**Degree-Constrained Pyramid Spanners**

Dana Richards and Arthur L. Liestman

Computer Science Report No. TR-90-16
July 10, 1990

Degree-Constrained Pyramid Spanners

Dana Richards
University of Virginia

Arthur L. Liestman
Simon Fraser University

ABSTRACT

A $t$-spanner of a pyramid network is a subnetwork in which every two nodes that were connected by an edge in the original pyramid can be connected by a path in the subnetwork with at most $t$ edges. We give several results that present trade-offs between $t$ and the maximum degree of a $t$-spanner.

## 1. Introduction

The use of parallel computers configured with a pyramid interconnection network has been increasingly investigated [DI 89, TANI83, UHR86]. These have been used in many application areas, principally in image processing. Only a few small pyramid computers have actually been fabricated. There are several reasons for this, not the least of which is the cost.

In this paper we explore one obstacle to fabrication. In particular, each node of the pyramid has large (but constant) degree. We explore the effect of using a subnetwork of the pyramid so that all nodes have small degree. There are two possible applications of this. First, this would permit prototyping of some larger pyramids with off-the-shelf low-degree components. Second, it would ease the layout of a VLSI implementation, since it would use less area for wires and each node could be smaller if it had fewer wires coming into it. In particular, if the maximum degree was four then a rectilinear wiring would permit small square components; in many applications the logic at each node is quite simple.

Our approach begins with a pyramid interconnection and removes edges between nodes. In the resulting subnetwork any missing edge between two nodes is effectively replaced by a (short) path, over the remaining edges, between the same two nodes. We should emphasize that this is distinct from another related technique in the literature: graph embedding. In the latter case the pyramid nodes would be mapped onto the nodes of another (presumably extant) fixed network and the pyramid edges would be mapped onto corresponding paths in the fixed network. (The pyramid is often called the "logical network" and the fixed network is the "host network".) For example, recently pyramids have been mapped onto the hypercube network (see [LAI90] and the references therein).

In section 2 we present some preliminaries including the 2-dimensional pyramid case. Section 3 presents various solutions to the 3-dimensional problem. The final section discusses some generalizations.

## 2. Preliminaries

A network, in this case a pyramid, is represented by a simple graph $G$. A spanning subgraph of $G$, $H$, is a $t$-spanner if for any edge $(a, b)$ of $G$ there is a path from $a$ to $b$ in $H$ with $t$ or fewer edges [PELE89]. The value $t$ is known as the *dilation* (or "stretch factor"). Obviously we want to have a small dilation and a sparse $H$; there are trade-offs between these goals. The maximum degree of $H$ is denoted $\Delta$.

The *congestion* of $H$, denoted $\gamma$, is a measure of the amount of traffic that can be expected across the edges of $H$. Given an assignment of paths in $H$ (for corresponding edges in $G$) the congestion of an edge in $H$ is the number of paths that use that edge. Then $\gamma$ is the maximum congestion of any edge in $H$, where $\gamma$ is minimized over all assignments of paths.

In this paper we are principally concerned with the 3-dimensional pyramid scheme (see [DI 89] for more formal definitions). In this scheme there are a series of layers with $4^i$ nodes on layer $i$, starting with 1 node on layer 0. The nodes on each layer are arranged in a $2^i \times 2^i$ square grid graph or "mesh", where each node is connected to its four orthogonal neighbors. Further each node is connected to the four corresponding nodes on the layer below it. This is seen in Figure 1, which shows a 4 layer pyramid. Since each node is also connected to its parent the degree of each node is 9 in general, ignoring the effect of the outer boundary. (Other pyramids are discussed in the last section.)

Formal presentations of the results in the next section tend to lead to an abundance of complex indexing schemes and cumbersome notations that tend to obscure the insights behind the results. In this section we discuss the simpler 2-dimensional pyramid, known as the X-tree, to introduce the informal style of argument used in the next section.

