# A New Technique for 3-D Domain Decomposition on Multicomputers which reduces Message-Passing

Joseph Gil
The Faculty of Computer Science
Technion – Israel Institute of Technology
Technion City, Haifa 32000, Israel.
yogi@CS.Technion.ac.il

Alan Wagner
Department of Computer Science,
University of British Columbia,
Vancouver, B.C., Canada. V6T 1Z4
wagner@cs.ubc.ca

February 17, 1996

## Abstract

Algorithms for many geometric and physical algorithms rely on a decomposition of 3-D space. The cubical decomposition that is typically used can lead to costly communication overheads when implemented on multicomputers since each cubical cell is adjacent to, and may interact with, as many as 26 neighbouring cells. We explore an alternate decomposition based on truncated octahedra that, along with other advantages, reduces message passing. The cost of implementing the communication structure resulting from this decomposition on low degree regular communication networks is studied. We show that this structure can be embedded with dilation 2 onto a 3-D mesh. A dilation 3 embedding onto a 4-regular graph is also presented.

## 1 Introduction

There are many computationally intensive problems that involve the discretization of two or three dimensional space (e.g., modelling of continuous physical processes, computational geometry). The standard technique used to parallelize these algorithms is to define a regular decomposition of the space and assign the elements in each region of the decomposition to a processor. Typically, each processor executes the same program (i.e., Single Program Multiple Data [3]) and needs to exchange boundary information with the processors assigned to neighbouring regions of space. A cubical decomposition of 3-dimensional space is frequently used for this purpose. In this paper we propose an alternative decomposition that reduces the number of neighbours of a region. This decomposition induces a communication structure that is easier to embed in low degree interconnection networks and, on message passing machines, it substantially reduces the amount of message passing. Moreover, we believe the advantages of this decomposition would also manifest themselves in sequential computation.

The decomposition that we consider is a partitioning of space into identically shaped regions where each region is a *truncated octahedron*. A truncated octahedron is the semi-regular solid with eight hexagonal faces and six rectangular faces (see Figure 1(a)) [2]. Alternatively,

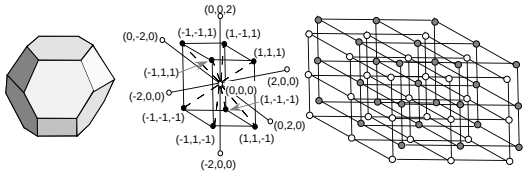


Figure 1: The truncated octahedron, the *bbc* lattice and the woven mesh.

these regions could be viewed as the cells of the Voronoi diagram of the body-centered cubic (*bcc*) lattice. The *bcc lattice* is the set of all points in 3-D space with integer coordinates that are all even or all odd (see Figure 1(b)) [1].

The communication structure of algorithms using this decomposition is defined by the graph whose vertices are the points of the *bcc* lattice and an edge $(u, v)$ whenever the Voronoi cells $u$ and $v$ are adjacent. We will call this graph the *woven mesh* and denote it by $W$. As shown in Figure 1(c), $W$ consists of a pair of interlaced 3-D meshes. The figure does not show the "diagonal" edges connecting an odd point to an even point.

An alternative to the *bcc* and cubic decompositions is the face centered lattice (or *fcc* lattice). The Voronoi diagram of the *fcc* lattice is a rhombic dodecahedron which has 14 quadrilateral faces, 24 edges and 14 vertices. There are two types of vertices, 8 of degree 3 and 6 vertices of degree 4. Although it has fewer neighbours and fewer types of neighbours in comparison to the cube it does not improve upon the number, or types, of neighbours of the truncated octahedron.

A summary of the relevant properties of the *bcc*, $fcc$, and cubic lattice is given in Table 1. As shown by Table 1, the *bcc* minimizes the number and types of interaction between neighbouring cells. Furthermore, it avoids the degenerate types of interaction that occur when two cells meet at an edge or a vertex. Parallel algorithms using the *bcc* lattice thus have fewer communications, and, because there are fewer types of interaction, the implementation of the algorithm may also be simpler. Finally, because it more closely approximates a sphere, it more closely matches the non-directional interactions of geometric and physical continuums such as those for the $N$-body problem.

| Property | *bcc* | *fcc* | *cubic* |
|---|---|---|---|
| No. of neighbours | 14 | 18 | 26 |
| No. vertex neighbours | 0 | 6 | 8 |
| No. edge neighbours | 0 | 0 | 12 |
| Types of neighbours | 2 | 2 | 3 |
| Sphere Approximation | 0.774... | 0.707... | 0.577... |

Table 1: Properties of the *bcc*, *fcc* and *cubic* lattices.

