Efficient Structural Outlooks for Vertex Product Networks

Ahmad M. Awwad

Dissertation Submitted for the Degree of Doctor of Philosophy
to the Faculty of Computer Science, Mathematics and Statistics
University of Glasgow

ABSTRACT

In this thesis, a new classification for a large set of interconnection networks, referred to as "Vertex Product Networks" (VPN), is provided and a number of related issues are discussed including the design and evaluation of efficient structural outlooks for algorithm development on this class of networks. The importance of studying the VPN can be attributed to the following two main reasons: first an unlimited number of new networks can be defined under the umbrella of the VPN, and second some known networks can be studied and analysed more deeply. Examples of the VPN include the newly proposed *arrangement-star* and the existing Optical Transpose Interconnection Systems (OTIS-networks).

Over the past two decades many interconnection networks have been proposed in the literature, including the star, hyperstar, hypercube, arrangement, and OTIS-networks. Most existing research on these networks has focused on analysing their topological properties. Consequently, there has been relatively little work devoted to designing efficient parallel algorithms for important parallel applications. In an attempt to fill this gap, this research aims to propose efficient structural outlooks for algorithm development. These structural outlooks are based on grid and pipeline views as popular structures that support a vast body of applications that are encountered in many areas of science and engineering, including matrix computation, divide-and-conquer type of algorithms, sorting, and Fourier transforms. The proposed structural outlooks are applied to the VPN, notably the arrangement-star and OTIS-networks.

In this research, we argue that the proposed arrangement-star is a viable candidate as an underlying topology for future high-speed parallel computers. Not only does the arrangement-star bring a solution to the scalability limitations from which the existing star graph suffers, but it also enables the development of parallel algorithms based on the proposed structural outlooks, such as matrix computation, linear algebra, divide-and-conquer algorithms, sorting, and Fourier transforms. Results from a performance study conducted in this thesis reveal that the proposed arrangement-star supports efficiently applications based on the grid or pipeline structural outlooks.

OTIS-networks are another example of the VPN. This type of networks has the important advantage of combining both optical and electronic interconnect technology. A number of studies have recently explored the topological properties of OTIS-networks. Although there has been some work on designing parallel algorithms for image processing and sorting, hardly any work has considered the suitability of these networks for an important class of scientific problems such as matrix computation, sorting, and Fourier transforms. In this study, we present and evaluate two structural outlooks for algorithm development on OTIS-networks. The proposed structural outlooks are general in the sense that no specific factor network or problem domain is assumed. Timing models for measuring the performance of the proposed structural outlooks are provided. Through these models, the performance of various algorithms on OTIS-networks are evaluated and compared with their counterparts on conventional electronic interconnection systems. The obtained results reveal that OTIS-networks are an attractive candidate for future parallel computers due to their superior performance characteristics over networks using traditional electronic interconnects.

Dedication

I have a strong believe that this work would not have been done without the help of Allah (SWT) and memory of my father and my mother. My only regret is that they are not with us to see that one of their wishes has come true.

This thesis is dedicated to my father and mother whose encouragement, during their lifetime and continuous reminder, after their death enabled me to continue working very hard during my research work.

Acknowledgement

I would like to express my grateful thanks to Dr. Mohamed Ould-Khaoua and Dr. Abdel-Elah Al-Ayyoub for their great help during my research work. They both provided me with continuous guidance, encouragement, and valuable feedback since the early stages of this thesis.

Also I would like to express my thanks to my external and internal examiners Professor Mohammed S. Obaidat and Dr. Lewis M. Mackenzie for their rich comments which, helped allot in improving the final format of the thesis.

I am particularly thankful to my wife and sons for their unlimited love, moral support and patience during my research work. I would also like deeply to thank my brothers, sisters, and nephews for their support and encouragement.

Finally, I am very grateful to the Centre for British Teachers for their financial support, especially Mr. Michel Davis.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction

The solution of many large-scale problems that require enormous computational power is now possible due to the availability of high-speed parallel computers employing hundreds or thousands of processors. When these machines were first proposed, two observations were made [23]. First, it was observed that in order to get the maximum possible benefit from working with multiple processors as opposed to a single processor, it was important to partition a task as evenly as possible among the processors. This was important for preventing the parallel computer from being dependent on one single processor. Second, it was observed that in order to complete a given computation, a processor could require access to the result of computation of other processors. This is particularly the case with multiple-instruction-multiple-data streams (MIMD) parallel computers, where different processors execute different sequence of tasks [39]. In fact, a significant amount of the time required for a parallel computer to solve a given problem is spent by one or more processors communicating the results of their computations to other processors. These processors are physically interconnected by an interconnection network, and co-ordinate their activities to solve a common problem by exchanging information, depending on the way communication is achieved between the processors. Figure 1.1 shows set of processors connected to each other through an interconnection network.

A major design issue involved in the construction of a parallel computer is the topology of the network interconnecting the processors as this has great impact on system performance. Many interconnection networks have been proposed over the past two decades [3, 8, 25, 27, 28]. The most popular examples include the hypercube [2], star [3], arrangement [28], Optical transpose Interconnection Systems (OTIS-networks) [20, 34], and many others [5, 26, 27].

1.2 Design Parameters for Interconnection Networks

The topology of most existing networks can be modelled as unidirectional graphs, where each processor corresponds to a vertex and each communication link to an edge. These networks are sometimes referred to as *direct networks* because the interconnection models between the processors are direct. The proposed design of the interconnection network determines many of the architectural features of a parallel computer and affects the overall system performance. Although the actual performance of a network depends on many technological and implementation factors [56, 57] several metrics have been extensively used in the literature to evaluate and compare different topologies including degree, diameter, scalability, number of links, implementation cost and ability for efficient algorithm development [3, 4, 11, 21, 28]. The most important of these considerations are listed below with brief descriptions. A comprehensive evaluation of a candidate topology must take into account all these characteristics and the choice of an appropriate topology depends on various trade-offs between them [2, 4, 7, 22, 25, 28, 32, 33].

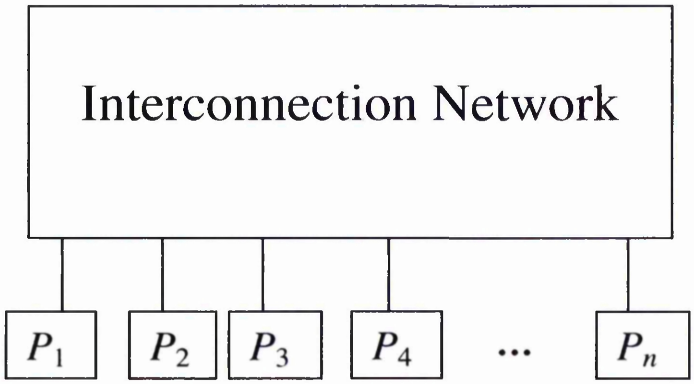


Figure 1.1: Set of processors connected through an interconnection network.

Degree:

The degree of a network is the number of neighbours of a processor, also called a node, (these two terms will be used interchangeably throughout the thesis). The degree relates to the port capacity of the processors and hence to the hardware cost of the network [3]. If each node is connected to a fixed number of neighbours, then the network topology is a regular graph.

Diameter:

The maximum distance between a pair of nodes in a network is referred to as the diameter of the network [28]. The diameter provides a rough estimate of the communication delay experienced by a message when crossing from source to destination.

Symmetry:

The simplest notion of *symmetry* relates to the vertex and edge symmetry. A vertex symmetric graph looks the same when viewed through any vertex. In graph-theoretic terms, a network is node-symmetric if, for every pair of nodes *a* and *b*, an automorphism of the network can be found that maps *a* into *b*. Also a network is edge-symmetric if, for every two edges *a* and *b*, an automorphism of the network can be found that maps *a* into *b* [3].

One important consequence of the vertex symmetry is that a guest structure embedded in one region of the host network can be readily translated into another region without affecting the quality of the original embedding [30, 42]. Symmetry in a network also allows nodes to use the same routing algorithm and simplifies the task of path selection [43]. Furthermore, common data movement operations, such as broadcast, can be implemented easily and efficiently on symmetric topologies.

Expandability/scalability:

Expandability relates to the possibility of building networks based on the same topology for different size requirements. An expandable network allows expansion of a parallel computer without having to replace the existing hardware [25]. A network is scalable if it continues to yield the same performance per processor as the number of processors increases.

Hierarchical structure:

This relates to the ability to build large networks from smaller sub-networks [25]. Recursively-defined networks naturally possess certain symmetries that are often exploited in the design of routing algorithms and in the evaluation of fault tolerance [43].

Node disjoint paths:

The existence and characterisation of parallel paths is crucial to the design of routing algorithms [52]. Furthermore, the existence of parallel paths is necessary to speed up transfer of large amounts of data and provide alternative routes in situations of processor failure [43].

Optimal algorithms for various modes of communication:

Algorithm designers have identified the need for different modes for parallel communication in a network, such as broadcasting communication [13]. It is important that the underlying network embeds adequate communication graphs leading to optimal algorithms for the various modes of communication. For example, the broadcast mode of communication often requires the use of the grid structural outlook where the network nodes are decomposed into subgraphs arranged as rows and columns of a grid in order to facilitate the broadcasting operation [53].

Cost/performance criteria:

A widely accepted criterion for cost-performance trade-off assessment is the cost/performance criteria [6]. This metric is defined as product of the number of links with another metric such as the degree or diameter of the network. The area of the VLSI layout needed to implement a network can also be used to estimate cost of an interconnection network.

Connectivity:

The connectivity of a network provides a measure of a number of independent paths connecting a pair of nodes [14]. The term node connectivity is defined as the minimum number of nodes that need to be removed to disconnect the network, and link connectivity is defined as the minimum number of links that has to be removed to disconnect the network.

Flexibility:

The flexibility of a network topology should be rich enough to allow frequently used topologies to be embedded so that algorithms designed for other architectures can be simulated.

Embedding:

The problem of assigning processes to processors in a parallel computer is also known as the mapping problem. A parallel algorithm could be represented as a guest network *G*, where the nodes in the guest network express processes and links express communication between processes. A host network *H*, where nodes express processors and links express communication between processors, represents a parallel computer. The mapping problem could be formed as a network embedding problem, mapping statically known networks, or guest network, into a network with fixed number of nodes [25, 30].

The need for embedding stems from the fact that if a network *G* can be embedded in a network *H*, then all the algorithms developed for one type of parallel systems with network *G* can be easily transported onto another systems based on the network *H* without any extra cost [12].

Dilation:

The *dilation* of an embedding is the length of the maximum path in the host network *H* that is associated with a link in the guest network *G*.

Expansion:

The *expansion* of an embedding is defined as the ratio $|V_H|/|V_G|$ where $|V_H|$ and $|V_G|$ represent the number of node in the networks H and G, respectively.

1.3 The Star Graph

During the last decade a large variety of interconnection networks for parallel computers have been investigated [1, 4, 6, 26, 27]. One of the well-known interconnection networks is the hypercube, also known as the binary n-cube. Figure 1.2 presents an example of a 4-dimensional hypercube [8]. The *star graph* [3] is another example, which has been proposed as an attractive alternative to the hypercube. This network has attracted a lot of research efforts. For instance, several properties of this network have been studied including its basic topological properties [2], parallel path characterisation [2], and embedding [29]. Akers and Krishnamurthy [3] have shown that the star graph has several advantages over the hypercube including a smaller diameter, smaller average diameter and lower degree for a fixed network size. The star graph is also edge and vertex symmetric and is maximally fault tolerant [3]. Furthermore, a limited number of parallel algorithms for some well-known problems have been reported in the literature including computing fast Fourier transforms [44], broadcasting [45, 46, 47], selection and sorting [48].

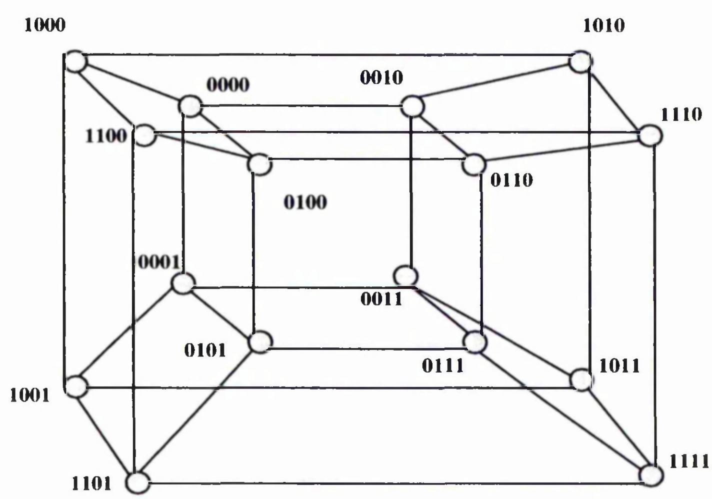


Figure 1.2: 4 -Dimensional hypercube.

Definition 1.1: The n-star graph, denoted by S_n, has $n!$ nodes each labelled with a unique permutation on $\langle n \rangle = \{1,...,n\}$. Any two nodes are connected if, and only if, their corresponding permutations differ exactly in the first and any other position.

