tag{33}$$

... for integer  $m \geq 0$  and real  $u, v, w$  ([Comtet 1974] p. 128). Substitute  $m = d-2, w = -1; u = 1, v = d$ , and multiply by 2:

$$2(d+1)^{d-2} = 2 \sum_{0 \leq q \leq d-2} \binom{d-2}{q} (q+1)^{q-1} (d-q)^{d-2-q} \tag{34}$$

Reminiscent of a stock-in-trade method for summing an arithmetic sequence (questionably ascribed to Gauss [Hayes 2006]), expand (34), reverse one of the series, and pair up terms with identical exponents and bases (*cf.* exercise, Table 8.d). We can therefore re-express  $2^d (d+1)^{d-2}$  as

$$\sum_{1 \leq q \leq d-2} \left[ \binom{d-2}{q} + \binom{d-2}{q-1} \right] (q+1)^{q-1} (d-q)^{d-2-q} + 2 \binom{d-2}{0} (0+1)^{0-1} (d)^{d-2} \tag{35}$$

Invoke Pascal's recurrence for the binomial coefficient ([Comtet 1974] p. 10), and expand the last term in (35):

$$2(d+1)^{d-2} = \binom{d-1}{0} (0+1)^{0-1} (d)^{d-2} + \sum_{1 \leq q \leq d-2} \binom{d-1}{q} (q+1)^{q-1} (d-q)^{d-2-q} + \binom{d-1}{d-1} (d)^{d-2} (1)^{-1} \tag{36}$$

Straightforward manipulation reduces (36) to (32); thus, (31) solves (29). Using the relation  $d = n-1$  of Lemma 2, (26) follows by re-writing (31) in terms of the order  $n$ . □

Comparing (25) with (31) unveils a pleasant bonus:

**Corollary 1.** The number  $\tau_x$  of trees of given order  $n$ , Hamming-labeled on the minimum number of bits, and with fixed but arbitrary label, equals the Cayley-Prüfer number  $n^{n-2}$  of free-labeled trees (25).

To illustrate Theorem 8 and Corollary 1: a) There are  $2^{6 \cdot 4} = 153664$  Hamming-labeled trees of order 7; Figures 7 and 8 depict two of these. b) The number of three-vertex Hamming-labeled trees equals  $2^2 \cdot 3^0 = 4$ ; Figure 9 exhibits all of these trees, i.e., the set  $\Phi(\mathcal{T}_3)_H$ . c) Exactly  $4^2 = 16$  Hamming-labeled trees on four vertices contain  $\mathbf{0} = 000$ ; Figure 10 enumerates the corresponding set  $\Phi_0(\mathcal{T}_4)_H$ .

In terms of Table 4.b,  $|\Phi(\mathcal{T}_{d+1})_H| = \tau(d)$  is the number  $2^d (d+1)^{d-2}$  of trees of maximum order  $(d+1)$  induced from  $K_2^d$ , and  $|\Phi_x(\mathcal{T}_{d+1})_H| = \tau_x(d)$  is the number  $(d+1)^{d-1}$  of such trees containing a fixed but arbitrary label.

### 3. WHIRLWIND TOUR: APPLICATIONS, RELATED WORK, ALGORITHMS, AND OPEN PROBLEMS

#### 3.1 Network Anatomy and Envelopes of Eccentricity

Tight metrizable graphs stand to benefit information communication. For example, configuring packet-switched network anatomies to match the adjacency of K-cubes achieves optimality in a multivariate sense of fault tolerance, throughput, latency, and power. More precisely: a) The degree  $d(j-1)$  of  $K_j^d$  equals its vertex connectivity (Table 4.c); thus,  $K_j^d$  minimizes size; i.e.,  $K_j^d$  gives equality in

$$|E(G): G \text{ is } (f+1)\text{-connected}| \geq \lceil \frac{1}{2}(f+1)n \rceil \quad (37)$$

... for given number  $f = d(j-1) - 1$  of faults tolerated, in the worst-case sense of [Harary 1962] and [Hayes 1974]. b) Under a model of uniform point-to-point channel capacity and power,  $K_j^d$  therefore optimizes the cost of fault tolerance that is super-logarithmic, but sub-linear, in the order. c) In the absence of faults,  $K_j^d$  optimizes the source-sink throughput, in the max-flow, min-cut sense of Ford and Fulkerson ([Corman et al 1993] p. 493). d) Graceful degradation of latency ([LaForge et al 2003] Cor. 13.2): for  $d \in o(j)$ , the diameter of a component induced by deleting any number of vertices of  $K_j^d$ , up to the worst-case fault tolerance  $f$ , converges to a lower bound on the radius of any minimum-size graph  $G$  with order  $n$  and connectivity  $f+1$ :

$$\text{RAD}(G) \geq \rho^-_{\text{MOL}} \stackrel{\text{def}}{=} \left\lceil \log_f \left[ \frac{n(f-1) + 2 + \lceil n(f+1) \bmod 2 \rceil}{f+1 + \lceil n(f+1) \bmod 2 \rceil} \right] \right\rceil \quad (38)$$

Not all hypercubes enjoy these advantages, especially (d), as summarized in the paragraph preceding (38). For example, P-cubes (always Hamming) and C-cubes (sometimes Hamming) diverge from (38), as  $n = j^d \rightarrow \infty$ , regardless of how  $d$  scales with  $j$  ([LaForge et al 2003] Thm 15).

Pioneering investigations that set the stage for the preceding model and results can be traced through Chapter IV of [Bollobás 1978], and references cited therein. Erstwhile subtle differences in formulation are in fact notable, as are limits to the practical applicability of the accompanying theorems. From [Erdős and Rényi 1962] to [Sampels 1997], researchers often seek to minimize size, for given diameter and maximum vertex degree, but without explicit concern for fault tolerance or, equivalently, connectivity. Adopting a model a bit closer to that synopsisized in (a) though (c) of the paragraph preceding (38), [Murty and Vijayan 1964] strive to minimize size, subject to given initial diameter and the maximum diameter of components induced by deletion up to any  $f$  vertices. Alas, solutions of this era are essentially confined to instances where the initial diameter equals 2, the induced diameter is less than or equal to 4, or the fault tolerance  $f$  equals 1.

By contrast, the classical upper bound  $n_{\text{Moore}}(f, h)$  ascribed to E. F. Moore (39) constrains the order  $n$  of an  $(f+1)$ -regular graph, for given  $f > 2$  and radius  $h$ . To visualize this, adjoin a root vertex to the root of each of  $f+1$  trees, each  $f$ -ary and of height  $h-1$ . This yields a tree of height  $h$  whose interior vertices are  $(f+1)$ -regular, hence

$$n \leq n_{\text{Moore}} \stackrel{\text{def}}{=} 1 + (f+1) \sum_{0 \leq q < h} f^q = 1 + (f+1) \frac{f^h - 1}{f - 1} \quad (39)$$

... and where we have invoked the formula for summing a geometric series ([Comtet 1974] p. 26). Maximally populating each level yields a Moore-complete tree with maximum order  $n_{\text{Moore}}$ . [Ore 1962] observed that, from any vertex  $x$  in connected graph  $G$ , there is spanning tree of  $G$  that preserves the edge distance between  $x$  and every other vertex. Such a tree is alternatively deemed Ore, distance, or breadth-first ([Corman et al 1993] p. 469). If every such  $f$ -ary Ore tree of height  $h$  is Moore-complete then  $G$  is diametrically Moore ([Bermond and Bollobás 1981] p. 7): a)  $G$  achieves equality in (39). b)  $G$  has minimum radius  $h$ , over all  $(f+1)$ -regular graphs of order  $n_{\text{Moore}}$ . c)  $G$  achieves equality in the leftmost two relations of:

$$\begin{aligned} \text{RAD}(G) \leq \text{ECC}(x) \leq \text{DIAM}(G) \leq 2 \cdot \text{RAD}(G) \\ \text{([Chartrand and Lesniak 1986] Thm 2.4)} \end{aligned} \quad (40)$$

Thus, the tightest graphs for which we could hope are diametrically Moore, wherein  $\text{DIAM} = h$ , as given by (39).