The analogue to the *bcc* decomposition in two dimensions is a hexagonal or beehive partitioning of the plane. The rationale for preferring a hexagonal decomposition over a rectangular one is similar. In comparison to a square the hexagonal cell gives a better approximation to the circle; it has fewer neighbours (six versus eight) which are all of the same type (versus two distinct types). Hexagonal-based interconnection networks have been proposed and algorithms based on hexagonal decompositions exist [4]. To a certain extent, our work is a 3-dimensional generalization.

The source graph for the embeddings discussed in this paper is the woven mesh. We consider finite realizations of the woven mesh constructed from the *bcc* lattice by taking each coordinate modulo an integer, the size of the dimension. To ensure that the wraparound edges are present it is assumed that the dimension size of each coordinate is even. That is, graph $W$ is a pair of interlaced 3-D tori. The host graph in Sections 4 and 5 are also mesh-like where again it is assumed that the wraparound edges are present.

The main question explored in this paper is, given the advantages of the truncated octahedron, are there "good" embeddings of this structure into low degree networks.

# 2 Definitions

The following definition formalizes our previous description of the woven mesh. Let $W = (V, E)$ be the infinite graph defined by

$$V = \{(2l, 2m, 2n) \mid l, m, n \in Z\} \cup \{(2l+1, 2m+1, 2n+1) \mid l, m, n \in Z\}$$
$$E = \{(u, v) \mid u - v \in \{(\pm 2, 0, 0), (0, \pm 2, 0), (0, 0, \pm 2), (\pm 1, \pm 1, \pm 1)\}\}$$

where $u - v$ is the coordinate-wise subtraction of $u$ and $v$.

The coordinates of a node are either all odd or all even. A node is *even* if all its coordinates are even, otherwise the node is *odd*. An edge between two even (odd) nodes is called an *even edge* (*odd edge*). All other edges are called *even-odd*. The set of all even (odd) edges forms the *even mesh* (*odd mesh*). The graph $W$ is regular of degree 14. Each even (odd) node is incident to 6 even (odd) edges and 8 even-odd edges which connect the even and odd meshes (see Figure 1-b).

An *embedding* $\langle \varphi, \rho \rangle$, of a *guest* graph $G$ into a *host* graph $H$ is a one to one mapping, $\varphi : V(G) \longrightarrow V(H)$,

along with a mapping $\rho : E(G) \longrightarrow \{\text{paths in } H\}$, of edges in $G$ to paths in $H$ [5]. The *dilation* of an edge $e \in E(G)$ is the length of path $\rho(e)$ in $H$. The *dilation* of an embedding is the maximum dilation over all edges. The *congestion* of an edge $e \in E(H)$ is the number of edges in $G$ whose image under $\rho$ contains $e$. The *congestion* of an embedding is the maximum congestion over all edges. In calculating the congestion of an embedding, we assume that simultaneous bidirectional communication is possible and thus we give the congestion of an arc rather than the edge.

# 3 Assigning points to cells

In order to use the *bcc* decomposition it is necessary to determine, for any given point in space, the Voronoi cell to which it belongs. The following algorithm computes, for an arbitrary point $x \in R^3$, the node closest to it in the *bcc* lattice.

