GLOBALLY DENSE (d,k) GRAPHS FOR COMPUTER NETWORK ARCHITECTURES

by

Will Leland, Li Quiao,
Leonard Uhr

Computer Sciences Technical Report #411

December 1980

Globally Dense (d,k) Graphs for Computer Network Architectures

Will Leland, Li Quiao, Leonard Uhr
University of Wisconsin

Abstract

This paper examines the techniques used to increase the number of nodes in a graph of given degree (the number of edges at each node) and diameter (the shortest path between the most distant nodes). A number of new "densest" graphs (i.e., with more nodes for a given degree and diameter) have recently been found, using several new methods for compounding, and for heuristically and algorithmically completing trees. These will be surveyed and compared, and some general tools and principles for graph-building will be proposed.

Index Terms: (d,k) graphs, Moore graphs, computer networks, packing density, compounding, graphs of computers, (n,d,k) graphs, size of graphs, network architectures.

Minimizing distances between nodes is a key problem for computer network architectures, where it is important that vertices (processor or computer nodes) are connected by paths along edges (lines or links) that are as short as possible, so that delays and disruptions from message-passing are minimized. This leads to "denser" (n,d,k) graphs, that is, with more nodes, n, for a given degree, d and diameter, k.

Hoffman and Singleton [1] and their students [14] proved that only the (n,d,2) complete graphs, the (n,2,k) polygon graphs, and (10,3,2) (the Petersen graph [2]), (50,7,2) and (if it exists) (3250,57,2) meet the upper "Moore Bound" n_b [3]:

$$n_b(d,k) = \frac{(d-1)^k - 2}{d - 2}$$

Elspas [4] described several construction methods that gave a number of non-optimal but densest-so-far graphs. A number of denser graphs were achieved by an elegant construction of Akers

[5]. Friedman [6], Korn [7], and Storwick [8] devised techniques for connecting copies of trees (results summarized in [8]). Recently, Arden and Lee [9] achieved still better results for d=3 graphs by combining trees, and Toueg and Steiglitz [10] used a directed-search computer program to find several additional denser d=3 and d=4 graphs.

Compounding Techniques for Denser (n,d,k) Graphs

A new compounding operation devised by Uhr [11] makes n+1 copies of an n-node graph and connects each copy to every other copy by adding a new link between one of the n nodes in each (see Figure 1 for one of the $(n!)^n$ ways this can be done):

Take any regular (n,d,k) graph, G.

Form n+1 copies, $G_0, G_1, \ldots G_n$.

For each node, $G_{i,j}$ (i = copy, j = node),

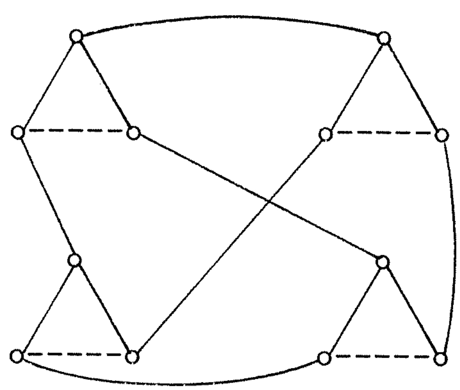
$i = 0,1,\ldots n-1$, $j = i,i+1,\ldots n$, $k = i+1,i+2,\ldots n$, $l = i$;

link (in ascending order of i,j,k) the node $G_{i,j}$ to $G_{k,l}$.

The resulting Compound Graph, C, is

$(n_G(n_G+1), d_G+1, <=2k_G+1)$.