Figure 1.3 shows the 4-star graph with 4 groups each containing 6 vertices (i.e. four copies of 3-star graphs). The diameter, δ, and the degree, α, of the star graph are as follows [3]:

δ, of n-star graph $= \lfloor \frac{3}{2}(n-1) \rfloor$

α, of the n-star graph = n-1, where n>1.

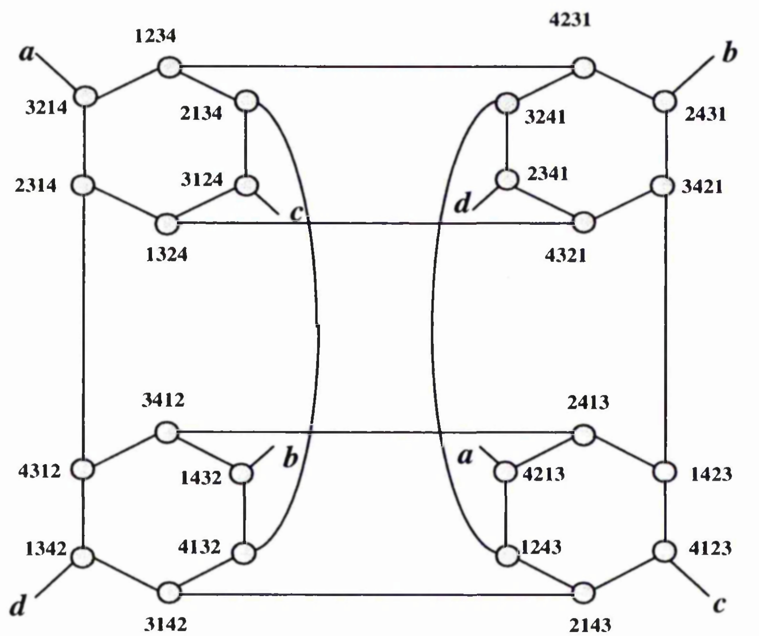


Figure 1.3: The 4-star graph, S_4.

The structural outlook can be issued by decomposing the nodes of the graph into subgraphs, this will facilitate data broadcasting between the different nodes of the network. As an instance of the structural outlooks are the well known grid and pipeline structures, where the obtained subgraphs are assigned on the rows and columns of the grid respectively, while for the pipeline structure these subgraphs will form the different stages of the structure. Actually there are two structural outlooks for the star graph, [10, 46]. Menn and Somani [10] have shown in their paper that the star graph can be viewed as $n \times (n\text{-}1)!$, where the rows and the columns in this grid are (n-1)-star and an n-linear array respectively. Furthermore, Ferreria and Berthome [46] have shown that the star graph can be viewed as a rectangular grid $R\times C$ (Rows by Columns) where the rows are substar-$S_{n\text{-}2}$ and the columns are $n(n\text{-}1)$ nodes on each column. However, these structural outlooks are insufficient for developing efficient algorithms on the star graph. For the star graph, the grid structural outlooks are unbalanced because they resemble column processing rather than grid processing. For instance, the 10-star can be viewed as 10×362880 or 90×40320 grid. Compared to the optimal square grid structure of the hypercube, this grid is inefficient because it generates unbalanced message dimension between the rows and columns; it follows that the communication cost between the rows and columns of the grid are not uniform.

The pipeline structure is another well-known structural outlook that is suitable for real applications. In the pipeline structure the issued stages from the decomposition of the nodes into different n stages is represented by ${}^{i}S_{n\text{-}1}$ for each different stage, where $1\leq i\leq n$. We know from the literature [18, 19] that the star graph can be viewed as a pipelined structure, where the star pipelined outlook arrange the n-star as a sequence of $n\times(n\text{-}1)$-stars forming an n-stage pipeline. The authors in [18, 19] have shown that when the nodes of an n-star are ranked using the pipeline structured outlook then there exists a path from any node in ${}^{i}S_{n\text{-}1}$ to its peer node in ${}^{i+1}S_{n\text{-}1}$ or ${}^{i\text{-}1}S_{n\text{-}1}$, where $1\leq i\leq n$. The length of this path is either one or at most three. However, the structural outlook based on the pipeline view for the n-star graph is also insufficient because it generates unbalanced stage with the number of nodes in each stage. For example, the number of nodes for each stage in the 11-stage pipeline is 3628800, which is highly unbalanced and this results in a non-desirable network.

In an attempt to address the problems of the star graph, Day and Tripathi [28] have proposed the arrangement graph as a generalisation of the star graph. The (m,k)-arrangement graph, denoted by $A_{m,k}$ where $1 \leq k \leq m-1$, has $m!/(m-k)!$ nodes. Each node is labelled with a unique arrangement of k symbols chosen from $\langle m \rangle$. The network has a diameter $\lfloor \frac{3}{2} k \rfloor$ and the node degree is $k(m-k)$ [28]. Two nodes are connected if, and only if, they differ in exactly one of their k symbols. Figure 1.4 shows the topology of $A_{4,2}$ where the nodes 41,42 and 43 are neighbours.

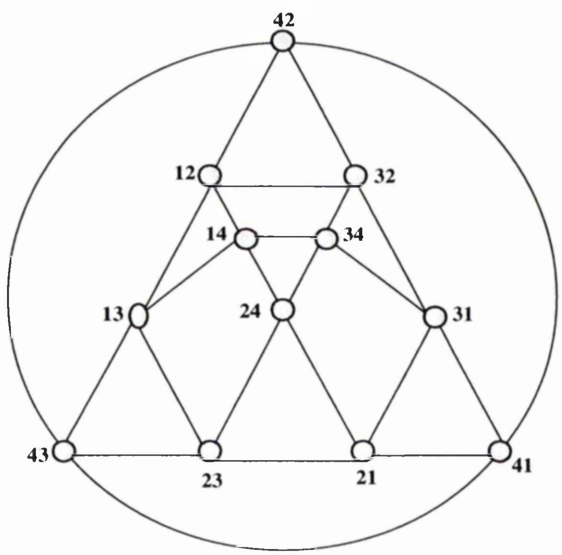


Figure 1.4: The arrangement graph $A_{4,2}$.

1.4 Vertex Product Networks

In this thesis a new classification for a large set of interconnection networks is provided. These networks are termed "*Vertex Product Networks*" (VPN). The VPN provides means for unification by which the studies on topological properties for these networks can be done at the same time. Studying the VPN is important because they enable us to define unlimited number of new networks and further study and analyse existing networks. As an example of the VPN is the existing Cross Product

Networks (CPN), which are constructed from the graph product of two factor networks [4, 33]. Another example of the VPN is the Optical Transpose Interconnection Systems (OTIS), which provides the best of both the "electronic" and "optics" worlds by using free-space optical interconnects to connect distant processors and electronic interconnect for processors that are in the same group [34, 35]. A formal definition for the vertex product network is given below.

Definition 1.2: Given any two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, where V_1 and V_2 are the sets of vertices, E_1 and E_2 are the sets of edges in G_1 and G_2, respectively.

The Vertex Product Network $(G_{VPN}) = (G_1 \otimes G_2)$ of two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ is represented by an undirected graph $(G_{VPN}) = (V_{VPN}, E_{VPN})$ where V_{VPN} and E_{VPN} are defined as follows.

1- $V_{VPN} = \{\langle u_1, v_1 \rangle \mid u_1 \in V_1 \text{ and } v_1 \in V_2\} = V_1 \otimes V_2$ ($\otimes$ is the Cartesian product)

2- For any $u = \langle u_1, u_2 \rangle$ and $v = \langle v_1, v_2 \rangle$ in V_{VPN}, (u, v) is an edge in E_{VPN} defined by the function f such that $E_{VPN} = \{(u, v) \mid f(u, v) = 1\}$. The function f is defined as follows: -

$$f(u, v) = \begin{cases} 1 & \text{if } ((u_2, v_2) \in E_2 \text{ and } u_1 = v_1) \text{ or } (u_1 = v_2 \text{ and } u_2 = v_1) \quad \text{"OTIS"} \\ 1 & \text{if } \{\{(u_2 = v_2) \text{ and } (u_1, v_1) \in E_1\} \cup \{\{(u_1 = v_1) \text{ and } (u_2, v_2) \in E_2\}\} \quad \text{"CPN"} \\ 0 & \text{otherwise} \end{cases}$$

From the above definition, we observe that the set of vertices in both the CPN and OTIS are composed from the Cartesian product of their constituents' vertex sets. The original definition for the CPN covers also the OTIS set of vertices. Simply the set of vertices in the VPN, $V_{VPN} = V_1 \otimes V_2$ (where $\otimes$ is Cartesian product on sets), that can be generated from $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$.

The set of edges of the CPN is defined in a mutual exclusive form while the OTIS defines edges in a transpose form. The above definition creates the linkage that is controlled by a "binary" function (f) whose domain is V_{VPN} and its range is {0,1}. Given any pair of vertices from V_{VPN}, f returns 1 if the two vertices are connected and 0 otherwise. Hence the set of edges is given by $E_{VPN} = \{(u, v) \mid f(u, v) = 1\}$. In the following subsections the CPN and OTIS are discussed in more detail.

1.4.1 Cross Product Networks (CPN)

The CPN were proposed by Day and Al-Ayyoub [4] as a theoretical framework and a tool for generating and defining new interconnection networks and further studying and analysing some of the known interconnection networks. The CPN are basically constructed by "multiplying" two known topologies of the same or different kinds. The cross product network of two interconnection networks given by two undirected graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$, where V_1 and V_2 are the set of vertices of G_1 and G_2 and E_1 and E_2 are the set of edges of G_1 and G_2, respectively. The formal definition of the cross product of the two graphs is defined as follows:

Definition 1.3: The cross product $G=G_1 \otimes G_2$ of two undirected connected graphs $G_1=(V_1, E_1)$ and $G_2=(V_2, E_2)$ is the undirected Graph $G=(V, E)$, where V and E are given by:

$V=\{\langle x_1, y\rangle \mid x_1 \in V_1 \textit{ and } y \in V_2\}$ *and*

$E=\{(\langle x_1,y\rangle, \langle y_1,y\rangle) \mid (x_1,y_1) \in E_1\} \cup \{(\langle x,x_2\rangle,\langle x,y_2\rangle) \mid (x_2,y_2) \in E_2\}$.

So for any $u = \langle x_1, x_2\rangle$ and $v = \langle y_1, y_2\rangle$ in V, (u, v) is an edge in E if, and only, if either (x_1, y_1) is an edge in E_1 and $x_2 = y_2$, or (x_2, y_2) is an edge in E_2 and $x_1 = y_1$. The edge (u, v) is called a G_1-edge if (x_1, y_1) is an edge in E_1, and it is called G_2-edge if (x_2, y_2)

is an edge in E_2 [4]. The size, degree, diameter and number of links of the cross product of two networks are defined next.

Definition 1.4: If G_1 and G_2 are two undirected connected graphs of respective size s_1 and s_2 and have respective diameters δ_1 and δ_2, then [4, 33]:

1) $G_1 \otimes G_2$ is connected.

2) The size s of $G_1 \otimes G_2$ is $s = s_1.s_2$.

3) The diameter δ of $G_1 \otimes G_2$ is $\delta = \delta_1 + \delta_2$.

4) The degree of a node $u = \langle x_1, x_2 \rangle$ in $G_1 \otimes G_2$ is equal to the sum of the degrees of vertices x_1 and x_2 in G_1 and G_2, respectively.

5) The number of links for the product network, is (size·degree)/2.

The *star-cube*, which was proposed in [43], is an example of the CPN. It is defined as the cross product of the star graph and hypercube. The (n, h)-star-cube, $SQ_{n,h}$ where $n, h > 1$, is undirected graph whose set of nodes is the set of pairs $\langle x_1x_2 \ldots x_n ,$ $a_1a_2 \ldots a_h \rangle$ such that $x_1x_2 \ldots x_n$ is a node of S_n and $a_1a_2 \ldots a_h$ is a node of Q_h. A node $u =$ $\langle x_1x_2 \ldots x_n , a_1a_2 \ldots a_h \rangle$ is connected to a node $y = \langle y_1y_2 \ldots y_n, b_1b_2 \ldots b_h \rangle$ if, and only, if $x_1x_2 \ldots x_n$ is connected to $y_1y_2 \ldots y_n$ in S_n (in this case u and v are said star-connected) or $a_1a_2 \ldots a_h$ is connected to $b_1b_2 \ldots b_h$ in Q_h (in this case u and v are said cube-connected) but not both [43].

When u and v are star-connected (respectively cube-connected) the edge (u, v) is called the star-edge (respectively cube-edge). Figure 1.5 shows the cross product of a 3-star graph and 2-cube. One of the major problems with the star-cube network is that, it does not have any proposed algorithmic structural outlooks in the literature to support parallel algorithms such as matrix computation [43].