An *X-tree* is a full balanced binary tree with all the leaves on the lowest level. Further, all the nodes on a level (or "layer") are connected, left-to-right, in a path; see Figure 2c. In general an X-tree node has degree 5.

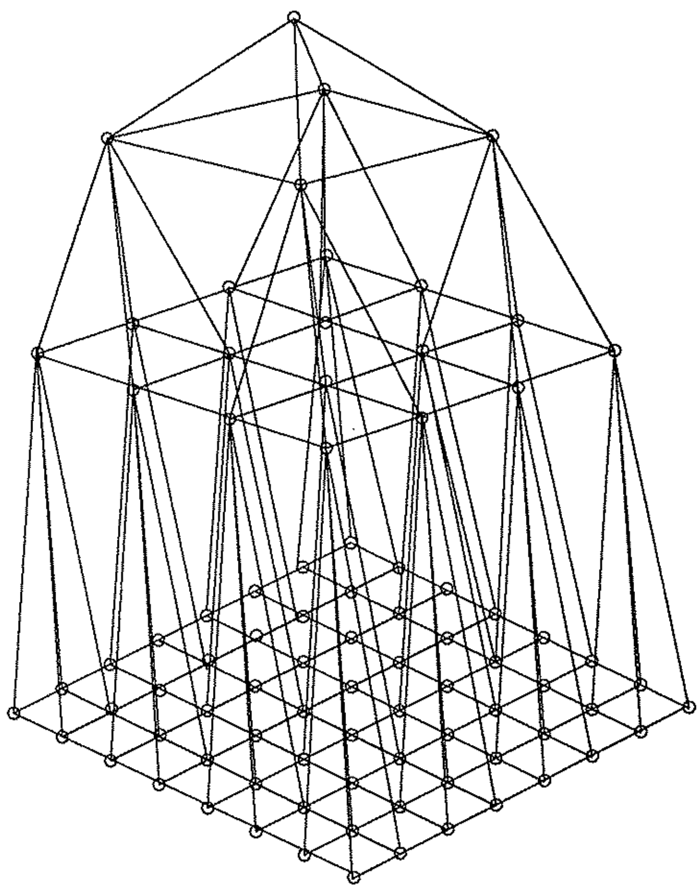
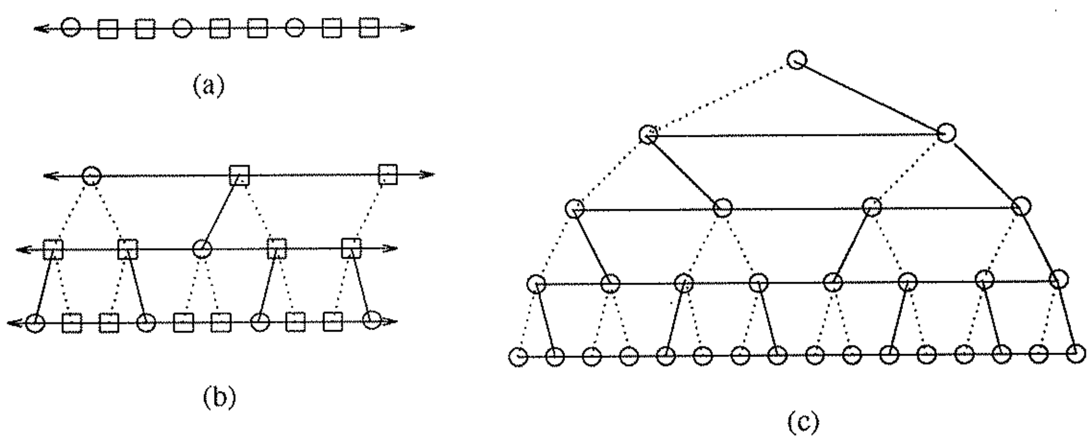


Figure 1.