To preempt potential confusion,  $n_{\text{Moore}}$  also bounds from below the order of an  $(f+1)$ -regular graph with odd girth  $g = 2h + 1$ . For even girth  $g = 2h$ , the lower bound reduces to  $2(n_{\text{Moore}} - 1) / (f+1)$ . Graphs of minimum order closest to these bounds are the  $(f+1, g)$  cages. Cages whose order exactly matches these bounds are girthed Moore graphs, for  $g = 2h + 1$ ; resp. generalized polygon graphs, for  $g = 2h$  ([Biggs 1996] p. 181; [Bollobás 1998] p. 106 uses Moore graph to subsume both odd and even girth).

While cages are fascinating, the practical applicability of girth falls short of what computer architects desire for fault-tolerant, high-throughput, low-latency, power-conserving multi-processor systems. Indeed, our own work on avionics for deep-space probes ([LaForge 1999], [LaForge 2000]) prompted us to adopt refinements as follows.

For a worst-case- $f$ -fault-tolerant network or computer bus, any  $f$  processors can fail, yet the remaining  $n - f$  healthy processors are guaranteed to remain connected. Under a point-to-point model of communication, the network or bus adjacency must therefore match that of an  $(f + 1)$ -connected graph. However,  $(f + 1)$ -regularity does not imply  $(f + 1)$ -connectivity (Table 4.c). In the interest of accurate modeling, we replace conditions of regularity and girth with connectivity and minimum size. Translated to count of discrete channels, the latter is consistent with practical goals for conserving power. As good luck would have it, these erstwhile more stringent conditions imply regularity or near-regularity:  $n - 1$  of the  $n$  vertices of an  $(f + 1)$ -connected minimum-size graph have degree  $f + 1$ , and one vertex has degree  $f + 1 + [(f + 1)n \bmod 2]$ . For all  $n > f + 1 > 1$ , [Harary 1962] and [Hayes 1974] give constructions for regular (or nearly regular) chordal graphs with the corresponding minimum size  $\lceil \frac{1}{2}(f + 1)n \rceil$ ; *i.e.*,  $(f + 1)$ -connected graphs that achieve equality in (37).

It appears that the edge distance in Harary-Hayes chordal graphs was not formulaically pinpointed until Theorem 1 of [LaForge et al 2006]. The upshot: even in the absence of faults, the bloated diameter is, alas, within one of that of a maximally eccentric  $(f + 1)$ -connected graph, where the latter is *without* the condition of minimum size. Figure 11 plots the formidable gap between this worst possible diameter  $\sim n / (f + 1)$  and the best possible radius  $\rho_{\text{MOL}}^-$ . All this reinforces our call for graphs that a) are  $(f + 1)$ -connected; b) have minimum size (37); c) are constructible for most (if not all)  $n$  and  $f$ ; d) unlike Harary-Hayes graphs, deliver eccentricity approaching  $\rho_{\text{MOL}}^-$ , even in components degraded by faults (*i.e.*, induced by deleted vertices).

With respect to (c), (d), and (39), note that, in the absence of faults, the root of at most one Ore tree has degree  $f + 2$ . In the infinitely many cases when both  $n$  and  $f + 1$  are odd, that is, a diametric Moore graph cannot exist. Otherwise,  $(f + 1)n$  is even, and any minimum-size  $(f + 1)$ -connected graph is  $(f + 1)$ -regular. For  $f \geq 2$ , therefore, any  $(f + 1)$ -connected diametric Moore graph also has odd girth  $2h + 1$ ; whence the diameter equals  $h$  as given in (39). But, apart from cycles and cliques, there either four or five Moore graphs of odd girth ([Biggs 1996] pp. 185 – 187). *E.g.*, the (3, 5) cage, a.k.a. the Petersen graph depicted in Figure 13, is one of them; the existence of a (3, 57) Moore graph remains open. In the interest of criterion (c), therefore, it behooves us to relax our quest for diametric Moore graphs.

To this end, note that adding an edge to any graph neither decreases the connectivity, nor does it increase the eccentricity of any vertex. By monotonicity, that is, we can shift our optimization spotlight from the graph order (39), and refocus on eccentricity as our primal objective:

What  $(f + 1)$ -connected minimum-size  $n$ -vertex graphs minimize the maximum a) radius or b) diameter of components induced by deleting up to  $f$  vertices? (41)

This gives rise to (38), deemed the "Moore-Ore-LaForge Bound" by colleagues at the Jet Propulsion Laboratory [LaForge 1999]. The calculation mod 2 accommodates odd values of  $(f + 1)n$ , and the ceiling function  $\lceil \cdot \rceil$  captures *incomplete* Moore trees. As a check at  $n = 10$  and  $f = 2$ , verify that, with diameter = 2, the Petersen graph (Figure 13) achieves equality in (38), without the ceiling.

Reminiscent of the formulation of [Murty and Vijayan 1964] mentioned earlier in this section, conditions (a) and (b) of (41) are considerably more severe than merely asking for the minimum eccentricity in the absence of faults. To balance this exigency, sequence a set of  $(f + 1)$ -connected minimum-size graphs whose order grows without bound. Such a set *degrades gracefully* if the diameter of any component induced by deleting up to  $f$  vertices is  $\Theta(\rho_{\text{MOL}}^-)$ . Of course, we also seek to minimize the  $\Theta$ -constant. While it is perhaps traditional to seek the minimum diameter (41)(b), life is often easier when dealing with the radius (41)(a). By (40), the diameter is at most twice the radius, a fact subsumed by our secondary objective of minimizing the  $\Theta$ -constant.

Which brings us to K-cubes whose radix  $j$  and dimension  $d$  are allowed to vary. Unlike Harary-Hayes chordal graphs,  $\{K_j^d\}$  can be tight in the absence of faults, remaining tight in the presence of faults. Figure 12 illustrates how this tight latency manifests even in the relatively small  $n = 9$  two-dimensional ternary K-cube  $K_3^2$ . In the large, and recalling the paragraph before (38),  $\{K_j^d\}$  degrades gracefully: as  $n = j^d \rightarrow \infty$ , and for  $d \in \mathcal{O}(j)$ , the ratio of  $\rho_{\text{MOL}}^-(K_j^d)$  to the diameter of any component, induced by deletion of up to  $f = d(j - 1) - 1$  vertices from  $K_j^d$ , converges to *one*.

[Bermond and Bollobás 1981] pose as open whether there exists a sequence of graphs with maximum degree  $f + 1$ , order  $\Theta(n_{\text{Moore}})$ , and diameter at most  $h$ , as given in (39). Since connectivity and eccentricity are monotone in size, the graceful degradation of  $\{K_j^d\}$  ([LaForge et al 2003] Cor. 13.2) answers their question in the affirmative.

This is by no means the final word on (41), however. For example, convergence of  $\{K_j^d\}$  to  $\rho_{\text{MOL}}^-$  is logarithmically slow. As Figure 11 elaborates, moreover,  $\{K_j^d\}$  is but a partial solution to (41), and falls short of our desire (c) for constructibility over a wide range of  $n$  and  $f$ . By contrast, we can synthesize a Harary-Hayes graph for *any*  $n > f + 1 > 1$ .