1. **Input**: $(x_1, x_2, x_3)$, a point in 3-D space.
2. **Output**: a node in the *bcc* lattice nearest to it.
3. $(n_1, n_2, n_3) \leftarrow 2\,\text{round}((x_1, x_2, x_3)/2)$
4. **If** $|x_1 - n_1| + |x_2 - n_2| + |x_3 - n_3| \leq 1.5$
5. **then** return $(n_1, n_2, n_3)$
6. **else** return $(m_1, m_2, m_3)$
   $\leftarrow 2\,\text{round}((x_1 + 1, x_2 + 1, x_3 + 1)/2) - (1, 1, 1)$

This algorithm is a result of the following derivation. Consider a point $(x_1, x_2, x_3)$. For $i = 1, 2, 3$, let $n_i$ be the closest even node to $x_i$ and let $m_i$ be the closest odd node to $x_i$. In addition, let $e_i = |n_i - x_i|$ and $o_i = |m_i - x_i|$. Since $|n_i - m_i| = 1$ and $x_i$ lies between $n_i$ and $m_i$, $o_i + e_i = 1$.

The $n_i$'s are calculated in Step 3 of the algorithm and the $m_i$'s, when needed, are calculated in Step 6. Let $x_i = y_i + \delta, 0 \leq \delta < 1$ and $y_i \in Z$. In Step 3, if $y_i$ is even then $y_i$ is the closest even point to $x_i$, and, as required, we have that $n_i = 2\,\text{round}(x_i/2) = y_i + 2\,\text{round}(\delta/2) = y_i$. If $y_i$ is odd then $y_i + 1$ is the closest even point to $x_i$, and $n_i = 2\,\text{round}(x_i/2) = y_i - 1 + 2\,\text{round}((1 + \delta)/2) = y_i + 1$. Similarly in Step 6, if $y_i$ is even then $y_i + 1$ is the closest odd point to $x_i$, and $m_i = 2\,\text{round}((y_i + \delta + 1)/2) - 1 = y_i + 2\,\text{round}((1 + \delta)/2) - 1 = y_i + 1$. If $y_i$ is odd then $y_i$ is the closest odd point to $x_i$, and $m_i = y_i + 1 + 2\,\text{round}(\delta/2) - 1 = y_i$. In summary, the nearest even node to $(x_1, x_2, x_3)$ is $(n_1, n_2, n_3)$, as calculated in Step 3, and the nearest odd node is $(m_1, m_2, m_3)$, as calculated in Step 6.

Now, between $(n_1, n_2, n_3)$ and $(m_1, m_2, m_3)$, $(n_1, n_2, n_3)$ is nearest to $(x_1, x_2, x_3)$ when

$$\begin{aligned} \sum_{1 \leq i \leq 3} (m_i - x_i)^2 - \sum_{1 \leq i \leq 3} (n_i - x_i)^2 &\geq 0 \\ \sum_{1 \leq i \leq 3} (o_i - e_i)(o_i + e_i) &\geq 0 \end{aligned}$$

by substituting $1 - e_i$ for $o_i$ we have $\sum_{1 \leq i \leq 3} e_i \leq 1.5$ which is the test used in Step 4 to return the closest even or odd node of the *bcc* lattice.

# 4 Embedding the woven mesh onto the 3-D mesh

In this section we describe a dilation 2 embedding of the woven mesh into the 3-D mesh. Formally, the mapping is

$$\varphi((x, y, z)) = \begin{cases} ((x - y)/2, (x + y)/2, (z - 1)/2) \\ \qquad\qquad (x, y, z) \text{ odd} \\ ((x - y)/2, (x + y)/2, z/2) \\ \qquad\qquad \text{otherwise} \end{cases}$$

From the definition of $\varphi$ it follows that

$$\varphi^{-1}((x, y, z)) = \begin{cases} (x + y, y - x, 2z) & x + y \text{ even} \\ (x + y, y - x, 2z + 1) & x + y \text{ odd.} \end{cases}$$

The notion of node parity in $W$ can be extended to the nodes of the 3-D mesh. The parity of a node in the 3-D mesh is the parity of its pre-image under $\varphi$ which is equivalent, for a node $(x, y, z)$ in the mesh, to the parity of $x + y$.

Now, with respect to $\varphi$, we define $\rho$ a mapping of the edges of $W$ to paths in the 3-D mesh. From the definition of $W$, it follows that the edges of $W$ can be classified according to the vector $v - u \in \{(\pm 2, 0, 0), (0, \pm 2, 0), (0, 0, \pm 2), (\pm 1, \pm 1, \pm 1)\}$. The edges in the 3-D mesh can be classified by the unit vectors $\{X^+, X^-, Y^+, Y^-, Z^+, Z^-\}$ where, for example, $X^+$ denotes vector $(1, 0, 0)$ and edge $((2, 3, 1), (2, 3, 2))$ is a $Z^+$ type edge.