We present a 3-spanner for the X-tree with $\Delta = 3$ and $\gamma = 3$. We normally just specify how the nodes are connected on one generic layer and which nodes are connected to parents and which nodes are connected to at least one child. For the X-tree this is shown in Figure 2a, where a circle indicates a node that is connected to its parent and a square indicates a node connected to a child. This is a simple example since every edge on each layer is used. Edges not in the spanner are shown with dotted lines.

The representation in Figure 2a essentially defines the entire scheme but the careful reader will want to supply additional details. First, one would draw two or more consecutive layers, ignoring the boundary, to visualize the pattern; see Figure 2b. One quickly sees there are only two types of missing edges: the edge between a square parent and a square child (with dilation 2) and the edge between a circle parent and a square child (with dilation 3). Hence $t = 3$ and it follows, by considering the paths that are necessarily involved, that $\gamma = 3$. Finally, $\Delta = 3$ since each node connects to at most one node above or below of its layer.

The above analysis ignores the effect of boundaries. A careful reader can see, as shown in Figure 2c, that these details can be worked out. Note that additional edges on the right boundary, not part of the pattern, were necessary. In general it is easy to add such edges without affecting $\Delta$ since on the boundary the degrees are truncated due to absent neighbors. Below we omit such



(a) (b) (c)

Figure 2.

details that can readily supplied by the reader.

We can show that if $\Delta = 3$ then there exists an (arbitrarily large) X-tree with no 2-spanner. Call a horizontal edge connecting two children of the same parent a *sibling edge*, and the other horizontal edges are *cross edges*. Note that $t = 2$ implies that every cross edge is used. Further every parent must be connected to one of its two children. Therefore one of each pair of siblings has degree 3 already so there can be no sibling edges. Hence every parent is connected to both children, forcing some nodes to have degree 4, giving a contradiction. (There is a simple 2-spanner with $\Delta = 4$: use all the edges except to one child from each parent.) Note that the above argument ignores boundary conditions; this is possible since the argument involves local constraints and can be applied to nodes far from the boundary.

## 3. Sparse Pyramid Spanners

In this section we present several results about spanners of 3-dimensional pyramids, hereafter just "pyramids". In particular we explore the trade-offs between $\Delta$ and $t$. The constructive results hold for all pyramids, regardless of the number of layers. The figures in this section show a *canonical block*: an $8 \times 8$ array of nodes from one layer in the interior of the pyramid such that the upper left square of four nodes share one parent. Every sufficiently large layer of the pyramid is defined by a "tiling" by translations of the same canonical block. (In general layer 1 is just one quarter of the canonical block.)

**Proposition 1**: There exists a 2-spanner of every pyramid with $\Delta = 6$ and $\gamma = 3$.

**Proof**: Using the conventions described in the previous section the canonical block in Figure 3 describes such a spanner. In this scheme we see that every node is connected to some of its children. Further the only edges missing from a layer connect two of four siblings, and these two siblings are connected to their parent while the other two siblings are not. There are two types of missing edges (between siblings on a layer or between a child and its parent) and in each case the dilation is clearly 2. Since each edge to a parent must be used on 3 paths we see $\gamma = 3$. □