	Combinatorial Hamming Properties of Interesting Graphs			{Free Labelings}
	Minimum Hamming Dimension	Extent-minimum Hamming Radix	{Hamming Labelings}	
C-mesh $C_j$	$\lceil \frac{1}{2}(d\mathbf{j}_d - 3t) \rceil + t$ $d =$ Lee dimension, $t =   \{ \text{ternary extents in the Lee radix } \mathbf{j} \}  $ , iff $d - t$ extents all even Lemma 1, Remark 1	3 for $t$ high-order extents in the Lee radix $\mathbf{j}$ else 2 for $\lceil \frac{1}{2}(d\mathbf{j}_d - 3t) \rceil$ low-order extents Lemma 1, Remark 1	1 if $b + f = d - t$ $b =   \{ \text{binary extents} \}  $ $f =   \{ \text{4-ary extents} \}  $ else $2^{b+t+f-d} \lceil \frac{1}{2}(d\mathbf{j}_d - 3t) \rceil !$ iff all extents $> 4$ are even Theorem 2	$2^{\lceil b/2 \rceil - d} (\mathbf{j}^d - 1)!$ $b =   \{ \text{binary extents in the Lee radix } \mathbf{j} \}  $  Theorem 3
C-cube $C_3^d$	Ternary hypercube same as K-cube $K_3^d$ . Lee dimension $d =$ Hamming dimension	3  Remark 1	1  Theorem 2	$2^{-d} (3^d - 1)!$  Theorem 3
C-cube $C_2^d$	Classic binary hypercube same as K-cube $K_2^d$ . Lee dimension $d =$ Hamming dimension	2  Remark 1	1  Theorem 2	$2^{-d} (2^d - 1)!$  Theorem 3
Cycle $C_n$	1 for $n = 3$ undefined for odd $n \geq 5$ $n/2$ for $n = 2d \geq 4$ $n =$ order of $C_n$	3 for $n = 3$ undefined for odd $n \geq 5$ 2 for $n = 2d \geq 4$ Table 4.e, f, k	1 for $n = 3$ 0 for odd $n \geq 5$ $\frac{1}{2}(\lceil n/2 \rceil !)$ for $n = 2d \geq 4$ Theorem 1	1 for $n = 3$ $\frac{1}{2}(n - 1)!$ for $n \geq 3$  Eq (4)
P-mesh $P_j$	$d(\mathbf{j}_d - 1)$ $d(j - 1)$ for P-cube $P_j^d$ $d =$ Manhattan dimension Theorem 4	2  Theorem 4	$2^{d(\mathbf{j}_d - 2)} [d(\mathbf{j}_d - 1)]!$  $2^{d(j - 2)} [d(j - 1)]!$ for P-cube Theorem 4	$(\frac{1}{2})^{\lceil b/2 \rceil + d} (\mathbf{j}^d)!$ $b =   \{ \text{binary extents in the Manhattan radix } \mathbf{j} \}  $ Theorem 5
Path $P_n$	$n - 1$ $n =$ order of $P_n$ Theorem 4	2  Theorem 4	$2^{(n - 2)} (n - 1)!$  Theorem 4	$(\frac{1}{2})n!$  Theorem 5
K-mesh $K_j$	$d$ (same)	$j$ (same)	1 (Mesh or cube is complete) For free labelings at right: $b =   \{ \text{binary extents in the Hamming radix } \mathbf{j} \}  $	$\left(\frac{1}{2}\right)^{\lceil b/2 \rceil} \frac{(\mathbf{j}^d)!}{\prod_{0 \leq q < d} (j_q)!}$  Theorem 6
{Trees} $T_n$	$n - 1$ $n =$ order of $T_n$ (New, streamlined proof) Lemma 2	2  (New, streamlined proof) Lemma 2	$2^{n - 1} n^{n - 3}$  (A Cayley-Prüfer type formula) Theorem 8	$n^{n - 2}$  (Cayley; our algorithmic proof implements the coding scheme of Prüfer) Theorem 7
$T_n$	As above, but with the condition that a fixed but arbitrary Hamming label, of minimum dimension $n - 1$ and extent-minimum radix 2, appears in each Hamming-labeled tree.		$n^{n - 2}$  (Another Cayley-Prüfer type formula) Corollary 1	

Table 6: Chief, specific, original contributions of this paper;  $j, \mathbf{j}, j^d, \mathbf{j}^d$  as in Table 1;  $\mathbf{j}_d$  defined in Table 4.c.

a.	Beyond stock-in-trade binary cubes [Armstrong and Gray 1981], or even $j$ -ary-cubes with $j > 2$ ([Brouwer et al 1989], [Bose et al 1995], [LaForge et al 2003]) our framework on a <i>mixed</i> mesh radix $\mathbf{j}$ facilitates fresh insights; <i>e.g.</i> , a Hamming graph is radix-invariantly induced from its mesh ideal.	Sections 1.1, 1.2
b.	General constructions for meshes and cubes metrized on the Lee, Manhattan, and Hamming distance, based on cycles, paths, resp. cliques. Enumeration of Hamming versus free labelings in each case.	Sections 1.2, 2
c.	Qualitative and quantitative description of the hyperfactorized anatomy of Hamming graphs.	Sec 1.3
d.	A C-mesh, P-mesh, or K-mesh has more free labelings than it has Hamming labelings. For given order, the set of trees has more Hamming labelings than free labelings.	Sec. 2
e.	Explicit formula for the number of vertices with degree $d + q$ in the irregular P-mesh $P_j$ : $2^{d-q} \sum_{j[q,b] \in \mathcal{J}[q,b]} \prod_{i \in j[q,b]} (i-2) \quad 0 \leq q \leq d-b, \text{ summation equals one if } q=0 \quad (14)$ $d = \text{Manhattan dimension}; b =  \{\text{binary extents in the Manhattan radix } \mathbf{j}\} ; \mathcal{J}[q,b]$ is the set of all subsets $j[q,b]$ of $q$ extents drawn from the subvector $(j_{d-1}, \dots, j_b)$ on the $d-b$ high-order extents	page 10
f.	Recurrence relation (15) and explicit solution (16) for the size of the irregular P-mesh $P_j$ and P-cube $P_j^d$ . $ E(P_j)  = \sum_{0 \leq q < d} (j_q - 1) \prod_{0 \leq i \neq q < d} j_i \quad \text{For P-cube, reduces to }  E(P_j^d)  = d(j-1)j^{d-1}$	page 10
g.	Recap of Hamming graph facts.	Table 4
h.	Sampling of pertinent applications, related work, key algorithms, and ( <i>cf.</i> Table 8) open problems.	Sec. 3

Table 7: General and ancillary contributions of this paper.