Figure 1. An Example of The Compound (12,3,3) of (3,2,1)



First Extensions and Explorations

After hearing about this compounding technique and its results, W. Leland reported the observation that a (n_G, d_G, k_G) "cluster" graph, G, can be compounded by any regular $H(n, n_G, k)$ "lacer" graph, H (that is, where the lacer's degree equals the cluster's number of nodes, joining each node to one link - call this "embossing"), to give an $I(n_G * n_H, d_G+1, k_G+2k_H)$. J. Halton [12] made the slight correction: $I(n_G * n_H, d_G+1, k_G(k_H+1))$. Leland [13] used (91,10,2) to compound the Petersen graph (10,3,2), giving (910,4,8) [improving on Friedman's (320,4,8)]. In general, these graphs will not be as symmetric as those constructed using complete graphs, and graphs of rather high degree are needed. But this extension makes possible a larger set of improvements over previous best graphs. And an intriguing possibility exists - to emboss (57,8,2) into the yet-to-be-discovered (3250,57,2).

Halton [12] has observed that $k_G(k_H+1)$ is actually the upper bound for this construction. There is some (small) possibility of reducing such a compound's diameter with judicious choices of nodes and links to lace together. Uhr noted that the clusters embossed into the lacer need not all be copies of the same graph; they need only have the same degree. This opens up more possibilities for reducing k_I below the Halton bound, e.g., by using several different clusters of the sort found in the Coxeter graph [14].

Halton and Uhr have been exploring ways of using a few scattered nodes of higher degree to "lace the graph tighter." This suggests using a measure like average degree (much like average distance) rather than insisting on regular graphs. It also seems reasonable for a computer network, where, e.g., almost all nodes have 4 links but a few "waystation" nodes might have 8.

Uhr and Halton have observed that the compounding technique can be generalized to use any lacer graph to compound any collection of (identical or different) clusters (that can vary in n, d and/or k). A cluster must now be assigned to several nodes in the lacing graph, and there must be additional links between those nodes to accommodate the links between the several parts of the cluster. In the general case each cluster can be laced using an arbitrary sub-graph of the lacer, and it therefore may become more likely, especially as n grows larger than D, that reductions in the diameter of the compound can be achieved.

Denser Graphs From de Bruijn Networks (Shift Registers)

Imase and Itoh [15] have developed an entirely different type of procedure that gives increasingly better graphs beyond 6,000 to 20,000 nodes (although rather poor graphs up to that point): Label n nodes 0,1,2,..,n-1; link all n_i to n_j that satisfy the equation: $j=i*d+a \pmod n$ $a=0,1,..d-1$. This apparently independent rediscovery of de Bruijn networks [16] (shift registers) applies them to the problem of density, and proves these graphs to have $[d/2]^k$ nodes.

These graphs, although relatively dense, can certainly be improved, since this method produces 2 loops and many short cycles (although if only minor improvements could be made it might be that only average distance, and not diameter, could be reduced). De Bruijn networks are intriguingly similar to the "perfect-shuffle" [17] and similar interconnection networks;

these similarities should be investigated. Goodman and Sequin [18] have compared their "hypertrees" (trees augmented with perfect-shuffle-like regularizing links that attempt to draw most-distant nodes closer) with de Bruijn networks with respect to average distance (but not diameter), and hypertrees may be 10% or so denser.

De Bruijn networks can be viewed as first constructing a (directed) tree, and then continuing to add links until a regular graph of degree 1 greater than the original tree's degree is completed. This, plus the fact that compounding also raises degree, suggests that adding links to increase degree may be a promising way to draw a graph closer together. (Most researchers have tended to start with a tree and then work only with different interconnections among its buds.) It also seems likely that an algorithm that continues to link pairs of nodes, but chooses each pair because it is "now-farthest-apart" would improve upon these results (as Goodman and Sequin do). Alternately, such an algorithm might be applied only to the buds.

Heuristic Searches to Augment Trees for Denser Graphs

Toueg and Steiglitz [10] used directed search computer programs to look for denser graphs. They looked only for graphs with 50 or fewer nodes, and remarked that 150 would be the limit for reasonable amounts of computing time. But Leland [13] has succeeded in developing and refining a set of heuristics for a directed search program that augments a tree, essentially by linking together its most distant nodes, achieving a number of new best graphs (with respect to density, although it is likely that they are less symmetric, and have fewer desirable characteristics, than graphs achieved by compounding good clusters), up through 525 nodes.