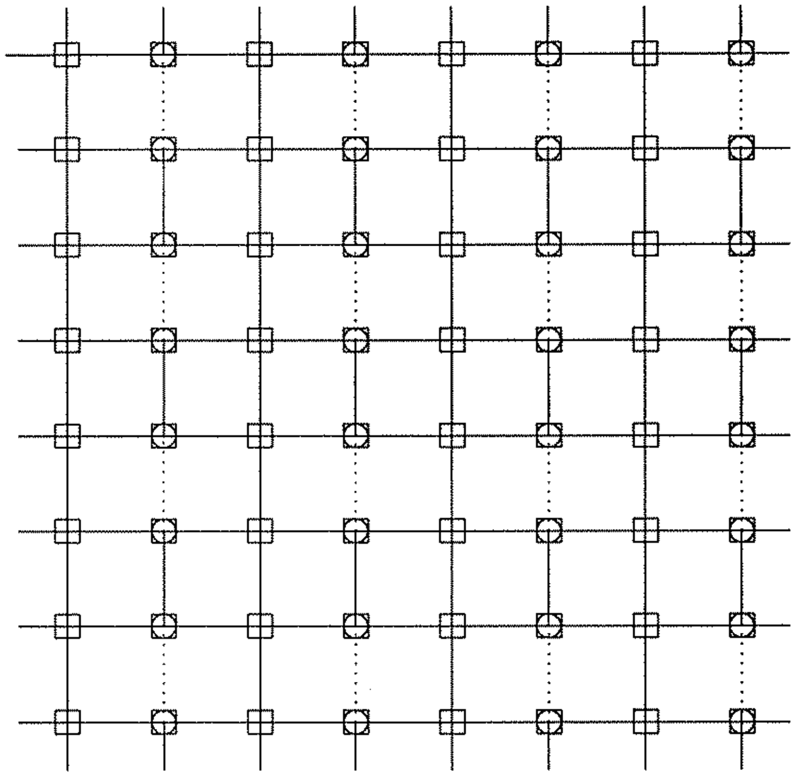


Figure 3.

For $t = 2$ this is as small as $\Delta$ can be.

**Proposition 2**: It is not possible to find 2-spanner with $\Delta = 5$ for all pyramids.

**Proof**: As in the previous section we define sibling and cross edges on each layer; in general each node is incident on two of each. If $t = 2$ then each cross edge must be used. Further each node must connect to two of its children, otherwise there must be one of its children it cannot reach in two steps. So far each node has four required edges. A node must also be able to reach its two neighboring siblings by paths of length at most two. Since none of the four required edges can contribute to these paths, the only way to reach both of its neighboring siblings, by adding at most one new edge, is to connect to the parent. Thus each node must be connected to its parent, each parent must be connected to all four of its children and some nodes will have degree at least six. □

A scheme for $\Delta = 5$ and $t = 3$ is possible but we can improve upon it.

**Proposition 3**: There exists a 3-spanner of every pyramid with $\Delta = 4$ and $\gamma = 4$.

**Proof**: The canonical block for such a spanner is shown in Figure 4. Note that each parent is connected to exactly one of its children. Further the one sibling connected to its parent has degree two within its layer so $\Delta = 4$.

Any missing edge on a layer can be routed with a path of three edges on the same layer. A missing edge to a parent can be routed through the circled sibling with a path of at most three edges. Hence $t = 3$. The most congested edges will be the links to parents giving $\gamma = 4$. □

For $t = 3$ this is as small as $\Delta$ can be.