a.	Generalize Table 6 to the number of Hamming-labeled graphs, in cases other than C-meshes, P-meshes, K-meshes, and trees; give sharp results in terms of parameters such as the order $n$ , dimension $d$ , and radix $\mathbf{j}$ . For example, how many binary Hamming graphs are there of order $n$ ? Of dimension $d$ ?	page 1 page 12
b.	Extend the theory of metrization begun for Hamming graphs, and as outlined in Sections 1.2 and 1.3. <i>E.g.</i> , are C-meshes and C-cubes the radix-invariant ideals of Lee graphs? (We conjecture that they are.) Are P-meshes and P-cubes the radix-invariant ideals of Manhattan graphs? (We suspect that they are.) Crystallize and prove results analogous to those of Table 4.b and d. Generalize to other distances.	Sections 1.2, 1.3
c.	Quantify the order-of-magnitude running times of Algorithms $\mathcal{A}_{\text{Prüfer}}$ and $\mathcal{A}_{\text{Prüfer}}^{-1}$ . Prove that these are best possible or, if not, then give optimum alternative(s).	proof of Thm 7
d.	Our proof of Theorem 8 hinges on Abel's generalization (33) of the binomial identity. We used it to verify, by substitution, that our exhibited solution (31) satisfies the Catalan-type recurrence (29) for the number $\tau$ of Hamming-labeled trees. Solve (29) by a different, illuminating method, such as generating functions.	proof of Thm 8
e.	K-cubes $\{K_j^d\}$ are known to multivariately optimize throughput, fault tolerance, latency, and power. Expand this class to, say, K-meshes, other Hamming graphs, or graphs metrizable on other distances.	Sec. 3.1 Eq (38)
f.	Extend the results of Table 4.b from 1-Hamming graphs to Hamming graphs with arbitrary weight vector $\boldsymbol{\mu}$ .	Sec. 3.2
g.	Prove that the to-date fastest Algorithm $\mathcal{A}_{\text{Label-Hamming}}$ for labeling an unweighted Hamming graph is run-time optimal, or replace it with an optimal one. Generalize to the weighted case of $\boldsymbol{\mu}$ -Hamming graphs.	Sec. 3.3
h.	Bridge the $\Theta( \boldsymbol{\lambda} )$ multiplicative gap between a lower bound $\Omega(d \cdot  \boldsymbol{\lambda} )$ on the running time to decide whether $\boldsymbol{\lambda}$ is Hamming graphic, and that $O(d \cdot  \boldsymbol{\lambda} ^2)$ for Algorithm $\mathcal{A}_{\text{Construct-Hamming}}$ of [LaForge 2004].	page 4 Sec. 3.3
i.	Derive expressions, in terms of parameters such as the order $n$ , dimension $d$ , and radix $\mathbf{j}$ , for the number $ \Phi^{-1}(\mathbf{G})_{\mathbf{H}} $ of Hamming graphic inverse metrizations; <i>i.e.</i> , $ \{\boldsymbol{\lambda} \subseteq L_j \text{ whose Hamming graph is connected}\} $ .	page 4 Sec. 3.3
j.	What (or how many) maximum weights are necessary to metrize all $\boldsymbol{\mu}$ -Hamming graphs of order $n$ ? <i>E.g.</i> , instances of $\boldsymbol{\mu}$ -Hamming graphs seemed confined to $\boldsymbol{\mu}$ 's with weights 1 or $1/2$ ; do we need more than this?	Sec. 3.3

Table 8: Research avenues, open problems, and exercises.

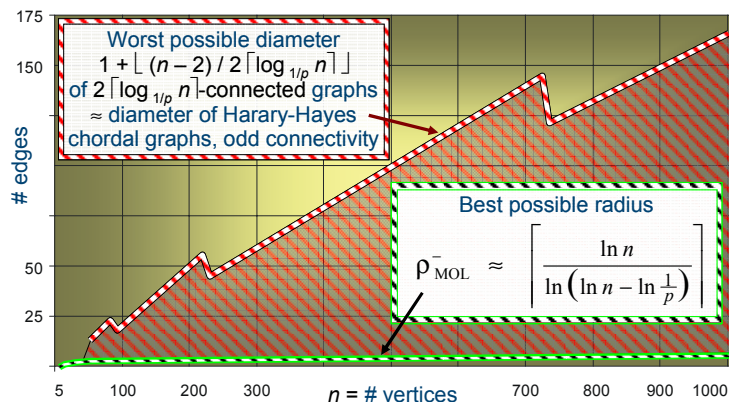


Figure 11: Envelope of eccentricity, best to worst. For even  $n$  connectivity, the Harary-Hayes diameter remains bloated at half the value of the topmost plot, and exhibits a similarly jagged trajectory. Here we have scaled the vertex degree at about the minimum  $2 \lceil \log_{1/p} n \rceil$  that assures  $p = 10\%$  probabilistic fault tolerance. With probability  $1 - o(1)$ , that is, Bernoulli deletion of up to  $p = 10\%$  of all  $n$  vertices induces a single component containing all of the remaining 90%. (This is somewhat different from the model of random graphs pioneered by Erdős and Rényi [LaForge et al 2006] Thm 3). The connectivity and degree of  $K$ -cubes is super-logarithmic in the order, hence  $\{K_j^d\}$  cannot converge to  $\rho_{MOL}^-$  at minimum size  $n \lceil \log_{1/p} n \rceil$ .

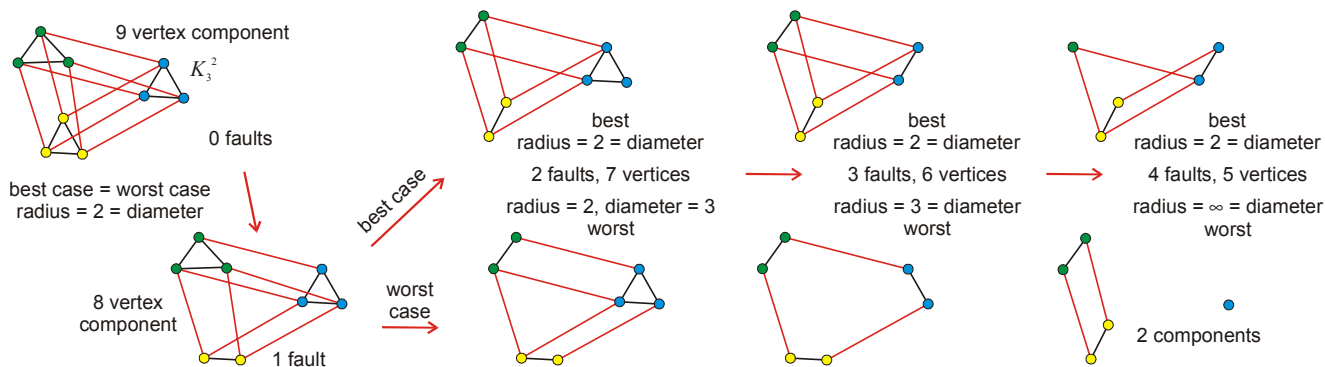


Figure 12: Envelope of degradable eccentricity. A two-dimensional ternary  $K$ -cube stays tight up to the fault tolerance  $f = 3$ . The diameter is the best possible  $\rho_{MOL}^-(n = 9, f = 3) = 2$  for zero or one fault; the radius equals  $\rho_{MOL}^-$  for up to 2 faults.

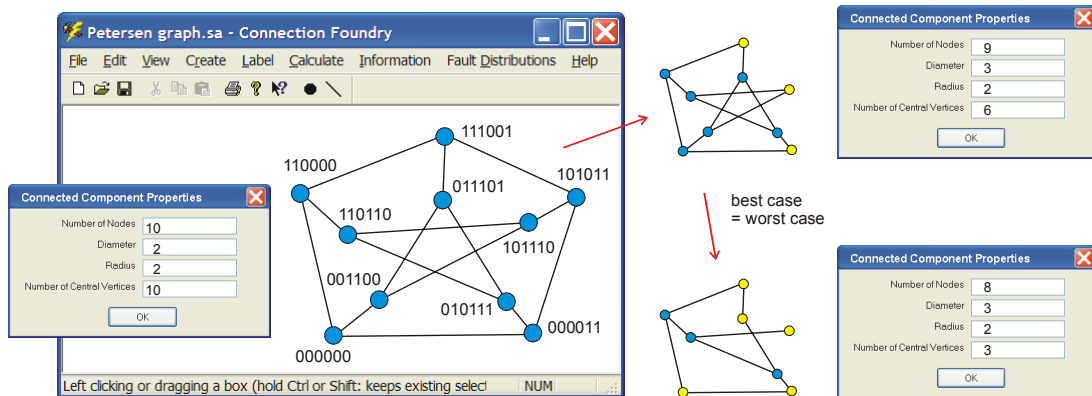


Figure 13: Envelope of degradable eccentricity for the Petersen graph. It is one of four known Moore graphs, hence tighter than any other 3-connected graph of order 10 and minimum size 15. Up to the fault tolerance  $f = 2$ , moreover, it stays tight. *I.e.*, deleting one or two vertices induces a component whose radius 2 is unchanged from the original, and whose diameter 3 is just one greater than the (optimum) original  $\rho_{MOL}^- = 2$ , as given by (38). While not Hamming in the unweighted sense considered throughout most of this paper, the Petersen graph *is* Hamming with respect to uniform weight  $1/2$ : the Hamming distance between labels as shown equals twice the edge distance between the corresponding vertices (*cf.* Section 3.2).