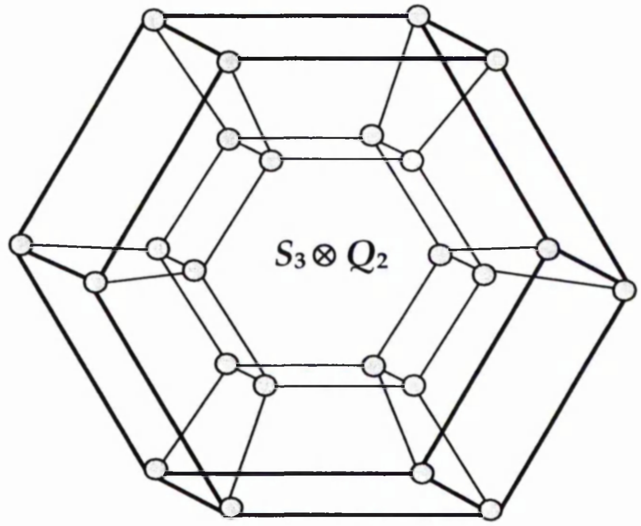


Figure 1.5: The star-cube network, $SQ_{n,h}$.

Another example of the CPN that has been investigated includes the hyper-Petersen network, which is a product of the binary hypercube and the well-known Petersen graphs [26]. As we will be discussing later, our proposed arrangement-star network is another example of the CPN as it is constructed from the product of the star and arrangement graphs [7, 37].

1.4.2 Transpose Networks

The Optical Transpose Interconnection Systems is a transpose network, first proposed by Marsden [34]; the abbreviation OTIS will be used to refer to the interconnection network build from transpose product. An OTIS-based computer contains N^2 processors partitioned into N groups with N processors each. A processor is indexed by a pair $\langle x, y\rangle$, $0 \le x,y < N$ where x is the group index and y is the processor index. Processors within a group are connected by a certain interconnecting topology, while transposing group and processor indexes achieve inter-group links. Figure 1.6 shows a 16 processor OTIS connection where the bold arrows represent an optical links between two processors of two different groups. For instance, the intra-group connects may be mesh-based, hence the term OTIS-mesh is used to denote this network. The terms OTIS-computer and OTIS-networks refer to parallel architecture based on transpose networks and will be used interchangeably in the rest of the thesis.

OTIS-networks are basically constructed by "multiplying" a known topology by itself. The set of vertices is equal to the Cartesian product on the set of vertices in the factor network. The set of edges consists of edges from the factor network and new edges called the *transpose* edges. The formal definition of OTIS-networks is given below.

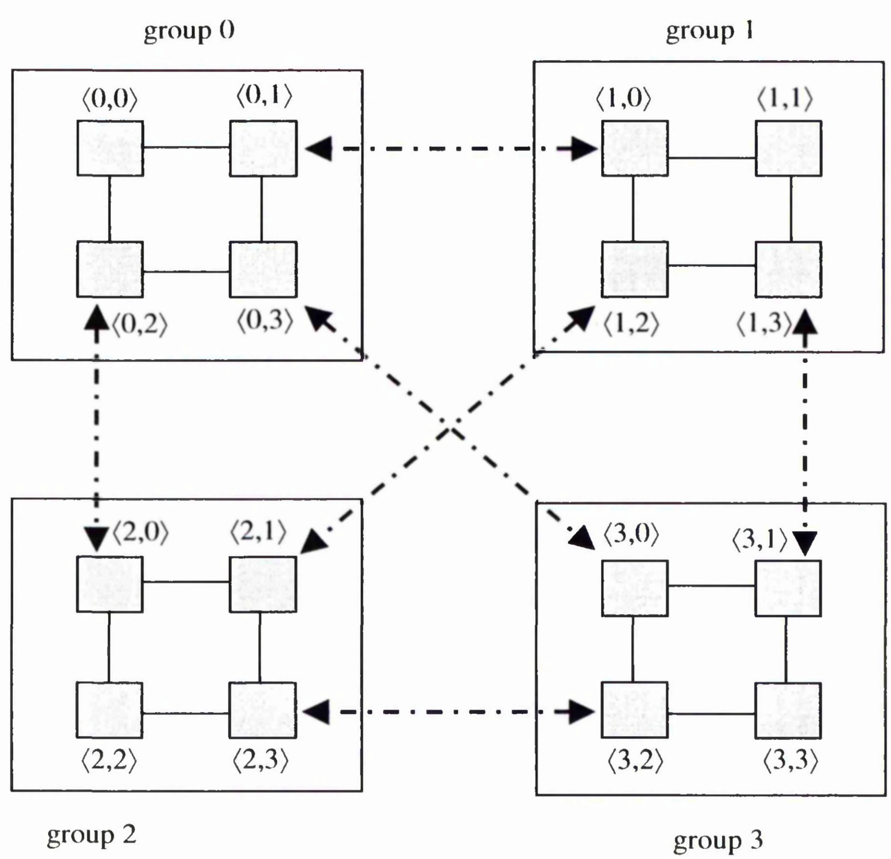


Figure 1.6: An OTIS connection with 4 groups of 4 processors each.

Definition 1.5: Let $G_0 = (V_0, E_0)$ be an undirected graph representing a factor network. The OTIS-$G_0 = (V, E)$ network is represented by an undirected graph obtained from G_0 as follows $V = \{\langle x, y\rangle \mid x, y \in V_0\}$ and $E = \{(\langle x, y\rangle, \langle x, z\rangle) \mid \text{if } (y, z)\in E_0\} \cup \{(\langle x, y\rangle, \langle y, x\rangle) \mid x, y \in V_0\}$.

The set of edges E in the above definition consists of two subsets, one is from G_0, called G_0-type edges, and the other subset contains the *transpose* edges. The OTIS approach suggests implementing G_0-type edges by electronic links since they involve intra-chip short links and implementing transpose edges by free space optics. Throughout this thesis the terms "*electronic move*" and the "*OTIS move*" (or "*optical move*") will be used to refer to data transmission based on electronic and optical technologies, respectively.

Definition 1.5 covers a wide class of OTIS-networks. In fact, for any known factor network G_0, a new OTIS-network can be obtained by the above definition. The OTIS-mesh [38], OTIS-hypercube [41] are only few instances of such networks that have been investigated in the literature.

1.5 Motivations

Recently, there has been an increasing interest in a class of interconnection networks called product networks [4, 7, 25, 27, 49]. In this thesis, the VPN is provided as a new classification for the set of product networks, and a number of related issues are discussed, including the design and evaluation of efficient structural outlooks for algorithm development on this class of networks. The importance of studying the VPN stems from the fact that it allows us to define unlimited number of new networks and further study and analyse more deeply some known networks such as star [3], hypercube [8], and arrangement networks [28].

A widely studied interconnection network is the *star graph* [3]. The star graph has been proposed as an attractive alternative to the hypercube due to its superior characteristics. The star graph has three significant advantages over the hypercube: a lower degree, a smaller diameter and a smaller average diameter for a comparable number of nodes. The star graph, however, suffers from some drawbacks [25]. One major problem with the star graph is related to its poor scalability. Despite its attractive topological properties, the star graph has not been used in practical systems yet. One reason for this may be attributed to the difficulty in developing parallel algorithms on this network for common parallel applications. Mapping of data and tasks on the star graph is not as obvious as it is the case for the hypercube and mesh [32].

The *arrangement graph* as a generalisation of the star graph was proposed by Day and Tripathi [28], in an attempt to address the scalability problem in the star graph. The arrangement graph slightly improves the scalability problem of the star graph and preserves the desirable properties of this graph. However, since its introduction there has been little work done on the development of new algorithms for this network. In fact, the arrangement graph inherits the major difficulties in developing efficient algorithms that could take advantage of the attractive topological properties of this network.

In this thesis, we propose a new VPN, referred to here as the *arrangement-star*. This network is constructed from the graph product of the arrangement and star networks. We prove the viability of the proposed arrangement-star as an underlying topology for parallel computers. The arrangement-star network not only brings a solution to the scalability problem from which the star and arrangement graph suffer, but also it preserves all the attractive features of these two networks.

Furthermore, we show that the arrangement-star network allows the development of efficient algorithmic structural outlooks that support a wide class of parallel applications, e.g. matrix computation problems, based on the grid and pipeline structures. Results from a performance comparison, based on the proposed gird and pipeline structural outlooks, reveal that the new arrangement-star network outperforms the existing star and hypercube networks in terms of communication cost.

Optical Transpose Interconnection Systems (OTIS) networks are another example of the VPN. These networks are basically constructed by "multiplying" a known topology by it self and are implemented using both free-space optical and electronic interconnect technologies. Recently there has been a growing interest in the study of parallel algorithms for optoelectronic networks in general and for OTIS-networks in particular [15, 20, 24, 36, 38, 41]. For instance, Sahni and Wang [38] have presented and evaluated various algorithms on the OTIS-mesh (a special case of OTIS-networks), including basic data rearrangements, routing, selection and sorting. They have also developed algorithms for various matrix multiplication operations [62] and image processing [24].

Aside from the above-mentioned works, there has been little research work devoted to the study of algorithms on OTIS-networks. In an effort to fill this gap, this thesis proposes two new structural outlooks for developing algorithms on these networks. These structural outlooks are based on grids and pipelines as popular structures that support vast body of applications ranging from linear algebra to divide-and-conquer type of algorithms, sorting, and Fourier transforms. The proposed structural outlooks are general in the sense that no specific factor network or problem domain is assumed. This study shows that for some factor networks, such as the mesh and hypercube, the performance of the proposed structural outlooks outperforms the best-known results for these factor networks. Also we show how the proposed structural outlooks can be used to design new parallel algorithms on OTIS-networks for solving systems of linear equations, an important problem that is encountered in various fields of science and engineering [9, 11, 12, 16].

1.6 Outline of the Thesis

Chapter 2 proposes the *arrangement -star* network, as a case study on the VPN. This network is constructed from the graph product of the star and arrangement graphs. Many of its topological properties are derived including degree, diameter, size, degree of accuracy, number of links and all-port one-to-all-broadcast. A comparative analysis of the star, arrangement and arrangement-star networks are conducted and the results reveal that the proposed network has superior topological properties over its constituents: the star and arrangement graphs.

Chapter 3 develops two *efficient structural outlooks*, namely the grid and pipeline structures, for algorithm development on the new arrangement-star network. The proposed structural outlooks are then used to conduct a performance comparison between the arrangement-star, star and hypercube.

Chapter 4 introduces the *Optical Transpose Interconnection Systems* (OTIS) networks as another case study on the VPN. Some of the basic topological properties of OTIS-networks are derived, including size, degree, diameter, number of links, routing and broadcasting. These properties are used in the subsequent two chapters.

Chapter 5 develops and evaluates two general *structural outlooks* for algorithm development on OTIS-networks. The proposed structural outlooks allow efficient mapping of a wide class of algorithms into OTIS-networks. *Timing models* for measuring the performance of the proposed structural outlooks are also provided. Through these models, the performance of various algorithms on OTIS-networks are evaluated and compared with their counterparts on conventional electronic interconnection systems.

Chapter 6 uses the proposed grid structural outlook, discussed in Chapter 5, to introduce *parallel algorithms* for solving systems of linear equations on OTIS-networks.

Chapter 7 summarises the results presented in the thesis and discusses some possible directions for future research work.

Chapter 2

A Case Study on Vertex Product Networks: The Arrangement-star Network

2.1 Introduction

Network topology defines the way nodes are connected. Graph theory is a useful tool in the description of network topologies, where each processor corresponds to a vertex and each communication link to an edge [11]. It is often conventional to refer to a network and its corresponding graph as if they were identical. Among the key features of a given topology are the degree, diameter, vertex symmetry, connectivity, and hierarchical structure.

A number of interconnection network topologies have been suggested in the literature, which address one or more of the above features [25, 27, 28, 36, 51]. The proposed topologies range from simple graph, such as cycles and complete graphs to more sophisticated graphs such as stars [3] hypercubes [2, 22], shuffle-exchanges [73], and mesh connected trees [49].

In this chapter a new interconnection network is proposed, referred to as the *arrangement-star* network as a case of study on vertex product networks. The arrangement-star is constructed from the cross product of the star and arrangement graphs. Our analysis reveals that the proposed topology has superior topological properties over its factors: the star and arrangement graphs. Besides having a smaller diameter, node degree, and number of links, it has a lower broadcasting cost and more flexibility in choosing the desired network size. Furthermore this chapter introduces a new measure called the *degree of accuracy*, which is used to compare the three topologies: star, arrangement, and arrangement-star.

The remainder of this chapter is organised as follows. Section 2 discusses the background and motivation behind the proposal of the arrangement-star network. Section 3 provides the necessary notation and definitions that will be useful in our study, and then formally presents the arrangement-star network. Section 4 discusses some general topological properties of the arrangement-star graph. Section 5 conducts a comparison of some of basic properties of the star, arrangement and arrangement-star graphs. Finally, Section 6 summarises this chapter.

2.2 Background

During the last decade a large variety of interconnection networks for high-speed parallel systems have been proposed [3, 8, 15, 25, 26, 28, 49]. The *star graph* [3] is one such example that has been widely studied in the literature [2, 3], it has been proposed as an attractive alternative to the hypercube. Several properties of this network have been investigated including its basic topological properties [3], parallel path characterization [2], and embedding [29, 50]. Akers and Krishnamurthy [1, 3] have shown that the star graph has several advantages over the hypercube including a smaller diameter, smaller average diameter and lower degree for a fixed network size. The star graph has also been shown to be edge and vertex symmetric [3] and is maximally fault tolerant [2]. Furthermore, a limited number of parallel algorithms for solving some well-known problems on the star graph have been reported in the literature, including computing fast Fourier transforms [44], matrix decomposition [32], broadcasting [45], and sorting [48].