**Proposition 4**: It is not possible to find 3-spanner with $\Delta = 3$ for all pyramids.

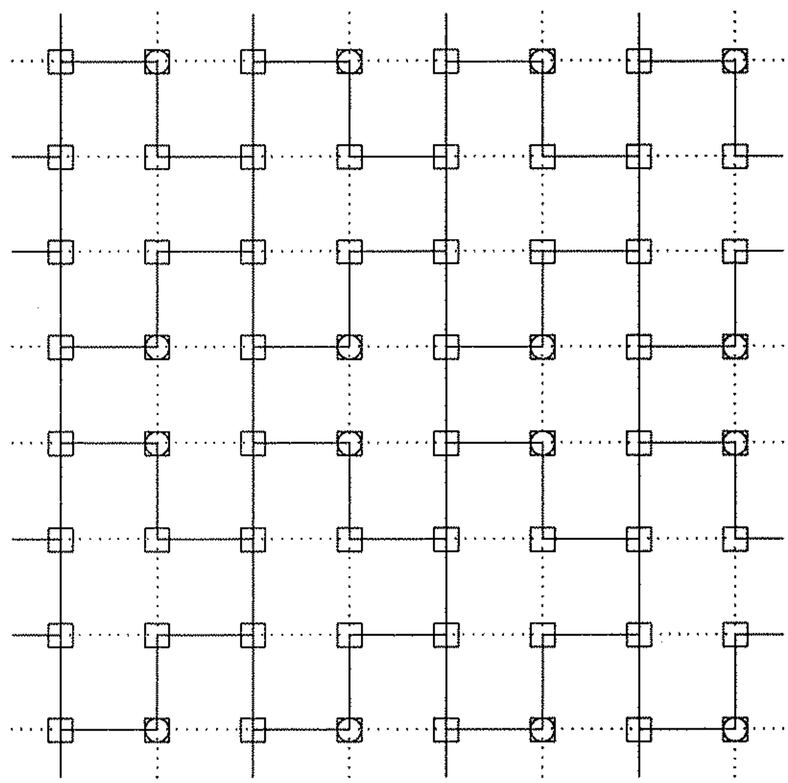


Figure 4.

**Proof**: First we observe that each parent is connected to one of its children. Suppose node $x$ is not connected to any child. Then there must be a neighbor $y$ of $x$ connected to both of its children as shown in Figure 5 (where the other edges incident to $x$ can be varied). In Figure 5 the nodes on $x$'s layer are large circles and the next lower layer is represented by smaller circles. We see that now every path from $y$ must begin on one of the three edges shown and it impossible for $y$ to reach $z$ using at most three edges. Hence every node is connected to one of its children.

Consider a node $a$ which is connected to its parent $b$, such that $b$ is connected to its parent. We note that $a$ cannot be connected to two of its children. If it were then $a$ would have no direct connection to its four *neighbors* (on the same layer). A simple case analysis shows that to find paths to these neighbors would force $b$ to have degree four.

Let $a$ be connected to its child $a'$. Since $a'$ is connected to its own child it only has one other edge $e$ to help $a$ reach its three other children. If $e$ is a sibling edge it can reach two more children; if $e$ is a cross edge it cannot be used to reach any children. Therefore $a$ must use an edge to its neighbor $c$ to reach its remaining child. Now $a$ may be able to reach one of its neighbors through $b$ but it must go through $c$ to get to its remaining two neighbors, but this forces $c$ to have degree 4. □

We show that a 7-spanner with $\Delta = 3$ exists. This leaves open whether for $\Delta = 3$ we can find a spanner with $3 < t < 7$. The above proof is detailed and it would be tedious to try to extend it to larger $t$. We conjecture that no 6-spanner exists with $\Delta = 3$..

**Proposition 5**: There exists a 7-spanner of every pyramid with $\Delta = 3$ and $\gamma = 16$.

**Proof**: The canonical block for such a spanner is shown in Figure 6. There are 16 nodes in the figure that connect to a child each with degree two within the layer. There are 4 nodes that connect to their parent (the large circles), again, each with degree two within the layer. Hence $\Delta = 3$.

Each of the four circled nodes is a "representative" for the $4 \times 4$ subblock containing it. The layer above the layer shown is (necessarily) aligned so that the parent of each representative