In the interest of expanding our menu of solutions to (41), what other Hamming graphs, apart from  $\{K_j^d\}$ , enable high throughput and fault tolerance, and low latency and power? Are there other *distances* whose metrized graphs a) are  $(f+1)$ -connected; b) have minimum size (37); c) are constructible for most (if not all)  $n$  and  $f$ ; d) deliver eccentricity approaching  $\rho_{\text{MOL}}^-$ , even in components degraded by faults? (Cf. research avenue, Table 8.e) The divergence of P-cubes and C-cubes from  $\rho_{\text{MOL}}^-$  does not bode well for the Manhattan or Lee metrics, and so prompts investigations into alternatives, such as the *weighted* Hamming distance of [Avis 1978], discussed in Section 3.2.

On an encouraging note, computer architects may at last be weaning themselves off of the venerable all-pairs crossbar, and instead building on results akin to (a) through (d) above. At SiCortex, for example, designers are leveraging such results to realize massively-parallel multi-computers [Metcalf 2007]. Splitting a bidirectional channel into two one-way channels, a minimum-size undirected  $(f+1)$ -connected graph is consistent with a strongly  $(f+1)$ -connected *digraph* on fewest arcs, such that each vertex has in-degree =  $f+1$  = out-degree. This changes the tree underlying (39) to one whose interior vertices all have out-degree  $f+1$ . The directed version of (38) thus emerges as:

$$\text{DIAM} \geq \text{RAD} \geq \rho_{\text{MOK}}^- \stackrel{\text{def}}{=} \lceil \log_{f+1}(1 + nf) \rceil - 1 \quad (42)$$

Our subscript  $\kappa$  acknowledges [Kautz 1968], whose constructions achieve equality in the *Moore-Ore-Kautz Bound* (42). SiCortex's SC5832 comprises 972 network nodes, each with six processor cores; the underlying (3, 3)-regular Kautz digraph has best possible diameter  $\lceil \log_3(1945) \rceil - 1 = 6$  [SiCortex Fabric 2006]. Interviewed for this paper, SiCortex's chief technology officer credited the Internet search engine Google as key to his (re)discovery of the Kautz digraphs [Leonard Phone Conv 2008].

Along with  $\rho_{\text{MOL}}^-$  (38) and  $\rho_{\text{MOK}}^-$  (42), the best possible diameter of a minimum-size (strongly)  $(f+1)$ -connected (di)graph grows logarithmically in the order  $n$ . However, and as Figure 11 illustrates, connectivity (hence degree, hence the logarithmic base) can scale as a function of  $n$ . To bound the diameter at  $q$ , for example, the degree is  $\Theta(n^{1/q})$ . While for computers that we wire together it may well be reasonable to hold the degree fixed, it is compellingly attractive to *tune* the degree when wireless communications make it relatively easy to (re)configure adjacency. To this end, the recently-invented *steerable* vertical cavity surface-emitting laser of [Choquette et al 2006] bodes promise for self-tuning multi-computers interconnected by free-space optics [LaForge et al 2006 VCSELS]. In the context of grid computing, [LaForge et al 2006] expound on tunable connectivity, with emphasis on mobile *ad hoc* networks (MANETs) interconnected by radio communications.

### 3.2 Network Physiology and Cross-Layer Optimization

Notwithstanding the attractive properties of K-cubes, is metrization *essential* to anatomical optima (41), such as graceful degradation, or maximization of the ratio of connectivity to the average degree? Not really. As with the K-cube of Figure 12, for example, graphs induced from the Petersen graph (cf. Figure 13) are attractively tight. However, the Petersen graph is not Hamming, at least in the unweighted sense used so far in this paper (Table 4.q). More generally, and in light of scalable constructions such as those of [Kautz 1968], should computer architects find metrized graphs *desirable*? If so, then why?

Our answer: yes, computer architects should indeed prefer metrized graphs, and for reasons of network *physiology*. For example, suppose we connect computers via communication links, in correspondence with the edges of some Hamming graph. Further suppose that each node stores its own Hamming label, along with the labels of its neighbors in the underlying graph. Then the Hamming labeling serves as a *distributed table* that drives geodesic routing Algorithm  $A_{\text{H-Route}}$ : forward a packet to a neighbor whose label minimizes the Hamming distance to the destination.

[LaForge 2004] introduced a pseudocoded version of Algorithm  $A_{\text{H-Route}}$ , since implemented in the Connection Foundry software shown in Figures 4, 8, and 13. If a graph is Hamming-labeled with fewest digits  $d$  then, by Table 4.b,  $d$  is the dimension of its Hamming ideal. The diameter is at most  $d$ , and no geodesic entails more than  $d+1$  routings of a single packet. Under a model where it takes constant time to compute the Hamming distance between two labels, Algorithm  $A_{\text{H-Route}}$  takes time  $O(d \cdot \mathbf{j}_d)$  to find the minimum Hamming distance between the destination label and some neighbor. Thus, the time to route any packet is  $O(d^2 \cdot \mathbf{j}_d)$ . If computing the Hamming distance is deemed to take constant time per digit, then multiply this cost (as well as any lower bound) by  $\Theta(d)$ . To account for bit-level representation of non-binary digits, multiply the cost by  $\Theta(d \lg \mathbf{j})$ , where the latter invokes the shorthand notation of Table 4.c, and where we have replaced  $j_q$  with the binary logarithm  $\lg$  of  $j_q$ .

In the language of computer architects, Algorithm  $A_{\text{H-Route}}$  implements a *distance-vector protocol*. Its per-node space  $O(d^2 j \lg j)$  is considerably more economical than that of legacy protocols of the *path-state* variety. In effect, the latter tabulate the cost of every path to every destination, at every routing node (cf. matrix  $H$  in Sec II.A of [Wang et al 2005]). However, the combinatorially explosive nature of path-state routing impedes the scalability of MANETs ([LaForge et al 2006] Sec. 2.4). Even in the sparse, 4-connected  $C_j^2$ , for example, the number of path states from any node to any other is at least  $j^2(2^{j+2} - 8)$ ; i.e., super-exponential in the square root of the total number of nodes in the network.

If the graph of a network is not Hamming then we shouldn't try to use Algorithm  $\mathcal{A}_{\text{H-Route}}$ . However, if it is metrized on distance  $|\cdot, \cdot|_{\Delta}$  then we can apply general geodesic Algorithm  $\mathcal{A}_{\Delta\text{-Route}}$ : forward a packet to a neighbor whose label minimizes the  $|\cdot, \cdot|_{\Delta}$  distance to the destination. This, in turn, underscores the practical importance of expanding our *menu* of distances, along with operational results (e.g., Table 4, Figure 4, Theorem 8) about graphs so-metrized.

For example, while the geodesic routing algorithm presented in [Avis 1978] is restricted to binary labels, and without explicit appeal to dimension or extent minimality, it reaches beyond this paper by considering *weighted* Hamming graphs. Set the  $q^{\text{th}}$  bit of the *Hamming indicator* of labels  $x$  and  $y$  (of identical length) if and only if  $x$  and  $y$  differ in the  $q^{\text{th}}$  digit. The  $\mu$ -Hamming distance  $|x, y|_{\mu\text{-H}}$  is the inner product of this Hamming indicator with vector  $\mu$ , comprising positive reciprocal-integer weights. Graph  $G$  is  $\mu$ -Hamming-labeled if the edge distance between every two vertices equals the  $\mu$ -Hamming distance between the respective labels.  $G$  is  $\mu$ -Hamming if it can be  $\mu$ -Hamming-labeled;  $G$  is *Hamming* if it is  $\mu$ -Hamming for some  $\mu$ .