We define $\rho$ by giving, for each type of edge $v - u$ in $W$, the path from $\varphi(u)$ to $\varphi(v)$ in the 3-D mesh.

This path is specified by considering the difference vector $\varphi(v) - \varphi(u)$, which gives the distance and direction in which $\varphi(v)$ differs from $\varphi(u)$, and listing the arcs in the path by the vectors $\{X^+, X^-, Y^+, Y^-, Z^+, Z^-\}$.

The complete mapping $\langle \varphi, \rho \rangle$ is given by Table 2. The last two columns in each table specify the path in

| $v-u$ | $\varphi(v)$ | Path | | $\varphi(v)$ | Path($v$ even, $u$ odd) | |
|---|---|---|---|---|---|---|
| | $-\varphi(u)$ | 1st | 2nd | $-\varphi(u)$ | 1st | 2nd |
| (1,1,1) | (0,1,0) | $Y^+$ | | (0,1,1) | $Z^+$ | $Y^+$ |
| (1,-1,1) | (1,0,0) | $X^+$ | | (1,0,1) | $X^+$ | $Z^+$ |
| (-1,1,1) | (-1,0,0) | $X^-$ | | (-1,0,1) | $X^-$ | $Z^+$ |
| (-1,-1,1) | (0,-1,0) | $Y^-$ | | (0,-1,1) | $Z^+$ | $Y^-$ |
| (1,1,-1) | (0,1,-1) | $Z^-$ | $Y^+$ | (0,1,0) | $Y^+$ | |
| (-1,1,-1) | (-1,0,-1) | $X^-$ | $Z^-$ | (-1,0,0) | $X^-$ | |
| (1,-1,-1) | (1,0,-1) | $X^+$ | $Z^-$ | (1,0,0) | $X^+$ | |
| (-1,-1,-1) | (0,-1,-1) | $Z^-$ | $Y^-$ | (0,-1,0) | $Y^-$ | |
| (2,0,0) | (1,1,0) | $X^+$ | $Y^+$ | | | |
| (0,2,0) | (-1,1,0) | $X^-$ | $Y^+$ | | | |
| (0,0,2) | (0,0,1) | $Z^+$ | | | | |
| (-2,0,0) | (-1,-1,0) | $X^-$ | $Y^-$ | | | |
| (0,-2,0) | (1,-1,0) | $X^+$ | $Y^-$ | | | |
| (0,0,-2) | (0,0,-1) | $Z^-$ | | | | |

Table 2: The mapping $\rho$ of the edges of $W$ into the 3-D mesh.

terms of the unit vectors where each unit vector is one step along the route from $\varphi(u)$ to $\varphi(v)$. For example, edge $\{(1,1,1),(2,2,2)\}$ with difference vector $(1,1,1)$ is mapped to nodes $(0,1,0)$ and $(0,2,1)$, respectively, with difference vector $(0,2,1) - (0,1,0) = (0,1,1)$ along the path $(0,1,0) \xrightarrow{Z^+} (0,1,1) \xrightarrow{Y^+} (0,2,1)$.

**Theorem 1** *There is a dilation 2, arc congestion 4 embedding of $W$ in the 3-D mesh.*

***Proof.*** By inspection of Table 2, the dilation of the embedding is two. Consider the congestion of the embedding.

First we show that all edges of the same type and parity have equal congestion. Let $e = (x, y)$ and $e' = (x', y')$ be two arcs in the 3-D mesh of the same type. Two arcs $(x, y)$ and $(x', y')$ are of the same type when they belong to the same unit vector (i.e., $y - x = y' - x'$). Furthermore, suppose the parity of $x$ equals that of $x'$. We will show that for every path in $\rho$ using $e$ there is a corresponding path in $\rho$ using $e'$.