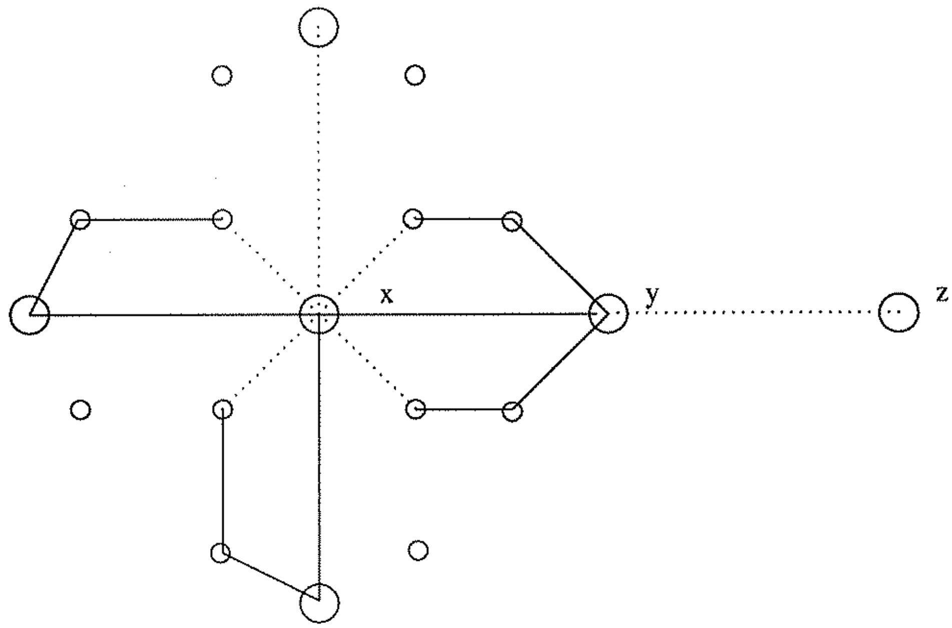



Figure 5.

is a square node. (Note that each square node can reach its siblings in at most two steps.) Every node reaches its parent by routing through the representative's parent (hence $\gamma = 16$). Some nodes take five steps to reach their representative but only require one more step on the next higher layer, while those nodes that need two steps on the next higher layer can reach their representative in at most four steps. In all cases these paths use at most seven edges. Since each missing edge on a layer uses a path of at most five edges, we see $t = 7$. □

**Proposition 6**: It is not possible, for a fixed $t$, to find a $t$-spanner with $\Delta = 2$ for all pyramids.

**Proof**: A spanner with $\Delta = 2$ is a Hamiltonian path or cycle of the pyramid; assume it is a path. Let the nodes be numbered consecutively along the path; it follows that the bandwidth of the pyramid is at most $t$. However it is known [FITZ74] that the bandwidth of an $n \times n$ grid graph is $n$. Since each layer of a pyramid is a grid graph there exists a pyramid (with at least $t+1$ layers) with bandwidth greater than any fixed $t$.

If the Hamiltonian subgraph was instead a cycle then use a numbering beginning at the node on layer 0. Again the bandwidth condition is violated at layer $t+1$. □

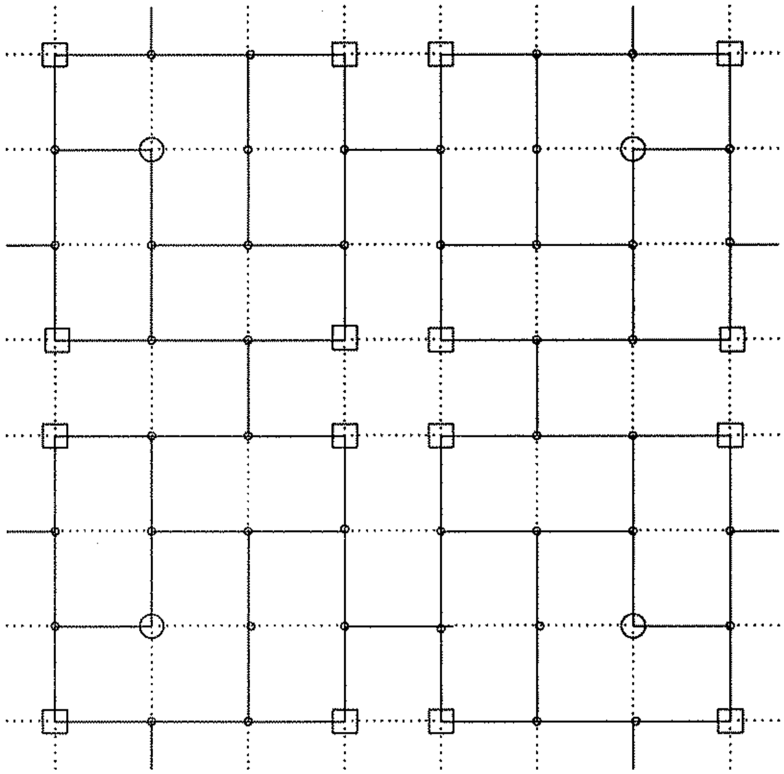


Figure 6.