Although this paper focuses on the unweighted case  $\mu = \mathbf{1}$ , there are most certainly graphs that are Hamming only when  $\mu \neq \mathbf{1}$ . E.g., while odd cycles  $C_{2q+1}$  (other than triangles) are not  $\mathbf{1}$ -Hamming (Table 4.k), [Avis 1978] shows how to Hamming-label them with weight  $\frac{1}{2}$  on each of  $2q + 1$  bits. However, we are left wondering whether  $2q + 1$  is the *minimum* Hamming dimension. Or, for that matter, whether Figure 13's six-bit  $\frac{1}{2}$ -Hamming-labeling of the Petersen graph is the shortest. What *is* clear is that the minimum Hamming dimension for  $\mu = \mathbf{1}$  is minimum over all weights. In this sense the nonzero results of Table 6 are best possible.

For illustration: a) Theorem 2.3 of [Avis 1978] uses  $q + 1$  bits to Hamming-label any even cycle  $C_{2q}$ , such that two of the weights equal  $\frac{1}{2}$ , and the remaining  $q - 1$  weights equal one. However,  $C_{2q}$  has minimum dimension  $q$  and extent-minimum radix two (Table 4.l, Theorem 1). Therefore,  $q$  is the minimum Hamming dimension of  $C_{2q}$ , over all weights, and [Avis 1978] exceeds this minimum by just one. b) [Avis 1978] uses  $n$  bits and uniform weight  $\frac{1}{2}$  to Hamming-label any  $n$ -vertex clique  $K_n$ . However,  $K_n$  is the one-dimensional unweighted  $n$ -ary Hamming ideal, hence has minimum dimension 1 and extent-minimum radix  $n$  (cf. Section 1.2). Even after translating the digits of  $L_n$  to binary, the  $\lceil \lg n \rceil$  bits to represent a minimum-dimension unweighted Hamming-labeling of  $K_n$  compare favorably to the  $n$  bits it takes to  $\frac{1}{2}$ -Hamming-label  $K_n$ . By focusing on the unweighted case, our lexicographically parsimonious approach – starting with mixed mesh radices, and emphasizing dimension and extent minimality – yields fresh contributions to what we know about Hamming graphs. (Cf. research avenue, Table 8.f.)

We therefore suggest that it would be profitable to thoroughly survey and integrate legacy developments about Hamming graphs (including weighted Hamming graphs) with those of Table 4, Table 6, and Table 7. Beyond our nascent sampling here, such a compilation might well commence with references cited in [Avis 1978] (or the less complete but more accessible [Avis 1981]). Many results therein have been independently rediscovered, and will quite possibly be rediscovered again, often with new insights. For example, [Kelly 1975] establishes that trees, cliques, and cycles are Hamming. [Avis 1978] gives independent proofs of these facts, in the context of rays defined by distances between points in the Euclidean plane. We learned of these works after proving essentially the same things, more-or-less directly, from first combinatorial principles [LaForge 2004].

As a second example of how rediscovery can lead to new insights, our theorem that the hyperfactorized primality of a Hamming graph equals its minimum dimension (Table 4.d) complements [Graham and Pollack 1971]'s algebraic bound: the dimension is at least the number of eigenvalues, with given sign, of the underlying distance matrix.

As a third example of rediscovery, the generalized geodesic routing Algorithm  $\mathcal{A}_{\Delta\text{-Route}}$  set forth in this section amounts to a contemporary twist on solutions to the *Loop-switching Problem* of [Graham and Pollack 1971], [Brandenburg et al 1972], and [Pierce 1972]. Algorithm  $\mathcal{A}_{\text{H-Route}}$ , introduced in [LaForge 2004] and analyzed in this section, is effectively that espoused nearly three decades prior by [Avis 1978].

*Cross-layer optimization* of wireless MANETs reinforces the benefit of (re)discoveries, such as those highlighted in this section and in Section 3.1. Imagine mobile sensor nodes whose mission calls for cooperation, hence robust communication. To this end, each node possesses a software defined radio (SDR) into which we embed algorithms for multivariately optimizing fault tolerance, throughput, latency, and power ([LaForge 2004 MANETs], [LaForge et al 2006], [Wang et al 2005]). SDRs *tune* the number of channels (i.e., the vertex degree in the underlying graph) to match mission objectives and constraints. This is similar to self-tuning packaged multi-computers, interconnected with free-space optics, and described at the end of Section 3.1.

Notwithstanding important details, such as radio interference, MANETs can (and should) tune their *anatomy* by self-configuring as graphs which degrade gracefully (41), or which maximize the ratio of connectivity to vertex degree. To enhance MANET *physiology*, SDRs embedded in nodes can assign metrized labelings that enable efficient distance-vector packet routing. For *cross-layer optimization*, we seek both: i.e., families of graphs, like  $\{K_j^d\}$ , which are anatomically optimum, and which are metrized to maximize the physiological routing benefit of the underlying anatomy.

Which brings us back to trees. Spanning trees, in particular, provide us with a fall-back when communication anatomy lacks network-wide metrization. We can compensate with a separate table of Hamming labels at each of  $n$  sensor nodes, one for each source  $x$ . Embedded Algorithm  $\mathcal{A}_{\text{Tree-Route}}$  forwards a packet to a neighbor that minimizes the remaining Hamming distance from  $x$  to the destination, within an Ore tree rooted at  $x$ , and Hamming-labeled on the fewest bits  $n-1$ . Assuming that each bit exacts constant cost, the  $O(n^2)$  per-node space of such tree-based routing remains favorable when compared to path-state alternatives.

We can modify Algorithm  $\mathcal{A}_{\text{Tree-Route}}$  to economize on electricity. Let  $w(q, r)$  be the least power to maintain a bidirectional channel between nodes  $q$  and  $r$ , subject to minimum average bit delay. A spanning tree which minimizes the sum of such  $w$ 's minimizes the overall power  $\times$  delay of any graph that connects all  $n$  sensors ([Corman et al 1993] Chap. 24). Our MANET may have channels other than those corresponding to the edges of this minimum power  $\times$  delay tree, but our overarching preference is to route traffic within the tree. Since the only labels we need to store at any node are those of its neighbors in this tree, we shave a  $\Theta(n)$  factor from the space cost of the Ore-tree version of Algorithm  $\mathcal{A}_{\text{Tree-Route}}$ , described above.

In light of variants on geodesic routing Algorithm  $\mathcal{A}_{\text{Tree-Route}}$ , cross-layer optimization is not likely to be severely hindered by a lack of network-wide metrization. To mitigate such physiological deficiencies in practice, we can Hamming-route packets within spanning trees. And, as we now know (26), the number of trees of order  $n$ , Hamming-labeled with minimum dimension  $n-1$ , equals  $2^{n-1}n^{n-3}$ .

### 3.3 From Hamming Algorithms to Combinatorics

Application-oriented algorithms such as  $\mathcal{A}_{\text{H-Route}}$  owe their lineage to the seminal error detection and correction scheme of [Hamming 1950], now a textbook standard ([MacWilliams and Sloane 1998] Sec. 1.7; [Wakerly 1994] Sec. 2.15.3). From a theoretical standpoint, we are interested in algorithms that are *about* Hamming labelings.