The star graph, however, has a few drawbacks [25]. One of the major problems of the star graph is related to its scalability. The size of the star graph increases according to a factorial function, and thus grows widely very rapidly; for example, the value of 7! is equal to 5040 while the value of 11! is about forty million. Despite its attractive topological properties, the star graph has not been used in practical systems yet.

In an attempt to address the scalability problem in the star graph, Day and Tripathi [28] have proposed the *arrangement graph* as a generalisation of the star graph. The arrangement graph is a family of undirected graphs that contains the star graph family. It slightly brings a solution to the problem of the scalability, which the star graph suffers from (i.e. the problem of growth of the number $n!$ of nodes in the n-star). It also preserves all the nice qualities of the star graph topology including, hierarchical structure, vertex and edge symmetric, simple shortest path routing and many fault tolerance properties [28]. Still a common drawback of the star and arrangement graphs is the restriction on the number of nodes: $n!$ for the star graph and $m!/(m-k)!$ for the arrangement graph. The set of values of $n!$ (or $m!/(m-k)!$) is spread widely over the set of integers; so, one will be faced with the choice of too few or too many available nodes.

The graph product has recently been investigated in [4, 22, 33] as a graph-theoretical framework for generating and analysing interconnection networks topologies. Day and Al-Ayyoub [4] have used this framework to investigate properties of existing networks such as scalability, vertex symmetry, routing, broadcasting, embedding, recursive structure, and the existence of maximum-size families of node-disjoint paths along with some results on node-connectivity and an upper bound for the fault-diameter [4, 25]. Several other researchers have investigated the graph product on existing networks. For instance, Das [53] has studied the graph product of the hypercube and Peterson networks. Al-Ayyoub and Day [25] have shown that the hyperstar (a product of star graphs) outperforms other product networks on various aspects, including a lower degree and diameter. Other examples of product networks that have been studied in the literature include the hyper-Debruijn [27], the star-hypercube [43], and mesh connected trees [49].

This chapter considers the graph product of the arrangement and star graphs with the aim to enhance the topological characteristics of these two graphs with the elegant capabilities of product networks [7, 37]. As we shall see below, our study reveals that the new arrangement-star network has superior topological properties over both the star and arrangement graphs and it brings a solution for the common problem, which these two networks suffer from (i.e. the scalability problem). The arrangement-star network also preserves all the nice qualities of the star and arrangement graphs topologies including, hierarchical structure, vertex and edge symmetric, simple shortest path routing and many fault tolerance properties.

2.3 Notations and Definitions

The n-star graph, denoted by S_n, has $n!$ nodes each labelled with a unique permutation on $\langle n \rangle=\{1,\dots,n\}$. Any two nodes are connected if, and only if, their corresponding permutations differ exactly in the first position and any other position.

Definition 2.1: An n-star graph, denoted S_n, is an undirected graph consisting of $n!$ vertices labelled with the $n!$ permutations of n symbols (we use the symbols 1,2, ... , n). There is an edge between any two vertices if, and only if, their labels differ in the first and any other position [3].

An arrangement graph is specified by two parameters m and k, satisfying $1 \leq k \leq m$. For simplicity let $\langle m \rangle = \{1,2,\ldots,m\}$ and $\langle k \rangle = \{1,2,\ldots,k\}$.

That is, the nodes of $A_{m,k}$ labelled with a unique arrangements of k elements out of m symbols $\langle m \rangle$, and the edges of $A_{m,k}$ connect arrangements which differ in exactly one of their k positions An edge of $A_{m,k}$ connecting two arrangements, which differ only in position i called an i-edge. In this case, p and q are i-adjacent and q is called (i, q_i)-neighbour of p [28]. The (m,k)-arrangement graph $A_{m,k}$ is regular of degree $k(m-k)$ and of size $m!/(m-k)!$, and diameter $\lfloor 3k/2 \rfloor$. The $(m, m-1)$-arrangement graph $A_{m,m-1}$ is isomorphic to S_n graph [28], and the $(m,1)$-arrangement graph is isomorphic to the complete graph with m nodes.

Definition 2.2: The (m,k)-arrangement graph $A_{m,k} = (V_1, E_1)$, $1 \leq k \leq m-1$ is defined as follows [28]:

$V_1 = \{p_1p_2 \ldots p_k \mid p_i \in \langle m \rangle \text{ and } p_i \neq p_j \text{ for } i \neq j\} = P_k^m$, and

$E_1 = \{(p,q) \mid p \text{ and } q \text{ in } V_1 \text{ and for some } i \text{ in } \langle k \rangle, p_i \neq q_i \text{ and } p_j = q_j \text{ for } j \neq i\}$.

The graph product is an elegant mathematical representation for studying interconnection networks; see Definition 1.3 in Chapter 1 for a formal definition of the graph product. It has been used as a tool for generating new attractive interconnection networks [7, 22, 25, 26, 27, 43]. The graph product of the arrangement and star graphs is applied to propose a new network, the *arrangement-*

star, with the aim of enhancing the topological characteristics of the factor networks [7, 37, 71]. Below, a formal definition of the arrangement-star graph is given.

***Definition* 2.3:** The arrangement-star graph is the cross product of the n-star graph and the (m, k)-arrangement graph, and is given by $AS_{n,m,k} = A_{m,k} \otimes S_n$ such that $n>1$ and $1 \leq k \leq m$.

Note that if G_1 and G_2 are two undirected graphs then for any node $X = \langle x_1, x_2 \rangle$ in the cross product graph, $G = G_1 \otimes G_2$, has an address consisting of two parts, one coming from G_1 and the other coming from G_2. We will denote the earlier part by $lp(X)=x_1$ and the later part by $rp(X)=x_2$.

Figure 2.1 shows the topology of $AS_{2,3,2}$ that is obtained from the graph product of S_2 and $A_{3,2}$ networks. A node $X = \langle u, v \rangle$ in $AS_{2,3,2}$ consisting of two parts, left part coming from the star graph and the right part coming from the arrangement graph. Two nodes $X = \langle u, v \rangle$ and $Y = \langle u', v' \rangle$ are connected if, $lp(X) = lp(Y)$ and $rp(X)$ is connected $rp(Y)$ in $A_{m,k}$ (in this case X and Y are said arrangement-connected) or $rp(X)= rp(Y)$ and $lp(X)$ is connected $lp(Y)$ in S_n (in this case X and Y are said star-connected). For instance in Figure 2.1 the node *ab*13 is connected to the node *ab*12, and the node *ab*23 is connected to the node *ba*23.

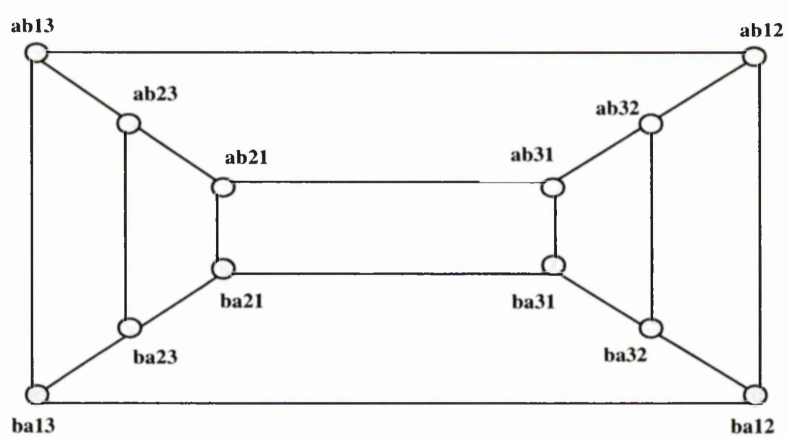


Figure 2.1: Arrangement-star graph, $AS_{2,3,2}$.

2.4 General Topological Properties

This section discusses and derives some of the basic topological properties of the arrangement-star network including size, degree, diameter, parallel paths, average diameter, optimal broadcasting, optimal routing, fault diameter and connectivity. Table 2.1 summarises the topological properties of the star and arrangement graphs along with those of the arrangement-star for comparison purposes. These topological properties have been derived using the theoretical framework for analysing product graphs, proposed by Al-Ayyoub and Day [4] as well as Youssef [33]. For instance, if G_1 and G_2 are two undirected graphs of respective sizes s_1 and s_2 and of respective diameters δ_1 and δ_2 then the size s and the diameter δ of the graph product $G_1 \otimes G_2$ is equal to $s = s_1.s_2$ and $\delta = \delta_1+\delta_2$. The degree of a node in $G_1 \otimes G_2$ is the sum of the degrees of G_1 and G_2 [4, 33]. It therefore follows that the size, degree and diameter of the arrangement-star network are equal $m!.n! / (m-k)!$, $n+k(m-k)-1$ and $\lfloor \frac{3}{2}(n-1) \rfloor + \lfloor \frac{3}{2}k \rfloor$ respectively. Similarly, it can be easily shown that the average diameter of the arrangement-star network is $n+2/n+H_n-4+H_k+k(k-2)/m$ where $n+2/n+H_n-4$ and $H_k+k(k-2)/m$ are the average diameters of the n-star graph and (m, k)-arrangement graph, where H_n, H_k are the Harmonic number of star and arrangement graphs, i.e., $H_n = \sum_{i=1}^{n} 1/i$ and $H_k = \sum_{i=1}^{k} 1/i$ [3, 51].

The existence of node-disjoint (parallel) paths between any pair of nodes is an important feature of any interconnection network. For instance, these paths are useful for speeding up transfer of large amounts of data and for offering alternative paths in the cases of node or link failure [4]. Tripathi and Day [52] have shown that there are $k\ (m-k)$ parallel paths between any two nodes of (m,k)-arrangement network, and the length of each parallel path is at most the minimum path between the two nodes plus four. Also, there are n-1 parallel paths between any given two nodes in the n-star; the length of each of these paths is at most the minimum distance between any two nodes plus four [2]. As shown in Table 2.1, the arrangement-star network has complete parallel paths with maximum length equal to the minimum path between the two nodes plus 4. Note that we have used the abbreviation *CPP/MLI*-4 to refer to the graph that has complete parallel paths with maximum length increase 4. The length of each of these paths is at most the minimum distance between any two nodes plus four [2, 4].

Property	Star	Arrangement	Arrangement-star
Size [4, 33]	$n!$	$m!/(m-k)!$	$m!\cdot n!/(m-k)!$
Degree [4, 33]	n-1	$k(m-k)$	$n+k\ (m-k)$-1
Diameter [4, 33]	$\lfloor \frac{3}{2}(n-1) \rfloor$	$\lfloor \frac{3}{2}k \rfloor$	$\lfloor \frac{3}{2}(n-1) \rfloor + \lfloor \frac{3}{2}k \rfloor$
Parallel paths [2, 4, 52]	*CPP/4-MLI*	*CPP/4-MLI*	*CPP/4-MLI*
Average diameter [3, 33, 51]	$n+2/n+H_n-4$	$H_k+k(k-2)/m$	$n+2/n+H_n-4+H_k+k(k-2)/m$
Optimal broadcasting [45]	B_1	B_2	*Apply* B_1 *then* B_2 *or* B_2 *then* B_1
Optimal routing [4]	$\Re_1$	$\Re_2$	$\Re(\langle x_1,x_2\rangle,\langle y_1,y_2\rangle) = \begin{cases} \langle \Re_1(x_1,y_1), y_2 \rangle & \text{if } x_1 \neq y_1 \\ \langle x_1, \Re_2(x_2,y_2) \rangle & \text{if } x_1 = y_1 \end{cases}$
Fault diameter [2, 4]	$\lfloor \frac{3}{2}(n-1) \rfloor + 4$	$\lfloor \frac{3}{2}k \rfloor + 4$	$\lfloor \frac{3}{2}(n-1) \rfloor + \lfloor \frac{3}{2}k \rfloor + 4$
Connectivity [4, 52]	n-1	$k(m-k)$	$n + k\ (m-k) - 1$

Table 2.1: The topological properties for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$

Another important measure of an interconnection network is its fault diameter, i.e. the maximum diameter of any graph obtained from the original graph by removing at most f nodes, where f is the number of nodes to be removed and the graph remains connected [2, 4]. The fault diameter of the arrangement and star graphs are $\lfloor \frac{3}{2}k \rfloor + 4$ and $\lfloor \frac{3}{2}(n\text{-}1) \rfloor +4$ [25, 28], respectively. The fault diameter of the arrangement-star graph is $\lfloor \frac{3}{2}(n\text{-}1) \rfloor + \lfloor \frac{3}{2}k \rfloor + 4$. Furthermore, the connectivity of the star and arrangement graphs are n-1 and k (m-k), we conclude that the connectivity of the arrangement-star graph is equal to $n + k(m\text{-}k)\text{-}1$, which is equal to the degree $AS_{n,m,k}$, so the proposed arrangement-star has the most-node connectivity [4, 25].