Such heuristic search techniques might also prove useful for finding good combinations of clusters, lacers, and waystations, and for searching with a whole set of properly weighted criteria, and not diameter alone.

Additional Graph Compounding Operations

Leland has made several important additional discoveries [13]. First, he noticed that although Elspas originally suggested taking the Cartesian product of two graphs, Storwick's table does not include several "best entries" that are simply Cartesian products. (The Cartesian product of two graphs, G and H with G and H vertices, gives n_G*n_H nodes, degree d_G+d_H, diameter k_G+k_H.)

Next, Leland augmented the Cartesian product of two graphs of the same degree, by splitting each node in the product graph in two and connecting them (doubling n, reducing degree to d+1, and increasing diameter to k_G+k_H+2), and thus achieved a number of new best graphs. [Linking n_G copies of H to n_H copies of G, forming a complete bipartite graph into which the G and H copies have been embossed, gives this same construction.]

Leland further "regularizes" the Cartesian product of two graphs of <u>different</u> degree, $d_G <= d_H$, by making m copies of G, interconnecting them using $(d_H-d_G)+m$ copies of H (this replaces each (2,1,1) line-graph linking each split pair of nodes). The resulting graph has $(2m+s)*n_G*n_H$ nodes, $d = m+d_H$, $k <= 2+k_G+k_H$.

Yet another useful compounding operation has been discovered by Li Qiao [19]: Emboss G copies of a graph G into each node of a graph H, connecting those in adjacent nodes of H to form a complete bipartite graph. This gives $n_G * n_G * n_H$ nodes, degree $d_G + d_H$, diameter $<= 2k_G + \max(2, k_H)$. [Li also independently discovered the Leland splitting of the Cartesian product.]

Toward a Set of Tools for Building Improved Graph Structures

In addition to many new denser graphs, produced by several new compounding, construction and heuristic search techniques, a variety of successful tools to build, stretch, pare, lace and shape graphs appear to be emerging. Compounds can be drawn together, split apart, and connected back upon themselves.

The standard way (among many) of drawing the Petersen Graph (10,3,2) is as a 5-node star, each node linked to the nearby node of a circumscribing pentagon. The links between star and pentagon are suggestive of the links bridging between clusters in (n(n+1),d+1,<=2k+1) graphs, and in Leland's split Cartesian product Graphs. Such a link stretches the graph's diameter slightly, but almost halves degree, and doubles the number of nodes. The Singleton graph (50,7,2) links each of 5 pentagons to each of 5 stars, suggestive of the linking of each copy of a graph to every other copy to form a compound.