For example, [Avis 1978] casts the problem of deciding whether an  $n$ -vertex graph is Hamming as a  $\frac{1}{2}(n^2-n) \times (2^{n-1}-1)$  linear program. Alas, even dynamic column-generation appears to fall short of guaranteeing pruned inputs whose size is sub-exponential in  $n$ . Other Hamming-labeling algorithms can be traced through [Aurenhammer and Hagauer 1995], as well as references cited therein. If computing the Hamming distance takes constant time per (unweighted) digit, and  $d$  is the minimum dimension, then the  $O(d \cdot \max[|E|, n(j-1)])$  running time of  $\mathcal{A}_{\text{Label-Hamming}}$ , illustrated by Figures 4 and 8, is optimal whenever the input is Hamming,  $|E| \in \Theta(n)$ , and the cube radix  $j$  is  $\Theta(1)$ .

Polynomially effective for mixed (albeit unweighted) radices,  $\mathcal{A}_{\text{Label-Hamming}}$  is evidently the fastest Hamming-labeling algorithm so far. With exceptions as noted in the preceding paragraph, however, there remains a open gap with respect to lower bounds on problem complexity (*cf.* open problem, Table 8.g). This gap is especially striking with the  $O(n|E|)$  running time of  $\mathcal{A}_{\text{Label-Hamming}}$ , when its input is not Hamming (Sec. 2.5 of [LaForge et al 2006]).

We conclude with an example of how improving one kind of Hamming-labeling algorithm could benefit another. Suppose we want to decide whether an unweighted labeling  $\lambda$  of an  $n$ -vertex graph  $G$  is Hamming (thus  $n = |\lambda|$ ). In time  $\Theta(|\lambda|+|E|)$  first determine if  $G$  is connected; if not, it cannot be Hamming. Otherwise,  $|E| \geq |\lambda|-1$ , and a straightforward way to accomplish our task is to compute, in time  $O(|\lambda| \cdot |E|)$ , the all-distance matrix of  $G$  ([Corman et al 1993] Chap. 26). For each pair of vertices, compare the Hamming distance between their labels with their edge distance, in overall time  $O(d \cdot |\lambda|^2)$ , where  $d$  is number of digits in  $\lambda$ , and where computing the Hamming distance takes constant time per digit. Output "yes" if and only if the two distances are equal, for all vertex pairs. This is essentially Algorithm  $\mathcal{A}_{\text{Check-Hamming}}$  presented in [LaForge 2004], and is currently implemented by the Connection Foundry software shown in Figures 4, 8, and 13.

Recalling the next-to-last paragraph of Section 1.2, Algorithm  $\mathcal{A}_{\text{Construct-Hamming}}$  decides if a set  $\lambda$  of labels is Hamming graphic. The run-time dominant portion of  $\mathcal{A}_{\text{Construct-Hamming}}$  constructs the (unique) Gray-code-transitive components induced by  $\lambda$ ; the less costly portion determines if there is one just one component. Focusing on the former, suppose that we could close the  $\Theta(|\lambda|)$  gap between our  $\Omega(d \cdot |\lambda|)$  lower bound on problem complexity and the  $O(d \cdot |\lambda|^2)$  algorithmic running time (*cf.* open problem, Table 8.h). For example, suppose that we found a  $\Theta(d \cdot |\lambda|)$  running-time algorithm that constructs the graph induced by  $\lambda$ . This, in turn, would shave a  $\Theta(|\lambda|)$  factor from the running time of Algorithm  $\mathcal{A}_{\text{Check-Hamming}}$ , described above. Feed the labeling  $\lambda$  to the improved portion of  $\mathcal{A}_{\text{Construct-Hamming}}$  that constructs the graph induced by  $\lambda$ . Then check, in time  $\Theta(|\lambda|+|E|)$ , whether the graph induced by  $\lambda$  equals the graph input to the improved version of  $\mathcal{A}_{\text{Check-Hamming}}$ .

Pursuing a combinatorial perspective often sheds light on metrized graphs in general (*e.g.*, Table 8.i), Hamming graphs in particular. With respect to the  $\mu$ -Hamming graphs of Section 3.2, for example, how many of them are there, as a function of the order  $n$ , minimum dimension  $d$ , extent-minimum radix  $j$ , and (Table 8.j) suitably maximized weight vector  $\mu$ ? As a step in this direction, how many  $n$ -vertex trees are there, Hamming-labeled with minimum dimension  $n-1$ , and necessarily with minimum radix 2 and maximum weight  $\mathbf{1}$ ? The answer, as we now know (26), is  $2^{n-1}n^{n-3}$ .

## ACKNOWLEDGEMENTS

I am indebted to Jeffrey R. Moreland, digital engineer with The Right Stuff of Tahoe, and to M. Sami Fadali, electrical engineering professor at the University of Nevada, for collaborations leading to the work espoused here. Also, to Dr. Alan Lindsey, formerly of the Air Force Research Laboratory, and now with Austral Engineering and Software. Dr. Lindsey was one of the first to recognize the importance of a rigorous theory of computational connectivity – including the results of this paper – to network applications as summarized in Section 3.

## REFERENCES

- [Armstrong and Gray 1981] J. R. Armstrong and F. G. Gray. Fault diagnosis in a Boolean  $n$  cube of microprocessors. *IEEE Transactions on Computers*. **C-30** (8), Aug-1981. pp. 587 – 590. Cited p. 4, Table 7.
- [Aurenhammer and Hagauer 1995] F. Aurenhammer and J. Hagauer. Recognizing binary Hamming graphs in  $O(n^2 \log n)$  time. *Math. Systems Theory*. **28** (5), 1995. pp. 387 – 395. Cited Table 4, p. 27.
- [Avis 1981] D. Avis. Hypermetric spaces and the Hamming cone. *Canadian Journal of Mathematics*. **33**, 1981. pp. 795 – 802. Cited p. 26.
- [Avis 1978] D. Avis. *On the Hamming Cone*. Department of Operations Research Technical Report 77-5. Stanford University. Apr-1977. Cited pp. 25, 26, 26, 26, 26, 26, 27.
- [Bermond and Bollobás 1981] J.-C. Bermond and B. Bollobás. The diameter of graphs – a survey. *Congressus Numerantium*. **32**, 1981. pp. 3 – 37. Cited pp. 20, 21.
- [Biggs 1996] N. Biggs. *Algebraic Graph Theory*. Cambridge University Press. 2nd ed. 1996. Cited pp. 12, 20.
- [Bollobás 1998] B. Bollobás. *Modern Graph Theory*. New York: Springer-Verlag, 1998. Cited pp. 2, 15, 20.
- [Bollobás 1978] B. Bollobás. *Extremal Graph Theory*. London: Academic Press, 1978. Cited p. 20.
- [Bose et al 1995] B. Bose, B. Broeg, Y. Kwon, and Y. Ashir. Lee distance and topological properties of  $k$ -ary  $n$ -cubes. *IEEE Transactions on Computers*. **44** (8), Aug-1995. pp. 1021 – 1030. Cited pp. 4, 8, Table 7.
- [Brandenburg et al 1972] L. H. Brandenburg, B. Gopinath, and R. P. Kurshan. On the addressing problems of loop switching. *Bell System Technical Journal*. **51**, 1972. pp. 1445 – 1469. Cited p. 26.
- [Brouwer et al 1989] A. E. Brouwer, A. M. Cohen, and A. Neumaier. *Distance-Regular Graphs*. Berlin: Springer-Verlag, 1989. Cited Table 4, Table 7.
- [Cayley 1889] A. Cayley. A theorem on trees. *Quarterly Journal of Mathematics*. **23**. 1889. pp. 376 – 378. Cited p. 12.
- [Chartrand and Lesniak 1986] G. Chartrand and L. Lesniak. *Graphs and Digraphs*. Belmont, CA: Wadsworth, Inc, 1986. Cited pp. 2, 4, 20.
- [Choquette et al 2006] K. D. Choquette, J. J. Raftery, Jr., and A. C. Lehman. Beam steering in photonic crystal vertical cavity. *Proceedings, 2006 IEEE Aerospace Conference*. 4-Mar-2006. Cited p. 25.
- [Comtet 1974] L. Comtet. *Advanced Combinatorics*. Dordrecht, Holland: R. Reidel Publishing, 1974. 2nd ed. Cited pp. 7, 12, 15, 19, 19, 19, 20.
- [Corman et al 1993] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. Cambridge, MA: MIT Press. Tenth printing, 1993. Cited pp. 12, 15, 20, 20, 27, 27.
- [Erdős and Rényi 1962] P. Erdős and A. Rényi. On a problem in the theory of graphs. (In Hungarian). *Publ. Math. Inst. Hungar. Acad. Sci.* **7**, 1962. Cited p. 20.
- [Graham and Pollack 1971] R. L. Graham and H. O. Pollack. On the addressing problem for loop switching. *Bell System Technical Journal*. **50** (8), 1971. pp. 2405 – 2419. Cited pp. 26, 26.
- [Graham and Winkler 1985] R. L. Graham and P. M. Winkler. On isometric embeddings of graphs. *Transactions of the American Mathematical Society*. **288** (2), Apr-1985. pp. 527 – 536. Cited p. 4.
- [Hamming 1950] R. W. Hamming. Error detecting and error correcting code. *Bell System Technical Journal*. **29**, 1950. pp. 147 – 160. Cited p. 27.
- [Harary 1962] F. Harary. The maximum connectivity of a graph. *Proceedings, National Academy of Science*. **48**, 1962. pp. 1142 – 1146. Cited pp. 20, 21.
- [Hayes 2006] B. Hayes. Gauss's day of reckoning. *American Scientist*. **94** (3), May-Jun-2006. pp. 200 – 205. Online at [www.americanscientist.org](http://www.americanscientist.org). Cited p. 19.
- [Hayes 1974] J. P. Hayes. A graph model for fault tolerant computing systems. *IEEE Trans. on Computers*. **C-25** (9), Sep-1976, pp. 875 – 884. Cited pp. 20, 21.
- [Kautz 1968] W. H. Kautz. Bounds on directed  $(d, k)$  graphs. *Theory of Cellular Logic Networks and Machines*. AFCRL-68-0668 Final Report. 1968. pp. 20 – 28. Cited pp. 25, 25.