Consider $e$ and $e'$, and suppose that $e$ is in $\rho((u, v))$ for some edge $(u, v)$ in $W$. Define the translation $f(w) = w + (x' - x)$. Since the parity of $x'$ equals the parity of $x$ it follows that $(x' - x)$ is a vector $(i, j, k)$ for which $i + j$ is even. Thus $f$ preserves parity in addition to distance and direction. It follows that the parity and type of $f(\varphi(u))$ and $f(\varphi(v))$ equals that of $\varphi(u)$ and $\varphi(v)$. Let $u' = \varphi^{-1}(f(\varphi(u)))$ and $v' = \varphi^{-1}(f(\varphi(v)))$. Since the type and parity of $f(\varphi(u))$ and $f(\varphi(v))$ are the same as $\varphi(u)$ and $\varphi(v)$, $(u', v')$ is an edge in $W$ of the same type and parity as that of $(u, v)$. Therefore, in Table 2, $(u, v)$ and $(u', v')$ in $W$ use the same unit vectors, in particular $e'$ is in $\rho((u', v'))$. Since $\varphi$ and $f$ are one to one functions, it follows that $e$ is in the image of exactly the same number of edges in $W$ as $e'$. That is the congestion of $e$ equals the congestion of $e'$.

Since the congestion on all arcs of the same type and parity is equal, it suffices to determine the congestion on one arc of each type and parity. Let $e = (x, y)$ be an arc of type $t$. If $x$ is an even (odd) node then $e$ either appears in the first step of paths originating from $x$ or in the second step of paths originating from an odd (even) node adjacent to $x$. In total, the number of paths in which $x$ appears is given by the number of times $t$, taking into account parity, appears in the 1st step of Table 2 plus the number of times $t$, again taking into account parity, appears in the 2nd step of Table 2. On examination of Table 2 we find that the congestion on each arc is at most 4.

Therefore, there is a dilation 2, arc congestion 4 mapping of $W$ into the 3-D mesh. ∎

**Corollary 2** *Embedding $\langle \varphi, \rho \rangle$ of $W$ in the 3-D mesh has minimum dilation and arc congestion.*

***Proof.*** There is not a dilation 1 embedding of $W$ in the 3-D mesh since $W$ is regular of degree 14 and the mesh is regular of degree 6. It follows that at least 8 edges must be dilated and that every node of $W$, when mapped to the 3-D mesh, uses at least 22 arcs of the mesh. This implies that the average congestion of any embedding in a 6-regular graph is at least $22/6 = 3\frac{2}{3}$. Since congestion is integral, there must exist a congestion 4 arc. By Theorem 1, our mapping is optimal. ∎

These results naturally extend to hypercubes since 3-D meshes of the appropriate dimension are subgraphs of the hypercube. One should also compare this to the cubic lattice where the best embedding has dilation 3 and congestion 9.

# 5 Dilation 3 embedding in a 4 regular graph

Consider the embedding of $W$ in the following 4-regular graph $H$. The graph $H$ will be defined on the same set of points as the 3-D mesh and we will use mapping $\varphi$ from Section 4 to map nodes of $W$ to the nodes of $H$. A portion of $H$ about points $(0,0,0)$ and $(0,1,0)$ is shown in Figure 2. Graph $H$, like $W$, is mesh-like and the mapping from $W$ to $H$ produces a checkerboard-like pattern of alternating odd and even nodes.

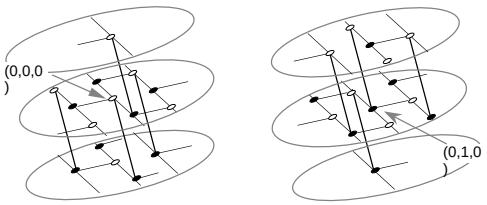


Figure 2: Neighbourhood of an even and odd node in $H$.