Table 1. Some of the High-Density (n,d,k) Graphs

d\k	2	3	4	5	6	7	8	9	10
3(Best)	10*P	20*E	34h	56As	78h	122h	176h	311h	525h
&Best:			30TS		72As	120As	164As		
Storwk:	10*P	20*E	28E	36E	60S	66S	90F	138S	216S
MBound:	10	22	46	94	190	382	766	1534	3070
4(Best)	15*E	35A	67h	134h	261h	425h	910er	1360s1	2312s2
&Best:			45A	110ec1	200s3	420ec2		1200s4	2240rs1
Storwk:	15*E	35A	40E	62S	114S	188S	320F	566S	996S
MBound:	17	53	161	485	1457	4373	13121	39365	118097
5(Best)	24*E	48h	126A	262h	505h	1260ec3	2450s5	4690s6	9380s7
&Best:				240ec4	450s8				
Storwk:	24E	36E	126A	120E	232S	442S	850F	1770S	3512S
MBound:	26	106	426	1706	6826	27306	109226	436906	17e6
6(Best)	31E	65h	164h	600ec5	1152s9	2520rs2	6561sr	19683sr	59049sr
&Best:			105Cp1			1728s10	6048s11	16002ec6	31752s12
Storwk:	31E	55E	105A	462A	447S	867S	1872F	4317S	9465S
MBound:	37	187	937	4687	23437	117187	585937	2929687	14648437
7(Best)	50*HS	88h	252h	992ec7	2880eb1	4680rs3	12250eb2	43200rs4	86400rs5
&Best:			150Cp2	378A	2304rs6	3410s13	12096rs7		71424s14
Storwk:	50*HS	80E	150E	378A	1716A	1574S	3626F	9422S	22836S
MBound:	50	302	1814	10886	65318	391910	3351462	141e5	846e5
8(Best)	57E	105E	384eb3	2550ec8	5760eb4	16384sr	65536sr	262e3sr	104e4sr
Storwk:	57E	105E	175E	504E	1716A	1574S	3626F	9422S	22836S
MBound:	65	457	32011	22409	156865	1098057	7686401	538e5	377e6
9(Best)	74E	150Cp3	600eb5	3306ec9	12500eb6	20160eb7	76500rs8	38e4rs9	10e5db
Storwk:	74E	150Cp	240Cp	666E	1904S	5148A	24310A	32706S	94416S
Mbound:	82	658	5266	42130	337042	2696338	216e5	173e7	138e7
10Best:	91S	200Cp4	864eb8	5550ec10	25000eb9	78125sr	39e3sr	19e4sr	98e5sr
Storwk:	91S	200Cp	320E	910E	2780S	6864A	19305A	92378A	170685S
MBound:	101	911	8201	73811	664301	5978711	538e5	484e6	436e7

Best: densest graph to date. &Best: additional dense graphs.
Storwk: Storwick table [8]. Mbound: Moore bound [1,13] *: Maximal
Cp: Cartesian product [4] 1:(4,3)x(2,1) 2:(4,2)x(3,2) 3:(7,2)x(2,1) 4:(7,2)x(2,1)
A: Akers [6] E: Elspas [4] F: Friedman [5] HS: Hoffman-Singleton [1]
P: Petersen [3] S: Storwick [8] AL: Arden-Lee [9] TS: Toueg-Steiglitz [10]
sr: shift register (de Bruijn network) [15]
h: heuristic search (Leland)
eb: embossed bipartite compound (Li) 1:(5,2)x(2,2) 2:(4,3)x(3,2)
3:(3,1)x(5,2) 4:(5,2)x(3,2) 5:(4,1)x(5,2) 6:(7,2)x(2,2)
7:(5,2)x(4,3) 8:(5,1)x(5,2) 9:(7,2)x(3,2)
ec: embossed complete (Uhr) 1:(3,2) 2:(3,3) 3:(4,3) 4:(4,2)
5:(5,2) 6:(5,4) 7:(6,2) 8:(7,2) 9:(8,2) 10:(9,2)
er: embossed regular compound (Leland-Uhr) (3,2)x(10,2)
s: split (Leland) 1:(3,3)x(3,4) 2:(3x4) 3:(3,2) 4:(3,3)x(3,4)
5:(4,3) 6:(4,3)x(4,4) 7:(4,3)x(4,5) 8:(4,2) 9:(5,2)
10:(5,2)x(5,3) 11:(5,2)x(5,4) 12:(5x4) 13:(6,2)x(6,3) 14:(6,2)x(6,6)
rs: regularized split (Leland) 1:(3,3)x(3,5) 2:(4,3)x(5,2) 3:(5,2)x(6,3)
4:(5,2)x(6,5) 5:(5,3)x(6,5) 6:(5,2) 7:(5,2)x(5,4) 8:(4,1)x(8,5) 9:(7,2)x(8,5)