- [Kelly 1975] J. B. Kelly. Hypermetric spaces. In *Geometry of Metric and Linear Spaces*. L. M. Kelly, ed. Lecture Notes in Mathematics. **490**. New York: Springer-Verlag, 1975. pp. 17 – 31. Cited p. 26.
- [LaForge 1999] L. E. LaForge. *Fault Tolerant Physical Interconnection of X2000 Computational Avionics*. Pasadena, CA: Jet Propulsion Laboratory, document JPL D-16485. Revised 18-Oct-1999. Online at <http://faculty.erau.edu/laforge/>. Cited Table 4, pp. 21, 21.
- [LaForge 2000] L. E. LaForge. *Architectures and Algorithms for Self-Healing Autonomous Spacecraft*. Grant report, NASA Institute for Advanced Concepts, 9-Jan-2000, revised 28-Feb-2000. Online at [www.niac.usra.edu/studies/347LaForge.html](http://www.niac.usra.edu/studies/347LaForge.html). Cited p. 21.
- [LaForge 2004] L. E. LaForge. Hamming graphs. *Proc., 2004 IPSI: Internet, Processing, Systems for E-education/E-business, and Interdisciplinaries*. 9-Oct-2004. Cited pp. 4, 4, Table 4, Table 8, pp. 25, 27.
- [LaForge 2004 MANETs] L. E. LaForge. *Clique-Factorized Optimum MANET Throughput Enabled by Directed, Power-Controlled Antennas*. Technical report, U.S. DoD contract FA8750-04-C-0020. Reno, NV: The Right Stuff of Tahoe, Incorporated. 14-Apr-2004 revised 7-Aug-2004. Cited pp. 15, 26.
- [LaForge et al 2003] L. E. LaForge, K. F. Korver, and M. S. Fadali. What designers of bus and network architectures should know about hypercubes. *IEEE Transactions on Computers*. **52** (4), Apr-2003. pp. 525 – 544. Cited pp. 4, 4, 8, 20, 20, 21, Table 7.
- [LaForge et al 2006] L. E. LaForge, J. R. Moreland, and M. S. Fadali. Spaceflight multi-processors with fault tolerance and connectivity tuned from sparse to dense. *Proc. 2006 IEEE Aerospace Conference*. 4-Mar-2006. Cited p. 8, Table 4, pp. 17, 24, 25, 25, 26, 27.
- [LaForge et al 2006 VCSELS] L. E. LaForge, J. R. Moreland, R. G. Bryan, and M. S. Fadali. Vertical cavity surface emitting lasers for spaceflight multi-processors. *Proceedings, 2006 IEEE Aerospace Conference*. 4-Mar-2006. Cited p. 25.
- [Leonard Phone Conv 2008] L. E. LaForge to J. Leonard (SiCortex, +1.978.897.0214). Telephone conversation. 18-Jan-2008. Cited p. 25.
- [MacWilliams and Sloane 1998] F. J. MacWilliams and N. J. A. Sloane. *The Theory of Error Correcting Codes*. Amsterdam: Elsevier Science. Tenth printing, 1998. Cited pp. 8, 27.
- [Metcalfé 2007] R. M. Metcalfé. It's all in your head. *Forbes*. 7-May-2007. pp. 52 – 56. Online at [www.forbes.com/forbes/2007/0507/052.html](http://www.forbes.com/forbes/2007/0507/052.html). Cited p. 25.
- [Murty and Vijayan 1964] U. S. R. Murty and K. Vijayan. On accessibility in graphs. *Sakhya Ser. A*. **26**, 1964. pp. 299 – 302. Cited pp. 20, 21.
- [Ore 1962] O. Ore. *Theory of Graphs* (p. 102). Providence: American Math. Society, 1962. Cited p. 20.
- [Pierce 1972] J. R. Pierce. Network for block switching of data. *Bell System Technical Journal*. **51** (6), 1972. pp. 1133 – 1145. Cited p. 26.
- [Prüfer 1918] E. P. H. Prüfer. Neuer Beweis eines Satzes über Permutationen. *Arch. M. under Phys.* **27**, 1918. pp. 142 – 144. Cited p. 15.
- [Sampels 1997] M. Sampels. Large networks with small diameter. In *Proceedings, Graph-Theoretic Concepts in Computer Science, 23rd International Workshop*. R. H. Möhring, ed. Berlin: Oct-1997. pp. 288 – 302. Cited p. 20.
- [SiCortex Fabric 2006] L. C. Stewart and D. Gingold. *A New Generation of Cluster Interconnect*. Maynard, MA: SiCortex, Inc. White paper. Dec-2006. Cited p. 25.
- [Wakerly 1994] J. F. Wakerly. *Digital Design: Principles and Practices*. Englewood Cliffs, NJ: Prentice-Hall. 2<sup>nd</sup> Edition. 1994. Cited p. 13, 27.
- [Wang et al 2005] J. Wang, L. Li, S. H. Low, and J. C. Doyle. Cross-layer optimization in TCP / IP networks. *IEEE / ACM Transactions on Networking*. **13** (3), Jun-2005. pp. 582 – 595. Cited pp. 25, 26.

## BIOGRAPHICAL SKETCH

**Laurence E. LaForge** is President of the Right Stuff of Tahoe. Previously on faculty with Embry-Riddle Aeronautical University, he twice held NASA/ASEE Fellowships at the Jet Propulsion Laboratory, where he worked on bus fault tolerance and the *Deep Space 1* probe. He has been guest editor for the *IEEE Transactions on Components, Packaging, and Manufacturing Technology*, and program chair for the IEEE International Conference on Innovative Systems in Silicon. His baccalaureate in mathematics is from the Massachusetts Institute of Technology; he earned his PhD in computer science from McGill University.