## 4. Generalizations

In general we could consider the problem of degree-constrained $t$-spanners of arbitrary graphs. We will only mention some well-known processor interconnection schemes.

We first consider a generalization which we call a *full pyramid*. A full pyramid is the same as before except that on a layer each node is connected to its eight nearest neighbors, including its four ''diagonal neighbors''. There is little intrinsic reason for these new edges but they do make certain image processing algorithms more natural to describe.

Obviously including the diagonal edges can at most double the dilation. In practice it seems to have little effect. For example, the construction used above for $t=2$ and $\Delta=6$ is unchanged. The construction used for $t=3$ and $\Delta=4$ before gives $t=4$ for full pyramids; we conjecture that $t=3$ and $\Delta=4$ is impossible for full pyramids. However $t=3$ with $\Delta=5$ is possible as indicated by the canonical block in Figure 7.

For $\Delta = 3$ the construction in Figure 6 almost gives $t = 7$ but the two nodes diagonally adjacent across the center of the cross-shaped region in the center gives $t = 8$. However a different more complex canonical block can achieve $t = 7$; see Figure 8. The proof of correctness is essentially unchanged. The only wrinkle is that the two nodes shown with only one connection on their layer must use paths that leave and re-enter the canonical block.

The two and three dimensional pyramids can and have been extended to $d$ dimensions. The $i$th "layer" of such a pyramid is $(d-1)$-dimensional $2^i \times 2^i \times \cdots \times 2^i$ cubic array and each node has $2^{d-1}$ children. We have not investigated these generalizations.

We have briefly considered cubic arrays by themselves. It is relatively simple to discover for a 3-dimensional (arbitrarily large) cubic array a 5-spanner with $\Delta = 3$ and a 3-spanner with

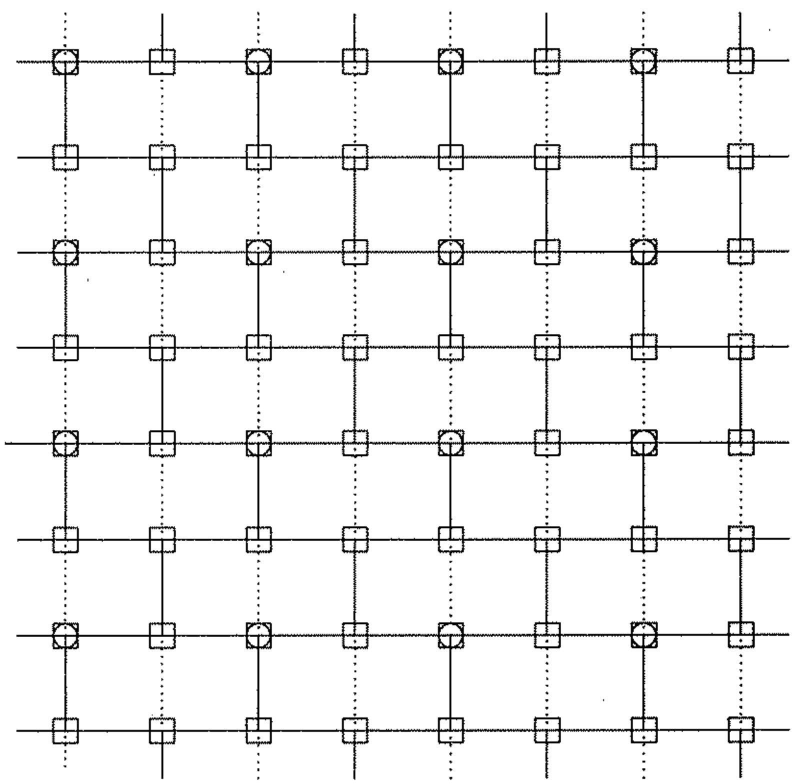


Figure 7.