Formally, graph $H$ is defined as follows. The edges of $H$ can be classified as either $X$-, $Y$- or $YZ$-dimension edges. There is an edge $((u_1,u_2,u_3),(v_1,v_2,v_3))$ in $H$ whenever one of the following conditions is met:

**$X$-dimension edges:** $u_1+u_2$ has odd parity, $v_1 = u_1+1$, $v_2 = u_2$ and $v_3 = u_3$;

**$Y$-dimension edges:** $v_1 = u_1$, $v_2 = u_2+1$ and $v_3 = u_3$;

**$YZ$-dimension edges:** $u_1+u_2$ has odd parity, $v_1 = u_1+1$, $v_2 = u_2$, and $v_3 = u_3-1$.

Graph $H$ contains all $Y$ dimension mesh edges. It contains those $X$ dimension mesh edges connecting an odd node (i.e., a node $(u_1,u_2,u_3)$ where $u_1+u_2$ is odd) to an even node (i.e., a node $(u_1,u_2,u_3)$ where $u_1+u_2$ is even). That is, all odd $(i,j,k)$ nodes are connected to $(i+1,j,k)$. It contains edges connecting an odd node on one level to the even node one step away from it in the level below (i.e., all odd $(i,j,k)$ nodes are connected to $(i+1,j,k-1)$ in level $k-1$). Note that every node in $H$ is connected to two nodes in the $Y$ dimension, one node in the $X$ dimension, and one node in the $Z$ dimension.

Function $\rho$, the mapping of edges in $W$ to paths in $H$ is given in Table 3. As in Section 4, Table 3 lists the path in $H$ for an edge in $W$ with respect to each difference vector. The routing is not as regular as in Table 2. In particular, the routing in $H$ between odd nodes is the opposite to that between even nodes. As in the proof of The-

| $v-u$ | $\varphi(v)$ | Path | | |
|---|---|---|---|---|
| | $-\varphi(u)$ | 1st | 2nd | 3rd |
| (1,1,1) | (0,1,0) | $Y^+$ | | |
| (1,-1,1) | (1,0,0) | $Y^+$ | $X^+$ | $Y^-$ |
| (-1,1,1) | (-1,0,0) | $X^-$ | | |
| (-1,-1,1) | (0,-1,0) | $Y^-$ | | |
| (1,1,-1) | (0,1,-1) | $Y^-Z^-$ | $Y^+$ | $Y^+$ |
| (-1,1,-1) | (-1,0,-1) | $X^-$ | $Y^+$ | $Y^-Z^-$ |
| (1,-1,-1) | (1,0,-1) | $Y^-Z^-$ | $X^+$ | $Y^+$ |
| (-1,-1,-1) | (0,-1,-1) | $Y^-Z^-$ | | |

| $v-u$ | $\varphi(v)$ | Path ($v$ even, $u$ odd) | | |
|---|---|---|---|---|
| | $-\varphi(u)$ | 1st | 2nd | 3rd |
| (1,1,1) | (0,1,1) | $Y^+Z^+$ | | |
| (1,-1,1) | (1,0,1) | $X^+$ | $Y^-$ | $Y^+Z^+$ |
| (-1,1,1) | (-1,0,1) | $Y^+Z^+$ | $X^-$ | $Y^-$ |
| (-1,-1,1) | (0,-1,1) | $Y^+Z^+$ | $Y^-$ | $Y^-$ |
| (1,1,-1) | (0,1,0) | $Y^+$ | | |
| (-1,1,-1) | (-1,0,0) | $Y^-$ | $X^-$ | $Y^+$ |
| (1,-1,-1) | (1,0,0) | $X^+$ | | |
| (-1,-1,-1) | (0,-1,0) | $Y^-$ | | |

| $v-u$ | $\varphi(v)$ | Path ($u$ even) | | Path ($u$ odd) | |
|---|---|---|---|---|---|
| | $-\varphi(u)$ | 1st | 2nd | 1st | 2nd |
| (2,0,0) | (1,1,0) | $Y^+$ | $X^+$ | $X^+$ | $Y^+$ |
| (0,2,0) | (-1,1,0) | $X^-$ | $Y^+$ | $Y^+$ | $X^-$ |
| (0,0,2) | (0,0,1) | $Y^-$ | $Y^+Z^+$ | $Y^+Z^+$ | $Y^-$ |
| (-2,0,0) | (-1,-1,0) | $X^-$ | $Y^-$ | $Y^-$ | $X^-$ |
| (0,-2,0) | (1,-1,0) | $Y^-$ | $X^+$ | $X^+$ | $Y^-$ |
| (0,0,-2) | (0,0,-1) | $Y^-Z^-$ | $Y^+$ | $Y^+$ | $Y^-Z^-$ |

Table 3: Mapping of the edges of $W$ into $H$.

orem 1, arc congestion is determined by first partitioning the edges in $H$ into a set of vectors (all unit except for two) of the form $\{X^+, X^-, Y^+, Y^-, Y^+Z^+, Y^-Z^-\}$. Note the cross-edges, $Y^+Z^+$ and $Y^-Z^-$.