Table 1. Some of the High-Density (n,d,k) Graphs

d\k	2	3	4	5	6	7	8	9	10
3(Best)	10*P	20*E	34h	56As	78h	122h	176h	311h	525h
&Best:			30TS		72As	120As	164As		
Storwk:	10*P	20*E	28E	36E	60S	66S	90F	138S	216S
MBound:	10	22	46	94	190	382	766	1534	3070
4(Best)	15*E	35A	67h	134h	261h	425h	910er	1360s1	2312s2
&Best:			45A	110ec1	200s3	420ec2		1200s4	2240rs1
Storwk:	15*E	35A	40E	62S	114S	188S	320F	566S	996S
MBound:	17	53	161	485	1457	4373	13121	39365	118097
5(Best)	24*E	48h	126A	262h	505h	1260ec3	2450s5	4690s6	9380s7
&Best:				240ec4	450s8				
Storwk:	24E	36E	126A	120E	232S	442S	850F	1770S	3512S
MBound:	26	106	426	1706	6826	27306	109226	436906	17e6
6(Best)	31E	65h	164h	600ec5	1152s9	2520rs2	6561sr	19683sr	59049sr
&Best:			105Cp1			1728s10	6048s11	16002ec6	31752s12
Storwk:	31E	55E	105A	462A	447S	867S	1872F	4317S	9465S
MBound:	37	187	937	4687	23437	117187	585937	2929687	14648437
7(Best)	50*HS	88h	252h	992ec7	2880eb1	4680rs3	12250eb2	43200rs4	86400rs5
&Best:			150Cp2	378A	2304rs6	3410s13	12096rs7		71424s14
Storwk:	50*HS	80E	150E	378A	1716A	1574S	3626F	9422S	22836S
MBound:	50	302	1814	10886	65318	391910	3351462	141e5	846e5
8(Best)	57E	105E	384eb3	2550ec8	5760eb4	16384sr	65536sr	262e3sr	104e4sr
Storwk:	57E	105E	175E	504E	1716A	1574S	3626F	9422S	22836S
MBound:	65	457	32011	22409	156865	1098057	7686401	538e5	377e6
9(Best)	74E	150Cp3	600eb5	3306ec9	12500eb6	20160eb7	76500rs8	38e4rs9	10e5db
Storwk:	74E	150Cp	240Cp	666E	1904S	5148A	24310A	32706S	94416S
Mbound:	82	658	5266	42130	337042	2696338	216e5	173e7	138e7
10Best:	91S	200Cp4	864eb8	5550ec10	25000eb9	78125sr	39e3sr	19e4sr	98e5sr
Storwk:	91S	200Cp	320E	910E	2780S	6864A	19305A	92378A	170685S
MBound:	101	911	8201	73811	664301	5978711	538e5	484e6	436e7

Best: densest graph to date. &Best: additional dense graphs.
Storwk: Storwick table [8]. Mbound: Moore bound [1,13] *: Maximal
Cp: Cartesian product [4] 1:(4,3)x(2,1) 2:(4,2)x(3,2) 3:(7,2)x(2,1) 4:(7,2)x(2,1)
A: Akers [6] E: Elspas [4] F: Friedman [5] HS: Hoffman-Singleton [1]
P: Petersen [3] S: Storwick [8] AL: Arden-Lee [9] TS: Toueg-Steiglitz [10]
sr: shift register (de Bruijn network) [15]
h: heuristic search (Leland)
eb: embossed bipartite compound (Li) 1:(5,2)x(2,2) 2:(4,3)x(3,2) 3:(3,1)x(5,2) 4:(5,2)x(3,2) 5:(4,1)x(5,2) 6:(7,2)x(2,2) 7:(5,2)x(4,3) 8:(5,1)x(5,2) 9:(7,2)x(3,2)
ec: embossed complete (Uhr) 1:(3,2) 2:(3,3) 3:(4,3) 4:(4,2) 5:(5,2) 6:(5,4) 7:(6,2) 8:(7,2) 9:(8,2) 10:(9,2)
er: embossed regular compound (Leland-Uhr) (3,2)x(10,2)
s: split (Leland) 1:(3,3)x(3,4) 2:(3x4) 3:(3,2) 4:(3,3)x(3,4) 5:(4,3) 6:(4,3)x(4,4) 7:(4,3)x(4,5) 8:(4,2) 9:(5,2) 10:(5,2)x(5,3) 11:(5,2)x(5,4) 12:(5x4) 13:(6,2)x(6,3) 14:(6,2)x(6,6)
rs: regularized split (Leland) 1:(3,3)x(3,5) 2:(4,3)x(5,2) 3:(5,2)x(6,3) 4:(5,2)x(6,5) 5:(5,3)x(6,5) 6:(5,2) 7:(5,2)x(5,4) 8:(4,1)x(8,5) 9:(7,2)x(8,5)