Let B_1 and B_2 be two optimal broadcasting algorithms for n-star and (m,k)-arrangement networks in sequential. The optimal broadcasting algorithm for the arrangement-star network is obtained by applying the broadcasting algorithm B_1 for n-star then by applying the broadcasting algorithm B_2 for (m,k)-arrangement network or by applying B_2 then B_1 [4, 33]. Another important measure for an interconnection network is related to its routing. In table 2.1 the symbols $\Re_1$ and $\Re_2$ stands for the optimal routing algorithms for the star and arrangement networks while $\Re$ stands the optimal routing algorithms for the arrangement-star network. In the routing in the arrangement-star network the message routes in G_1-edges, where G_1 can be the star (or arrangement graph), until the message reaches its destination in G_1. Once this is achieved the routing continues along G_2-edges, where G_2 can be the arrangement (or star graph) [4].

2.5 Comparative Results on the Topological Properties

This section conducts a comparative study between the three graphs star, arrangement, and arrangement-star. We base our comparison on the most widely used criteria such as degree, diameter, scalability, number of links and broadcasting cost [4, 6, 25, 45]. For most of these criteria, the results confirm the superiority of the arrangement-star graph, making it an attractive alternative to the star and arrangement graphs. Furthermore, a new criterion referred to as, the d*egree of* *accuracy*, will be used in our present comparative analysis to provide further evidence on the superior characteristics of the proposed arrangement-star graph.

In what follows, we compare the three static parameters: *size, degree* and *diameter* of the star graph, S_n, arrangement graph, $A_{m,k}$, *and arrangement-star* graph, $AS_{n,m,k}$. We plot in Figure 2.2 the network size against the matching probability of the desired size for the three graphs when the network size is in the range of $[2^{10}–2^{18}]$. All the possible sizes for the star, arrangement and arrangement-star graphs were generated, and then arranged in decreasing order. The probability of the number of network sizes was issued by dividing the number of the actual network sizes over the desired network size. The results reveal in Figure 2.2 that the number of the topologies with different sizes that can be issued in the arrangement-star graph is larger than that in either star or arrangement graphs. This finding demonstrates that the $AS_{n,m,k}$ brings a solution to the scalability problem of the star graph and thus overcomes the limitation that makes the star graph a less desirable topology [25].

Figure 2.3 plots the degree of the three graphs against the network sizes as an integer number ranging from $[2^{10}–2^{26}]$. The results show the superiority of the arrangement-star network in terms of node degree as it has a smaller degree than the arrangement graph and the same degree of the star graph for the same number of nodes. In term of the diameter, on the other hand, Figure 2.4 reveals that the arrangement-star graph has a smaller diameter than the star graph for the same number of nodes.

The number of links that is required by a given network topology is an important factor that affects its feasibility of implementation and therefore its adoption in practical systems [6]. This is because the number of links can provide a rough estimate of the cost for implementing a topology [6]. Figure 2.5 plots the obtained number of links against network size (in Logarithmic scale) for the three networks. The result shows that the arrangement-star graph has more flexibility than the star and arrangement graphs in choosing the number of required links for a given graph size. This indication gives another evidence on the superiority of the arrangement-star graph over the star and arrangement graphs. This is a motivation for future considerations for the arrangement-star graph to be studied much more deeply and to give more effort for this graph.

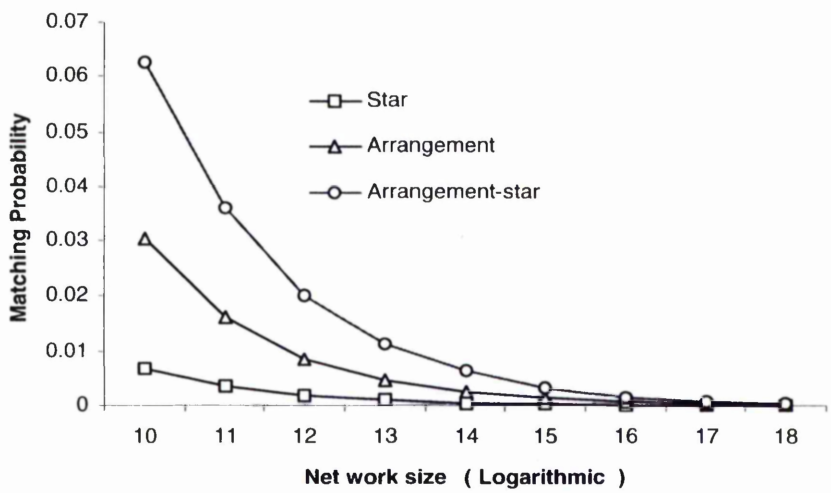


Figure 2.2: The matching probability for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

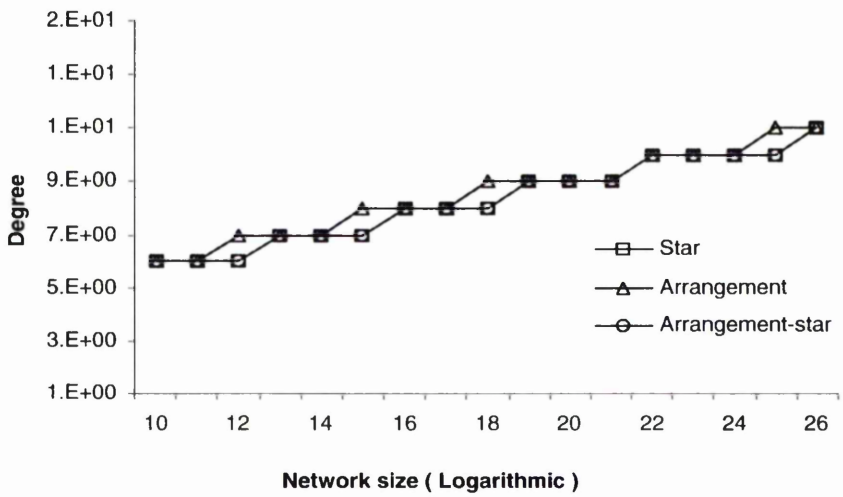


Figure 2.3: Comparing the Degree for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

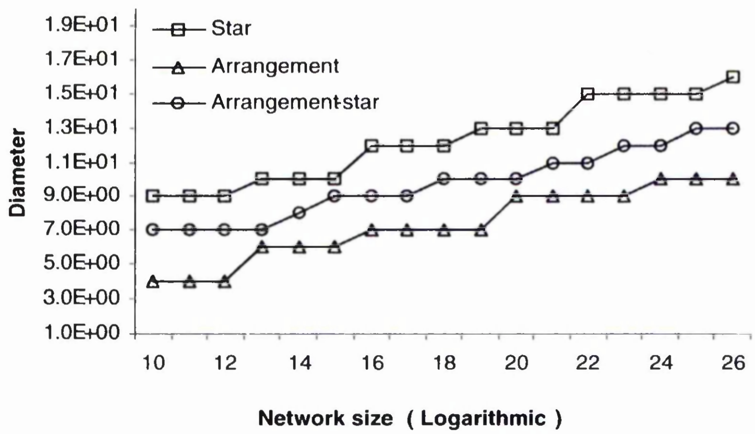


Figure 2.4: Comparing the Diameter for the three graphs, S_n, $A_{m,k}$ and

One of the most widely used criteria to evaluate a given interconnection network is the cost of performing broadcast communication as this is an important communication operation required by many parallel applications [45]. Table 2.2 shows the *lower bound* expressions for obtaining latency to perform one-to-all broadcasting in the three graphs. These expressions have been derived using the results of [45]. According to [45], the cost of a multiple-port one-to-all broadcasting in a network of degree Δ and diameter δ is given by $\left(\sqrt{M \cdot a/(\beta \cdot \Delta)} + \sqrt{\delta - 1}\right)^2 \beta$, where M is the message length, β is the message latency and a is the unit transmission cost [13]. This model determines the lower bound on the broadcasting cost in any vertex transitive network.

For the sake of the present discussion M, β and a have been set to 1024 byte, 1000 μs and 1μs, as suggested in similar previous studies [32, 45]. Figure 2.6 depicts broadcasting latency in the three graphs as a function of the network size by using the expressions of Table 2.2. The figure shows that the $AS_{n,m,k}$ outperforms the S_n. However, it has a lower performance than the $A_{m,k}$. This is because the former topology has a larger diameter.

Network	All-port broadcasting cost
Arrangement, $A_{m,k}$	$(\sqrt{\frac{M \cdot a}{\beta \cdot k(m-k)}} + \sqrt{\lfloor \frac{3}{2}k \rfloor - 1})^2 \beta$
Star, S_n	$(\sqrt{\frac{M \cdot a}{\beta \cdot (n-1)}} + \sqrt{\lfloor \frac{3}{2}(n-1) \rfloor - 1})^2 \beta$
Arrangement-star, $AS_{n,m,k}$	$(\sqrt{\frac{M \cdot a}{\beta[(n-1) + k(m-k)]}} + \sqrt{\lfloor \frac{3}{2}(n-1) \rfloor + \lfloor \frac{3}{2}k \rfloor - 1})^2 \beta$

Table 2.2: The broadcasting cost for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

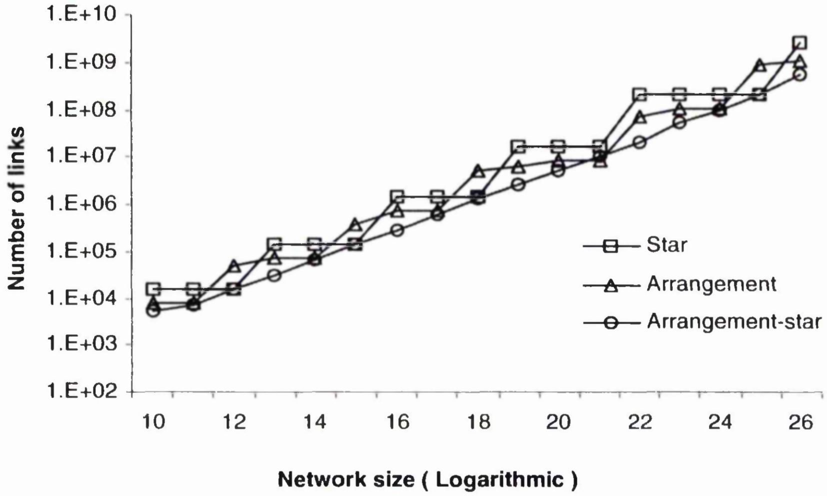


Figure 2.5: Comparing number of links for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

The degree of accuracy

This section introduces a new measure called the *degree of accuracy*, which is used to compare the three topologies. As mentioned above, the star graph does not scale easily to different sizes; all sizes have a problem related to the number of nodes: $n!$ for the n-dimensional star graph. This measure will assess whether the arrangement-star graph suffers from the same limitation. The degree of accuracy on the issued topology size (Θ) measures how far is the issued size, Θ, from the desired size, N. In this measure we will consider only those issued network sizes which they are 10 percent of the desired size N such that $(0.10*N-N) \leq \Theta \leq (0.10*N+N)$. This measure is important because it affects the cost of the topology in direct way, and hence the decision of designing a new topology. For the sake of illustration, let us assume that the desired size is in the range of $[2^{10} - 2^{26}]$ nodes.

In this measure, we find the required issued size $(0.10{*}N\text{-}N) \leq \Theta \leq (0.10{*}N{+}N)$ of the three graphs then we divide this result by the desired size (N) multiplying the result by 100. Figure 2.7 summarises the obtained values for the three graphs. For the arrangement-star graph 100% of the issued sizes are within 10% of the desired size. While for the arrangement graph 59% of the issued sizes are within 10% of the desired size, for the star graph the figure shows that 0% from the issued values are within 10% of the desired size. The figure gives a clear indication on the superiority of the arrangement-star graph over the graphs star and arrangement.

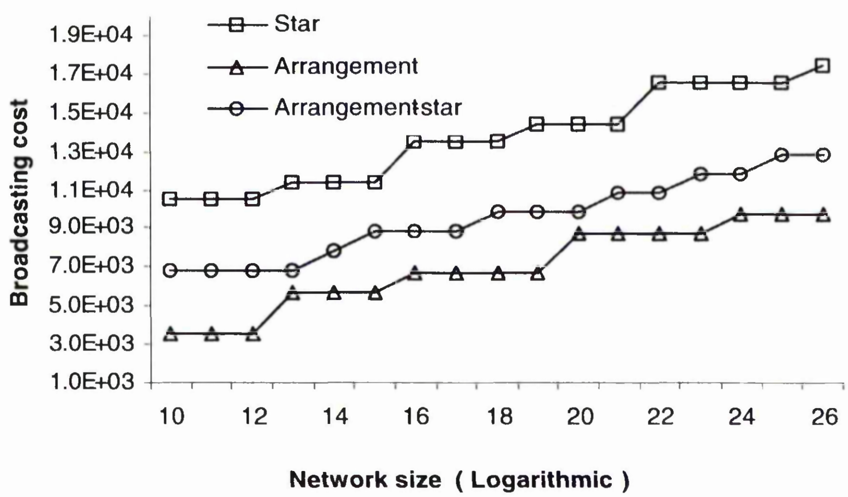


Figure 2.6: Comparing the broadcasting cost for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

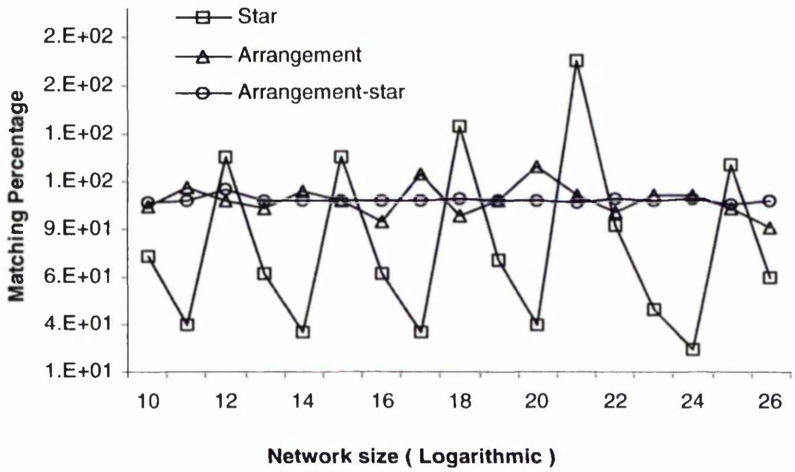


Figure 2.7: The Degree of accuracy for the three graphs, S_n, $A_{m,k}$ and $AS_{n,m,k}$.