**Theorem 3** *There is a dilation 3, congestion 8 embedding of $W$ in $H$.*

***Proof.*** By inspection of Table 3, the dilation of the mapping is 3. Consider the congestion of the embedding.

As in Theorem 1, as long as two arcs are of the same type and parity there is a translation $f$ such that whenever an edge of $W$ is in the image of one arc there is a corresponding edge of $W$ in the image of the second. This relies on the fact that $\varphi$ has remained the same as in Theorem 1 and that, as in Theorem 1, $f$ preserves parity in addition to distance and direction. Therefore, arcs of the same type and parity have the same congestion. In turn this implies that the congestion of an arc equals the number of times, with respect to paths originating at both odd and even nodes, an arc of a specific type is used. In this case, arcs are of the following type: $X^+, X^-, Y^+, Y^-, Y^+Z^+, Y^-Z^-$. By counting the occurences of arcs of each type in Table 3 we find that the congestion on each arc is at most 8.

Therefore there is a dilation 3 arc congestion 8 mapping of $W$ into $H$. ∎

# 6 Conclusions and open problems

We have seen that the regular partitioning of three dimensional space into truncated octahedrons has several advantages over the more traditional cubic decomposition. This decomposition has the potential of significantly improving the performance of algorithms for geometric and physical problems whose solution involves decomposition or discretization of space. However, the more complex structure of the truncated octahedron cells and the less regular structure of the underlying *bcc* lattice makes it necessary to develop tools and techniques for using this newly proposed decomposition. To that end, we gave a simple and efficient algorithm to determine, for a given point in space, the cell it resides in.

We continued our investigation by exploring some aspects of using this decomposition on a network of multicomputers where we believe it may reduce the communication overhead. We gave optimal embeddings of the *bcc* lattice on a 4-regular graph, suitable for transputer networks and on the 3-D mesh, and consequently on the hypercube. We have also proven that there does not exist a dilation 2 embedding of the woven mesh on any 4-regular graph. However, due to space limitations, the proof has been omitted.

What is the best embedding of the woven mesh on a 3-regular graph? Clearly, the dilation must be at least 3, but is this realizable? Similarly, it would be of interest to find a dilation 2 embedding of the woven mesh on a 5-regular graph. Finally, a related problem is that of embedding the 26-regular graph induced by the cubic decomposition with dilation 3 on a 4-regular graph.

It would also be interesting to explore other collections of space-filling polytops to attempt to further reduce the communication overhead. Peter Shor and David Epstein (private communication) suggested a set of points whose space-tiling Voronoi cells have an interesting property. There are two types of Voronoi cells: one type has 12 neighbours (making up a quarter of the cells) while the other type has 14 neighbours (making up three quarters of the cells), so the average number of neighbours is 13.5. Determining the minimum average number of neighbours in a tiling is certainly an interesting question.

# References


[1] J. Conway and N. Sloane. *Sphere Packings, Lattices and Groups*. Springer-Verlag, Grundlehren der mathematischen Wissenchaften 290, New York, 1988.

[2] H. Coxeter. *Regular Polytopes*. Pitman Publishing Co., New York, Chicago, 1947.

[3] G. Fox, M. Johnson, G. Lyzenga, S.Otto, J. Salmon, and D. Walker. *Solving Problems on Concurrent Processors*. Prentice Hall, Englewood Cliffs, New Jersey, 1988.

[4] M. Nakagawa, T. Mori, and M. Sasaki. Hexagonal lattice geometry for monte carlo calculations. *Ann. Nuclear Energy*, 18(8):467–477, 1991.

[5] A. L. Rosenberg. Issues in the study of graph embeddings. In H. Noltemeier, editor, *Graph-Theoretic Concepts in Computer Science (Proc. of the Int. Workshop WG80)*, pages 151–176. Springer-Verlag, New York, 1981. Lecture Notes in Computer Science 100.