The (2,2k+1) polygons can be used to build (n,d,k) graphs [e.g., the (10,3,2) Petersen graph can be constructed from one (2,5) by taking each adjacent pair of nodes as the start of another (2,5)]. In general, this gives graphs with high girth, which often appears to be associated with low diameter (e.g., the Moore graphs have maximal girth and minimal diameter).

Trees can be laced internally-at-great-distances, and at their buds. Heuristically completed trees, de Bruijn shift registers and hypertrees are successful examples of such an approach.

It seems possible that a systematic technique for completing a tree, by lacing its buds together, might reduce diameter sufficiently close to (k/2)+1 to improve upon the de Bruijn networks. In small graphs most (over 90%) of the nodes can be pulled within (k/2)+1 without much trouble, by linking "most distant" nodes, or linking according to a pattern like i-i((d/2)+1). Possibly more important, this suggests that diameter is too coarse a measure, one that greatly overemphasises boundary conditions, in contrast to weighted average distance, which gives a far more sensitive measure, and one that would provide more feedback in lacing graphs tighter.

A general technique appears to be emerging as underlieing the good compounding operations - replacing each node in complete graphs, and in complete bipartite graphs, with a carefully chosen cluster containing as many nodes as the degree of the graph replacing each node (called "embossing"). The original construction embosses into a complete graph; Leland's split embosses into a complete bipartite graph; and Li's construction embosses complete bipartite graphs into other graphs. It appears that other combinations of this sort may give additional good graphs, and, possibly, some still denser ones. For example, complete bipartite graphs can be embossed into adjacent nodes of the Cartesian product (extending Li's construction).

Since compounding depends upon good clusters, as better clusters are found there should be more winning compounds. Several promising variants on compounding have not yet been investigated at all: e.g., the compounding of distant clusters in a larger graph, the embossing of n node graphs into n copies of another graph, the embossing of bipartite and n-partite graphs into another graph, and the lacing of distant clusters together - as trees are laced together by linking distant nodes.

Local Structural Properties of Networks and Algorithms

There is reason to think the compounds have especially good local properties. Graphs that complete trees achieve high densities by having most pairs of nodes diameter distance apart (just as most nodes in a tree are maximally distant from the root). In contrast, compounds tend to distribute nodes evenly throughout the graph.

Since compounding can be effected over any type of cluster, the clusters can be chosen for whatever set of properties is deemed most desirable, and not merely good diameter. The Moore graphs, including (d,1) and (2,k), are especially strong candidates - not only because of their optimal densities but also because they are highly symmetric, and have maximal girth and connectivity.

When the structure of algorithms to be executed on the network is known, and programs are mapped onto local regions of the network so that they can be executed as efficiently as possible, the program's structure can best be handled by using clusters with that same structure. For example, arrays often have the best structure for pattern recognition programs, and lattices for programs that model sections of the physical world. Now the diameter of the sub-graph actually used by an individual program becomes much more relevant than the diameter of the whole graph. So we should probably prefer graphs with good local, rather than global, density.

This suggests the need for a good measure of the local properties of sub-graph clusters of a larger graph, and a better grasp on the whole set of properties with which to evaluate graphs to be used for large computer networks. But from the point of view of local density and symmetry, and other local properties, the compounded graphs appear much better than the globally denser de Bruijn networks and heuristically augmented trees.