2.6 Conclusions

This chapter has proposed a new interconnection network, called the *arrangement-star*, as the cross product of two existing networks the arrangement and star networks. Some general topological properties of the arrangement-star network have been discussed and proved. A comparative analysis has revealed that the proposed network possesses superior topological properties over its arrangement and star counterparts in terms of degree, diameter and more flexibility in choosing the desired network size and it has a smaller number of links and lower broadcasting cost.

The analysis presented above has also shown that 100% of different size topologies of the arrangement star are within 10% of the required size. This study has shown that the arrangement-star overcomes the scalability that both the star and arrangement networks suffer from.

The next chapter will propose new structural outlooks for developing efficient algorithms on the arrangement-star for an important class of applications based on grid and pipeline structures, demonstrating the ability of this new network to overcome the limitations of the star and arrangement at the algorithmic front

Chapter 3

Efficient Structural Outlooks for the Arrangement-star Network

3.1 Introduction

Having an effective structural outlook for an interconnection network greatly facilitates the design of efficient parallel algorithms on the network [10, 18, 19, 42]. Through the structural outlook of the network topology, the possible communication patterns are anticipated more easily when developing algorithms on the network. In this chapter, two different structural outlooks will be introduced namely; the grid structure and the pipeline structure for algorithm development on the arrangement-star network. The ability of the proposed structural outlooks for the arrangement-star network in handling broadcast communication using the proposed structural outlooks will be examined. The results presented in this chapter will add further evidence to the viability of the arrangement-star network as a potential high-performance interconnection network for future parallel computers.

The present study shows that the proposed structural outlooks enable the new network to provide efficient support for an important class of algorithms. This class of algorithms is based on grid and pipeline structures, and includes matrix decomposition [32], computing fast Fourier transforms [44], broadcasting [13], and sorting [10].

A desirable property when developing parallel algorithms for any interconnection network is the vertex symmetry, which allows the network topology to look the same from any of its vertices. This property facilitates the development of parallel algorithms on networks, such as our proposed arrangement-star, because it allows all the processors to be treated identically. We will show in this chapter that the arrangement-star network is vertex symmetric. Furthermore, the hierarchical structure of the arrangement-star topology, which is another important property that affects the design of routing algorithms for this network, will be also discussed and proved.

3.2 Background

The star graph has been a subject of a lot of studies in the past [10, 18, 19, 32, 44, 46]. This interest can be attributed to the desirable properties of this network [2, 3] over the existing and well-know topologies such as the hypercube. A number of issues related to the star graph have been investigated including its topological properties [3], broadcasting [45, 47], fault tolerance, [2], routing [48], sorting [48], and embedding [29, 50]. Furthermore, algorithms have already been reported in the literature for performing Fourier transforms [44], matrix decomposition [32], prefix-like operations [46], and ascend /descend type of divide-and-conquer problems [18, 19] on the star graph.

However, the study of algorithms for the star graph from the mentioned investigations is yet to mature [25]. The difficulty in developing efficient parallel algorithms on this network can be attributed to two main reasons: the first reason is the proposed static embedding of some popular topologies such as meshes [54] and hypercubes [55] into the star graph does not allow efficient simulation and mapping of known algorithms. The other reason is related to the fact that the structural outlooks for the star graph are very limited and unbalanced [53], and as a result they do not allow efficient algorithm development.

The grid structural outlook for the star graph is unbalanced because it generates column processing rather than grid processing [18,19]. For instance, the 10-star can viewed as 10×362880 or 90×40320 grid. Comparing this grid to the optimal square grid view of the hypercube, this grid is inefficient because it generates unbalanced message dimension between the rows and the columns of the obtained grid. This results in non-regular communication cost across the rows and the columns of the grid. Moreover, the structural outlook based on the pipeline structure for the star graph is also insufficient because it generates an excessive number of nodes in each stage of the pipeline. For instance, the number of nodes for each stage in the 10-stage pipeline is 362880, which can lead to high communication cost in each stage.

Day and Tripathi [28] have proposed the arrangement graph as a generalisation of the star graph in an attempt to solve the star graph scalability problem. However, since the introduction of the arrangement graph, there has been little work on the development of new algorithms for this network. This is due to the fact that the arrangement graph inherits the major difficulties from which the star graph suffers due to the lack of efficient structural outlooks that can take advantage of its attractive topological properties.

This chapter addresses the problem of developing algorithms that the star and arrangement networks suffer from by proposing efficient structural outlooks for algorithm development on the new arrangement-star network [7, 37]. This study will reveal that the algorithmic structural outlooks enable the new network to provide efficient support for an important class of parallel applications that are based on grid and pipeline structures.

The rest of the Chapter is organised as follows. Section 3 proves that the arrangement graph is vertex symmetric. Section 4 discusses and proves the hierarchical structure of the arrangement-star network. Section 5 develops algorithmic structural outlooks that allow the proposed network to support important class of parallel applications that are based on the grid and pipeline structures. Section 6 shows that the algorithmic structural outlooks enable the arrangement-star graph to outperform both the star graph and hypercube in terms of communication cost. Finally, Section 7 concludes this chapter.

3.3 Vertex Symmetry

Symmetry in graphs is a desirable property that makes the graph suitable for real applications such as solving matrix computations [32], Fourier transform [44], and ascend/descend type of divide-and-conquer algorithms [18, 19]. A graph is vertex symmetric if it looks the same from any of its vertices. A graph also is said to be edge symmetric if it looks the same from any of its edges. Vertex symmetry is a useful property that can ease the development and implementation of algorithms for a given graph topology. Akers and Krishnamurthy [1] have shown that the star graph is vertex symmetric. Moreover, Day and Tripathi [28] have proved that the arrangement graph is vertex symmetric. Since Al-Ayyoub and Day [4] have not addressed, when proposing their theoretical framework for product graphs, the issue of "vertex symmetry", we will show that the product of graphs preserves the vertex symmetry property. To do so, let us provide the following definitions that will be helpful in proving that the arrangement-star graph is vertex symmetric. Note that X is an address of a node in $\psi_{n,n-k}$.

***Definition 3.1*:** Let $\wp_n$ be the set of all permutations on $\langle n \rangle$. And $\psi_{n,k}$ be the set of all permutations of k symbols out of $\langle n \rangle$ symbols.

***Definition 3.2*:** Let $r: \langle n \rangle \times \psi_{n,k} \rightarrow \langle k \rangle$ be a ranking function given by $r(x_i, X) = y_i$ such that $y_i \in \langle k \rangle$ corresponds to the rank of x_i in X.

***Definition 3.3*:** Let Φ: $\psi_{n,k} \rightarrow \wp_k$ be a function that maps an arrangement of the form $X=x_i\ x_2 \ldots x_k$ into an permutations of the form $Y = y_1 y_2 \ldots y_k$ such that $r(x_i,X)= y_i$.

Proposition 3.1: The set of vertices in $AS_{n-k,n,k}$ is isomorphic to the set of vertices in S_n.

Proof: Recall that the set of vertices $AS_{n-k,n,k}$ are $V(AS_{n-k,n,k}) = \{\langle y,z \rangle \mid y \in \wp_n$ and $z \in \psi_{n,k}\}$, and the set of vertices in S_n are $V(S_n) = \{ X \mid X \in \wp_n \}$.

Let f : $V(S_n) \rightarrow V(AS_{n-k,n,k})$ be a function that maps vertices from $V(S_n)$ to $V(AS_{n-k,n,k})$. The function f is given by $f(X)= \langle \Phi(x_1\ x_2 \ldots x_{n-k}) x_{n-k+1} \ldots x_n \rangle$. Given any two distinct vertices $X= \langle x_1\ x_2 \ldots x_{n-k}\ x_{n-k+1} \ldots x_n \rangle$ and $W= \langle w_1\ w_2 \ldots w_{n-k}\ w_{n-k+1} \ldots w_n \rangle$ in $V(S_n)$, their images in $V(AS_{n-k,n,k})$ are $f(X) = \langle \Phi((x_1\ x_2 \ldots x_{n-k}), x_{n-k+1} \ldots x_n \rangle$ and $f(W) = \langle \Phi(w_1\ w_2 \ldots w_{n-k}), w_{n-k+1} \ldots w_n \rangle$. Next we will prove that this function is one-to-one. To do so, the following three cases need to be considered:

Case 1: If $X_{n-k+1} \ldots x_n = w_{n-k+1} \ldots w_n$ then $\Phi(x_1\ x_2 \ldots x_{n-k})$ and $\Phi(w_1\ w_2 \ldots w_{n-k})$ would be equal only if $X= W$, which contradicts the assumption.

Case 2: If $\Phi(x_1\ x_2 \ldots x_{n-k}) = \Phi(w_1\ w_2 \ldots w_{n-k})$ and the set of symbols $x_1\ x_2\ \ldots x_{n-k}$ is different than the set of symbols in $w_1\ w_2 \ldots w_{n-k}$ then $x_{n-k+1} \ldots x_n \neq w_{n-k+1} \ldots w_n$ and therefore $f(X) \neq f(W)$.

Case 3: If $\Phi(x_1\ x_2 \ldots x_{n-k}) = \Phi(w_1\ w_2 \ldots w_{n-k})$ and the set of symbols in $x_1\ x_2 \ldots x_{n-k}$ is the same as set of symbols in $w_1\ w_2 \ldots w_{n-k}$ then $X=W$ which contradicts the assumptions.

Also, since $|V(S_n)| = n! = (n-k)! \times n! / (n-k)! = |V(AS_{n-k,n,k})|$ and f is one-to-one, the claimed result follows.

Proposition 3.2: Given two vertex symmetric graphs G_1 and G_2, $G_1 \otimes G_2$ is also vertex symmetric.

Proof: Let G_1 and G_2 be two graphs such that $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. If f_1 and f_2 are two functions defined as follows f_1: $V_1 \rightarrow V_1$ and f_2: $V_2 \rightarrow V_2$. The graph $G = G_1 \otimes G_2$ is the cross product of the two graphs G_1 and G_2, where $G= (V, E)$, $V=\{\langle u, v\rangle \mid u \in V_1 \text{ and } v \in V_2\}$ *and* $E=\{(\langle x_1, y\rangle, \langle y_1, y\rangle) \mid (x_1, y_1) \in E_1\}$ or $\{(\langle x, x_2\rangle, \langle x, y_2\rangle) \mid (x_2, y_2) \in E_2\}$ [4]. The function f is defined as f: $V \rightarrow V$ such that $f(u, v)= \langle f_1(u), f_2(v)\rangle$, where $\langle u, v\rangle \in V$.

Firstly, we will prove that f is one-to-one. Let $u = \langle x_1, y_1\rangle$ and $v=\langle x_2, y_2\rangle$ such that $u, v \in V$ and $u \neq v$, we need to show that $f(u) \neq f(v)$, $f(u) = f(\langle x_1, y_1\rangle) = \langle f_1(x_1), f_2(y_1)\rangle$ and $f(v) = f(\langle x_2, y_2\rangle) = \langle f_1(x_2), f_2(y_2)\rangle$. We know from the above assumption that $f_1(x_1)=f_1(x_2)$ only if $x_1= x_2$. Similarly, $f_2(y_1)=f_2(y_2)$ only if $y_1= y_2$. Hence, for $f_1\langle (x_1), f_2(y_1)\rangle$ to be equal to $f_2\langle (x_2), f_2(y_2)\rangle$ then we should have $x_1= x_2$ and $y_1= y_2$, which contradicts our initial assumption $(u \neq v)$.

Secondly, we need to prove that for any two nodes u and v in V such that $(u, v) \in E$ then their images are also neighbours, i.e. $\langle f(u), f(v)\rangle \in E$. Let $u= \langle x_1, y_1\rangle$ and $v= \langle x_2, y_2 \rangle$ such that $(u, v) \in E$. Now consider $f(u)=f(\langle x_1, y_1\rangle)= \langle f_1(x_1), f_2(y_1)\rangle = f(v)=f(\langle x_2, y_2\rangle) = \langle f_1(x_2), f_2(y_2)\rangle$, here we have two cases:

Case 1: $x_1= x_2$ and $(y_1,y_2) \in E_2$. Since $f_1(x_1)=f_1(x_2)$, $(f_2(y_1), f_2(y_2)) \in E_2$. Hence $(u, v) \in E$.