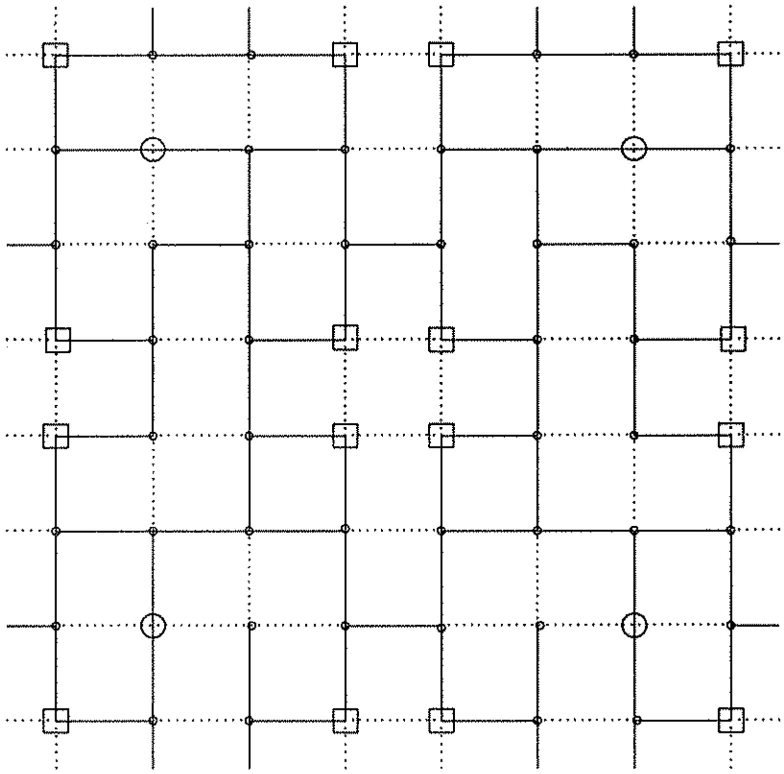


Figure 8.

$\Delta = 4$. These two schemes can be generalized to $d$-dimensional "cubic" arrays as a $(2d-1)$-spanner with $\Delta = 3$ and as a 3-spanner with $\Delta = d+1$, respectively. However it is open whether there is 3-spanner with $\Delta = 3$ for every 3-dimensional cubic array.

Finally we mention that the $d$-dimensional hypercube interconnection network, which is widely used, does have a 3-spanner with *average* degree of 7 [PELE87]. Unfortunately, the known spanner has $\Delta = d$. It is open whether there exists a constant $\Delta$ so that for every $d$ there is a 3-spanner.

## 5. References


[DI 89] V. Di Gesu, An Overview of Pyramid Machines for Image Processing, *Information Sciences*, 47, 1989, pp. 17-34.

[FITZ74] C. H. FitzGerald, Optimal Indexing of the Vertices of Graphs, *Math. of Computation*, **28**, 1974, pp. 825-831.

[LAI90] T. Lai and W. White, Mapping Pyramid Algorithms into Hypercubes, *J. Parallel and Distributed Computing*, **9**, 1990, pp. 42-54.

[PELE87] D. Peleg and J. D. Ullman, An Optimal Synchronizer for the Hypercube, *Proc. 6th Annual ACM Symp. on Principles of Distributed Computing*, 1987, pp. 77-85.

[PELE89] D. Peleg and A. A. Schaffer, Graph Spanners, *J. Graph Theory*, **13**, 1989, pp. 99-116.

[TANI83] S. L. Tanimoto, A Pyramidal Appraoch to Parallel Processing, *Proc. 10th Ann. Symp. on Computer Architecture*, 1983, pp. 372-378.

[UHR86] L. Uhr, Parallel, Hierarchical Software/Hardware Pyramid Architectures, Tech Rept #646, Computer Sciences Dept, Univ. Wisconsin - Madison, June 1986.