Summary Discussion

All of these newly found graphs are substantially denser than those that have been implemented, or proposed, for computer networks (e.g., rings, n-cubes, stars, arrays, trees), with the exception of hypertrees [18]. E.g., an 8-cube is (256,8,8), whereas compounding gives (240,5,5) and (12250,7,8) [Li], and the de Bruijn network gives (65536,8,8).

It seems likely that additional tools, better combinations and greater skill in using them will come with more experience. This should lead to substantial additional improvements in global density.

It seems especially important to develop a good set of global and local criteria relevant to the formal and structural aspects of good structures, and then use these to find and evaluate still other networks. The networks that have already been found should then be re-evaluated, since those that are "best" with respect to global density may not be best with respect to a more appropriate, or a more complete, set of criteria.

At the present time, up to roughly 500 nodes the densest graphs are those found by Leland's heuristic search program (which completes a tree). Imase and Itoh prove that de Bruijn networks are asymptotically best found so far, and they start being densest around 20,000 nodes. (But de Bruijn networks can be improved upon, at least slightly. And Goodman and Sequin's comparisons with hypertrees, using average distance, suggest that hypertrees may be better still.) In the middle ground, between 500 and 20,000 nodes, the compounding methods appear best.

With three quite different types of approaches yielding new results and denser graphs, good reason to believe these methods can be further strengthened, and real possibilities of applying judicious combinations of these methods where each is appropriate, it now appears that computer architects will be able to consider a far wider choice of substantially denser graphs for actual implementations of networks.

It should also be possible to develop systematic techniques for evaluating architectures on a wider range of criteria, including those that focus on local, and on structural properties. This should lead to the discovery of many appropriate new candidates for computer network architectures.

References

[1] Hoffman, A.J. and Singleton, R.R., On Moore graphs with diameter 2 and 3, IBM J. Res. Devel., 1960, 4, 497-504.

[2] Moore, E.F., unpublished communication to A.J. Hoffman.

[3] Petersen, J., Die Theorie der regularen Graphen, Acta Math., 1891, 15, 193-220.

[4] Elspas, B., Topological constraints on interconnection-limited logic, Switching Circuit Theory and Logical Design, 1964, S-164, 133-147.

[5] Akers, S.B., On the construction of (d,k) graphs, IEEE Trans. Elect. Computers (Short Notes), 1965, EC-14, 488.

[6] Friedman, H.D., A design for (d,k) graphs, IEEE Trans Electronic Computers (Short Notes), 1965, EC-14, 488.

[7] Korn, I., On (d,k) graphs, IEEE Trans. Electronic Computers (Short Notes), 1967, EC-15, 253-254.

[8] Storwick, R.M., Improved construction for (d,k) graphs, IEEE Trans. Computers (Short Notes), 1970, 19, 1214-1216.

[9] Arden, B.W. and Lee, H., A multi-tree-structured network, Proc. Fall Compcon 78, 1978, 201-210.

[10] Toueg, S. and Steiglitz, K., The design of small-diameter networks by local search, IEEE Trans. Computers, 1979, EC-28, 537-542.

[11] Uhr, L., Compounding denser (d,k) graph architectures for computer networks, Computer Sciences Dept. Tech. Rept., Univ. of Wisconsin, 1980.

[12] personal communication.

[13] personal communication.

[14] Bondy, J.A. and Murty, U.S.R., Graph Theory with Applications, New York: Elsevier, 1976.

[15] Imase, M. and Itoh, M., Design to minimize diameter on building-block networks, paper in press, 1980.

[16] de Bruijn, D.G., A combinatorial problem, Koninklijke Nederlandsche Academie van wetenschappen et Amsterdam, Proceedings of the Section of Sciences 49, 1946, 7, 758-764.

[17] Stone, H.S., Parallel processing with the perfect shuffle, IEEE Trans. Computers, 1971, 20, 153-161.

[18] Goodman, J.R. and Sequin, C.H., Hypertree, a multiprocessor interconnection topology, submitted for publication.

[19] personal communication.