Case 2: $y_1= y_2$ and $(x_1,x_2) \in E_1$, here we have $f_2(y_1)=f_2(y_2)$ and $((f_1(x_1), f_1(x_2)) \in E_1$. Hence $(u, v) \in E$. Hence $G_1 \otimes G_2$ is also vertex symmetry as claimed.

Both the star and arrangement graphs are vertex symmetric [1, 28]. By the above Proposition 3.2 we have shown that the cross product of two vertex symmetric graphs is also vertex symmetric. It follows that the arrangement-star graph is vertex symmetric.

3.4 Hierarchical Structure for the Arrangement-star Network

The hierarchical structure of a graph relates to the ability to build large graphs from smaller graphs of the same nature. Hierarchical graphs have attractive symmetry properties that are important in the design of routing algorithms and in constructing grid and pipeline structures [18, 42]. In this section, the hierarchical structure of the arrangement-star graph is discussed. The properties of a new decomposition method for this network are presented and proved. These properties are then used in the following sections to develop grids and pipelines as methods for developing various algorithms on the arrangement-star graph.

Proposition 3.3: The $AS_{n,m,k}$ can be decomposed into $\frac{n!}{(n-q)!}\ \frac{m!}{(m-p)!}$ disjoint copies of $AS_{n-q,m-p,k-p}$.

Proof: Recall that [53] S_n can be decomposed into $\dfrac{n!}{(n-q)!}$ disjoint copies of S_{n-q} while $A_{m,k}$ can be decomposed into $\dfrac{m!}{(m-p)!}$ [53] disjoint copies of $A_{m-p,k-p}$. Let X be a node in $AS_{n,m,k}$ and let q and p be two integers such that $1 \leq q < n$ and $1 \leq p < k$. Pick any q symbols out of $\langle n \rangle$ and fix them in the last q positions of $lp(X)$.

Similarly, pick any p symbols from $\langle m \rangle$ and fix them in the last p positions of $rp(X)$. Now, varying the remaining symbols in $lp(X)$ and the remaining symbols in $rp(X)$ will produce a new copy of $AS_{n-q,m-p,k-p}$ subgraph. For each new $(n\text{-}q)$-permutations there are $\binom{n}{q}$ different ways of choosing the q symbols from $\langle n \rangle$ and there are $q!$ ways of fixing these symbols in $lp(X)$. Also, for each new $(k\text{-}p)$-arrangement there is $\binom{m}{p}$ different ways of choosing p symbols from $\langle m \rangle$ and there is $p!$ ways of fixing these symbols in $rp(X)$. Hence, the number of disjoint copies that made is equal to $\dfrac{n!}{q!(n-q)!}q!\dfrac{m!}{p!(m-p)!}p! = \dfrac{n!}{(n-q)!}\dfrac{m!}{(m-p)!}$. Figure 3.1 Shows a node in $AS_{n-q,m-p,k-p}$ consisting of two parts $lp(X)$ and $rp(X)$.

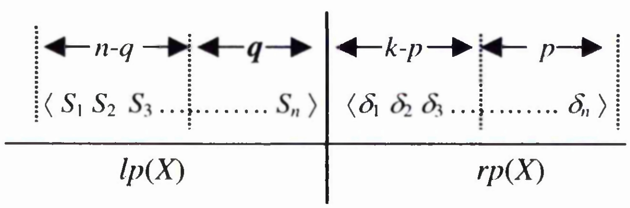


Figure 3.1: A node address in $AS_{n-q,m-p,k-p}$ where $S_1\ S_2 \ldots. S_n \in \wp_n$ and $\delta_1\ \delta_2 \ldots.. \delta_n \in \psi_{n,k}$.

Corollary 3.1: The $AS_{n,m,k}$ can be decomposed into $\dfrac{n!}{q!}\dfrac{m!}{(m-k+p)!}$ disjoint copies of $AS_{q,m-k+p,p}$.

From Proposition 3.3 and Corollary 3.1 it can be concluded that $AS_{n,m,k}$ can be decomposed into $\binom{n-1}{q}\frac{n!}{(n-q)!}\frac{m!}{(m-p)!}\binom{k}{p}$ non-disjoint copies of $AS_{n-q,m-p,k-p}$. The decomposing process here is similar to that in proposition 3.3, except there is an additional free parameter which it is the set of positions in lp(X) and rp(X). We can change all the positions in lp(X) except the first one. So, we will have $\binom{n-1}{q}$ different ways of setting the positions. Furthermore, we can change all the positions in rp(X), hence we will have $\binom{k}{p}$ different ways of set positions. Hence, the $AS_{n,m,k}$ can be decomposed into $\binom{n-1}{q}\frac{n!}{(n-q)!}\frac{m!}{(m-p)!}\binom{k}{p}$ non-disjoint copies of $AS_{n-q,m-p,k-p}$.

3.5 Developing Efficient Algorithms for the Arrangement-star Network

Among the well studied structural outlooks that provide a structural view of a given network are the grid and pipeline structures as they allow efficient simulation and mapping of a wide class of parallel algorithms [19, 32, 44]. In an attempt to address the problem of designing algorithms for the star and arrangement networks we proposed the arrangement-star network. In this section we develop efficient structural outlooks for algorithm design on the proposed network. These structural outlooks can be used to design a wide class of parallel algorithms on the arrangement-star network such as matrix computation problems, Fourier transforms and broadcasting. Moreover, well-known topologies, such as meshes and tori, can be embedded in the arrangement-star network using these outlooks. Two structural outlooks, grid and pipeline, are developed in the following 2 subsections. Timing models for measuring the performance of the proposed structural outlooks are also provided. Through these models, the performance of various algorithms on the proposed network based on grid and pipeline structures are evaluated and compared with their counterparts such as star and hypercube networks. In this comparison, we have excluded the arrangement network due to the extreme difficulty in proposing algorithmic structural outlooks for this network.

3.5.1 The Grid Structural Outlook

For any network G that can be factored into two sub-networks G_1 and G_2, we can obtain a 2-dimensional grid structure from G, where rows and columns in this structure are respectively G_1-connected and G_2-connected or vice versa. A few structural outlooks have been proposed for the star graph, and the most well studied outlooks have been introduced in [10, 46]. Menn and Somani [10] have shown that the star graph can be viewed as $n \times (n\text{-}1)!$, where the rows and the columns in this grid are $(n\text{-}1)$-star and an n-linear array, respectively. Furthermore, Ferreria and Berthome [46] have shown that the star graph can be viewed as a rectangular grid $R \times C$ (Rows by Columns) where the rows are substar-S_{n-2} and the columns are $n(n\text{-}1)$ nodes on each column. However, these structural outlooks are insufficient for developing efficient algorithms on the star graph. For the star graph, the grid structural outlook is unbalanced because it resembles column processing rather than grid processing. For instance, the 10-star can be viewed as 10×362880 or 90×40320 grid. Compared to the optimal square grid view of the hypercube, this grid is inefficient as it generates unbalanced message dimension between the rows and columns of the grid structural outlook. Consequently, the communication cost between the rows and columns of the grid is not regular.

The decomposition of the arrangement-star graph as a two-dimensional grid structure can be achieved by arranging the nodes of $AS_{n-q,m-p,k-p}$ on the rows and columns of the grid where $AS_{n-q,m-p,k-p}$ embeds R× C grid where R *is* S_{n-q}–subgraph and C *is* $A_{m-p,k-p}$ –subgraph. In this grid each columns has $(m\text{-}p)!/(m\text{-}k)!$ nodes and the number of nodes assigned on each row is $(n\text{-}q)!$. Another way of viewing $AS_{q,m-k+p,p}$ as grid structure can be achieved since $AS_{q,m-k+p,p}$ embeds $R \times C$ grid where R equal to $q!$ and C equal to $(m\text{-}k\text{+}p)!/(m\text{-}k)!$. In this grid Columns are S_q-connected and rows are $A_{m-k+p,p}$-connected.

As a particular case $AS_{n,m,k}$ embeds $n!\cdot m!/(m\text{-}k)!$ grid where columns are S_n-connected and rows are $A_{m,k}$-connected. It is clear that $AS_{n,m,k}$ can be viewed as two orthogonal partitioning, $R^X \times C^Y$ where, R^X = $n!$-subgraphs S_1, $1 \leq X \leq n!$ and C^Y = $m!/(m\text{-}k)!$-subgraphs $A_{1,1}$. $1 \leq Y \leq m!/(m\text{-}k)!$ [4]. This partition is a particular case of $AS_{n,m,k}$ when the value of q and p are equal to n-1 and k respectively. The important issue of the above two orthogonal partitioning is that the vertices of each subgraph in the first partitioning are contained in the second subgraph, one vertex per subgraph. In addition, the vertices of each subgraph in the second partitioning are contained in the first subgraph one vertex per subgraph. This decomposition will facilitate the broadcasting of data from one node to another in different rows and columns of the graph. The above decomposition of $AS_{n,m,k}$ can be the basis for developing algorithms on the arrangement-star specially the algorithms, which they need the matrices form. The rows of the matrix can be distributed on the $n!$ subgraphs S_1 where each of these sub-graphs has a vertex in each $m!/(m\text{-}k)!$ subgraphs $A_{1,1}$ one vertex per each subgraph. This will yield broadcasting on rows and columns of the arrangement-star graph.

3.5.2 The Pipeline Structural Outlook

The pipeline structure is another well-known structural outlook that is suitable for real applications [18, 19]. It is know from the literature that the star graph can be structured according to a pipelined view [18, 19] where the n-star can be arranged as a sequence of $(n\text{-}1)$-stars forming an n-stage pipeline. The authors in [18, 19] have shown that when the nodes of an n-star are ranked using the pipeline structured ranking then there exists a path from any node in ${}^{i}S_{n-1}$ to its peer node in ${}^{i+1}S_{n-1}$ or ${}^{i-}$

${}^{1}S_{n-1}$, where $1 \leq i \leq n$. The length of this path is either one move or at most three moves [18, 19]. However, the structural outlook based on the pipeline view for the star graph is insufficient as it generates a large number of nodes in each pipeline stage.

The arrangement-star graph possesses a structural outlook that provides a pipelined structure in a more balanced manner. From the literature [31, 50], we know that both the arrangement and star graphs are Hamiltonian. Therefore, the pipeline structure can be issued for the arrangement-star graph in two different ways: by the graph $A_{m,k}$ and the Hamiltonian path S_n or vice versa [4]. We can have full control over the number of stages and the size of each stage by tuning the parameters n, m, k.

3.6 Performance Comparison Using the Proposed Structural Outlooks

This section conducts a comparative study between the three graphs star, hypercube and arrangement-star. The comparison uses the cost of broadcasting across the row and the column for the grid structural outlook as a performance measure. Moreover, it uses the broadcasting cost in one stage plus the cost of shifting the data to the next stage for the pipeline structure as a performance measure. The results will be reported for different network sizes and a fixed message of length M =1024 byte [45].

For the grid structure, we estimate the communication cost of broadcasting across a row plus the communication cost across a column [32, 45]. To estimate the broadcasting in both directions. We use the lower bound formula that has been extensively used in existing similar studies [25, 32, 42, 45, 53]:

$$\left(\sqrt{\frac{M.a}{\beta\Delta}} + \sqrt{\delta - 1}\right)^2 \beta .$$

The parameters Δ and δ are the degree and diameter of the graph, respectively and the symbols M, a, and β are the message length, unit transmission cost and the message latency cost. The values of these parameters are set to 1024 byte, 1 μs and 1000 μs, respectively [45]. Figure 3.2 plots the obtained communication cost against the network size (in Logarithmic scale) for the three graphs: star, hypercube and arrangement-star; recall that we excluded the arrangement graph in this comparison because there is no any algorithmic framework for this graph. Figure 3.2 shows clearly that the proposed arrangement-star network outperforms both the star and hypercube in terms of communication cost.

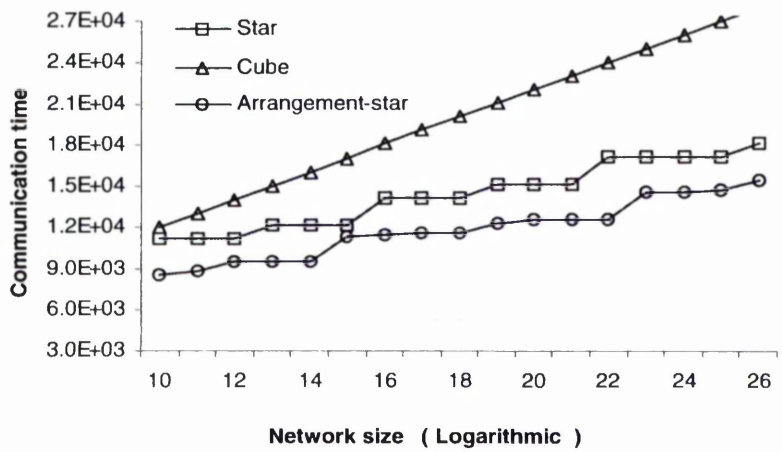


Figure 3.2: The communication cost based on grid view.

The broadcasting cost based on the pipeline structure can be estimated in two different ways; by shifting the data through the Hamiltonian cycle of $A_{m,k}$ or by shifting the data through the Hamiltonian cycle of S_n. The lower bound of the broadcasting cost in the pipelined structure is equal to the lower bound of the broadcasting cost in the arrangement graph across one stage plus the cost of shifting the data to the next stage or vice versa [53]. The cost of shifting the data to the next stage is equal to $\delta(\beta + Ma)$, where the parameters M, β and a are as defined above [32].

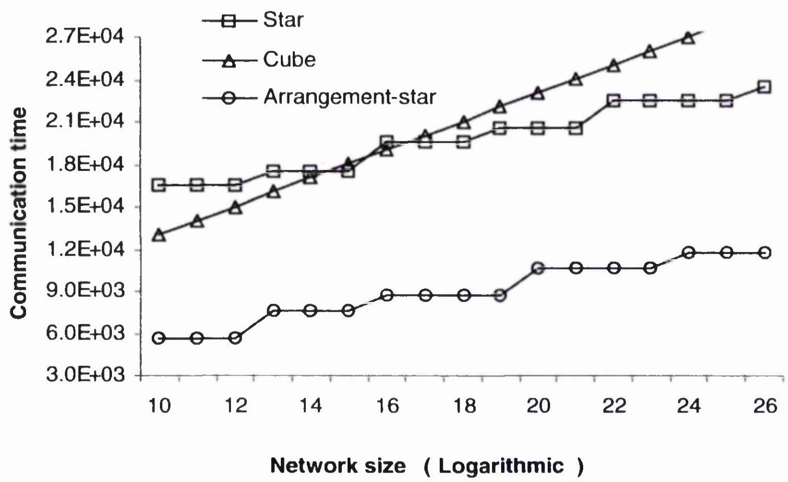


Figure 3.3: The communication cost based on pipeline view.

Figure 3.3 plots the estimated communication cost for the graphs star, hypercube and arrangement-star against the network size in a Logarithmic scale. The figure reveals that the communication cost based on the pipeline structure in the arrangement-star graph is lower than that in the star and hypercube.

From the results shown in both Figures 3.2 and 3.3 it can be concluded that the proposed graph is more suitable than the star and hypercube for real parallel applications as it exhibits the lowest communication cost.

3.7 Conclusions

In this chapter, we have proved that the new arrangement-star network is vertex symmetric and possess a hierarchical structure. These are desirable properties that make the proposed network suitable for real applications. We have also shown that the arrangement-star network overcomes the major limitations of both the star and arrangement networks on the algorithmic front by proposing two different structural outlooks for developing parallel algorithms on the new network, based on the well-known grid and pipeline structures.

A comparative study against the star and hypercube networks has been conducted. Performance results based on the cost of broadcast communication using the structural outlooks have revealed that the proposed network has a lower communication cost than both the star and hypercube, and therefore is more suitable for supporting real-time applications. This finding demonstrates the suitability of the arrangement-star network as a high-performance network for future parallel computers.

Chapter 4

A Case Study on Vertex Product Networks: OTIS-Networks

4.1 Introduction

The choice of the network topology for a high-speed parallel computer is an important design decision that involves inherent trade-offs in terms of efficient algorithm support and network implementation cost. For instance, networks with large bisection width allow fast and reliable communication. However, such networks are difficult to implement using today's electronic technologies that are two dimensional in nature [57]. In principle, free-space optical technologies offer several fronts to improve this trade-off. The improved transmission rate, power consumption, and signal interference are few examples on these fronts [20, 57, 58, 59].

Optoelectronic and optical networking will become the key enabling technologies of the future communications infrastructure through the elimination of the difficult limitation of bandwidth and bit-error rate inherent in traditional electromagnetic signal-based communications [34, 35]. Electromagnetic signals carried over copper (or coaxial) wires suffer from loss of strength and are subject to errors due to noise and hence such systems have limited data rates [60]. When copper or coax is replaced by fiber technology the achievable bandwidth is in excess of 50 terabits/second with an almost zero bit-error rate [60]. The full implications of essentially huge bandwidth and extremely low loss rates are only beginning to be recognised and will radically reshape the future network technologies. While in the past the communication link was the bottleneck, this link now holds the potential to become the enabler of new modes of computing far beyond those existing today [60].

4.2 Optical Transpose Interconnection Systems (OTIS)

This study focuses on a specific optical interconnect, namely the *Optical Transpose Interconnection Systems* (OTIS), as a case of study on vertex product networks. Marsden *et al* were the first to propose the OTIS [34]. A number of computer architectures have subsequently been proposed in which the OTIS were used to connect different processors [34]. Krishnamoorthy *et al* [59] have shown that the power consumption is minimised and the bandwidth rate is maximised when the OTIS computer is partitioned into N groups of N processors each. Zane *et al* [36] have limited their study to this type of the OTIS. In our present study, we will focus on OTIS-networks where the number of processors in each group is equal to the number of groups; the terms OTIS-computer and OTIS-network refer to parallel architectures based on the OTIS and will be used interchangeably in the rest of the thesis.

OTIS-networks are implemented using free-space optoelectronic technology [34]. In this model, processors are partitioned into groups, where each group is realised on a separate chip with electronic inter-processor connects. Processors on separate chips are interconnected through free space interconnects. The philosophy behind this separation is to utilise the benefits of both the optical and electronic technologies.

The advantage of using the OTIS as an optoelectronic architecture lies in its ability to manoeuvre the fact that free space optical communication is superior in terms of speed and power consumption when the connection distance is more than few millimetres [59]. In the OTIS, shorter (intra-chip) communication is realised by electronic interconnects while longer (inter-chip) communication is realised by free space interconnects.

Extensive modelling results for the OTIS have been reported in [20]. The achievable terabit throughput at a reasonable cost makes the OTIS a strong competitor to the electronic alternatives [34, 59]. These encouraging findings prompt the need for further testing of the suitability of the OTIS for real-world parallel applications. A number of recent studies have been conducted in this direction [15, 24, 38, 41, 61, 62]. Sahni and Wang [38, 41] have presented and evaluated various algorithms on the OTIS-mesh and OTIS-hypercube including basic data rearrangements, routing, selection and sorting. They have also developed algorithms for various matrix multiplication operations [62, 72] and image processing [24]. Zane *et al* [36] have shown that the OTIS-mesh (which will be discussed below) efficiently embeds four-dimensional meshes.

By using different electronic topologies, we arrive at different classes of OTIS-networks. For Instance, the OTIS-mesh and OTIS-hypercube are two different classes of the OTIS-networks where the mesh and hypercube topologies are used to realise the electronic interconnect. In this thesis, two well-studied OTIS-networks will be used in our performance comparison along with their electronic counterpart networks, the mesh and hypercube. A brief description of the OTIS-mesh and OTIS-hypercube along with some of their topological properties is discussed in the following two sections.

4.3 The OTIS-mesh and OTIS-hypercube

The OTIS-mesh consists of an N^2 processor; there are N groups and each group forming $\sqrt{N} \times \sqrt{N}$ mesh. In the OTIS-mesh, processors in the same group are connected as two-dimensional mesh [15, 34, 36]. Figure 4.1 shows a 16 processor OTIS-mesh, the notation $\langle g, p \rangle$ is used to refer to the group and processor addresses, respectively. Two nodes $\langle g_1, p_1 \rangle$ and $\langle g_2, p_2 \rangle$ are connected if, and only if, $g_1 = g_2$ and $(p_1, p_2) \in E_0$ (such that E_0 is the set of edges in factor network) or $g_1 = p_2$ and $p_1 = g_2$ and in this case the tow nodes are connected by transpose edges.

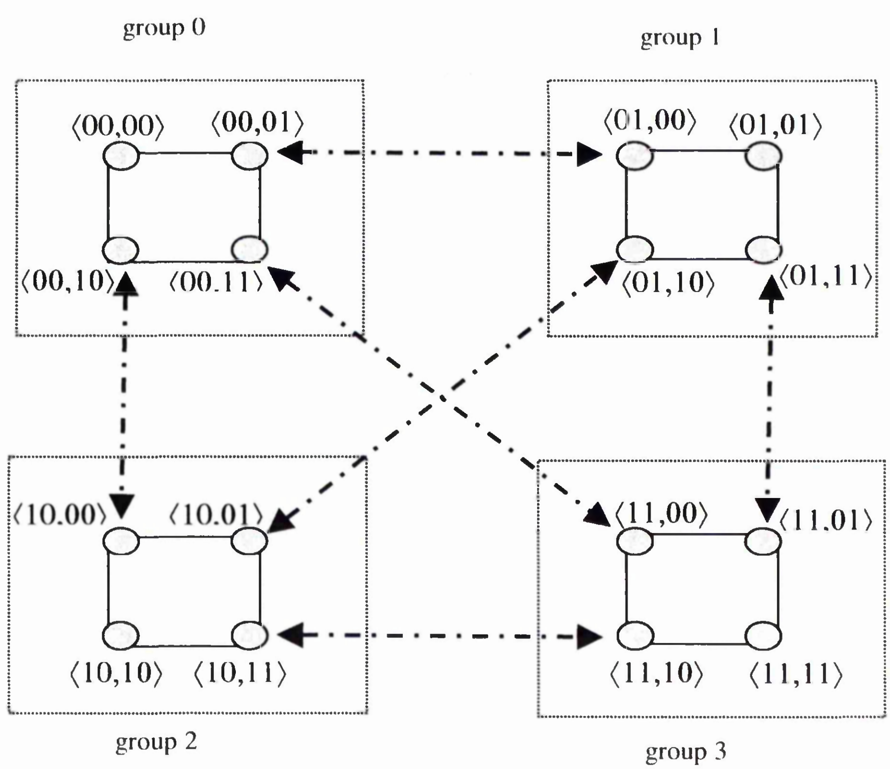


Figure 4.1: The OTIS-mesh with 16 processors.

The distance in the OTIS-mesh is defined as the shortest path between any two processors, $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$, and involves one of the following forms [15]:

1- When $g_1 = g_2$ then the path involves only one electronic moves. ...(*a*)

2- When $g_1 \neq g_2$ and if the number of optical moves is an even number of moves and more than two, then the paths can be compressed into shorter path of the form:

$$\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, p_2\rangle \xrightarrow{O} \langle p_2, g_1\rangle \xrightarrow{E} \langle p_2, g_2\rangle \xrightarrow{O} \langle g_2, p_2\rangle . \qquad ...(b)$$

Where the symbols O and E stand for optical and electronic moves respectively.

3- When $g_1 \neq g_2$, the path involves an odd number of OTIS moves. In this case the paths can be compressed into a shorter path of the form:

$$\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, g_2\rangle \xrightarrow{O} \langle g_2, g_1\rangle \xrightarrow{E} \langle g_2, p_2\rangle . \qquad ...(c)$$

Theorem 4.1: The length of the shortest path between any two processors $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ in OTIS-mesh is $d(p_1, p_2)$ when $g_1 = g_2$ and $\min\{d(p_1, p_2) + d(g_1, g_2) +2, d(p_1, g_2) + d(g_1, p_2) + 1\}$ when $g_1 \neq g_2$, where $d(p, g)$ stands for the shortest distance between the two processors p and g using any path a, b and c [15].

It is obvious from the above theorem that when $g_1 = g_2$, then the length of the path between the two processors $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ is $d(p_1, p_2)$. From the path construction methods in (*b*) and (*c*) above, it can be easily verified that the length of the path equal min $\{d(p_1, p_2) + d(g_1, g_2) +2, d(p_1, g_2) + d(g_1, p_2) + 1$ when $g_1 \neq g_2\}$.

Since each group in the OTIS-mesh is an $\sqrt{N} \times \sqrt{N}$ mesh, all the distances $d\langle p_1, p_2\rangle$, $d\langle p_2, g_1\rangle$, $d\langle p_1, g_2\rangle$ and $d\langle g_1, g_2\rangle \leq 2(\sqrt{N} -1)$. It follows that any two processors in the OTIS-mesh are at a distance at most $4(\sqrt{N} -1) +1$ $(= 4\sqrt{N} -3)$, which is the diameter of the OTIS-mesh [63].

The OTIS-hypercube is another example of OTIS-networks. In this class of networks the electronic interconnect follows the hypercube topology. The OTIS-hypercube consists of N^2 processors and N groups, each group is a hypercube of dimension $\log_2 N$ and of size equal $N=2^h$ (where h is the degree of the hypercube). Figure 4.2 shows the topology of the OTIS-hypercube with 4 groups where each group consists of 4 processors. Two nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ are connected if, and only if, $g_1 = g_2$ and $(p_1, p_2) \in E_0$ (such that E_0 is the set of edges in the factor network) or $g_1 = p_2$ and $p_1 = g_2$ and in this case the tow nodes are connected by transpose edges [41].

The length of the shortest path from processor i to the processor j in the hypercube is defined as $d(i, j)$. If $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ are any two processors of the OTIS-hypercube, then the shortest path between these two processors follow the same path construction method as described above, i.e paths of type a, b or c.

Theorem 4.2: The diameter of the OTIS-hypercube is $2h+1$ where h is the dimension of the hypercube [41].

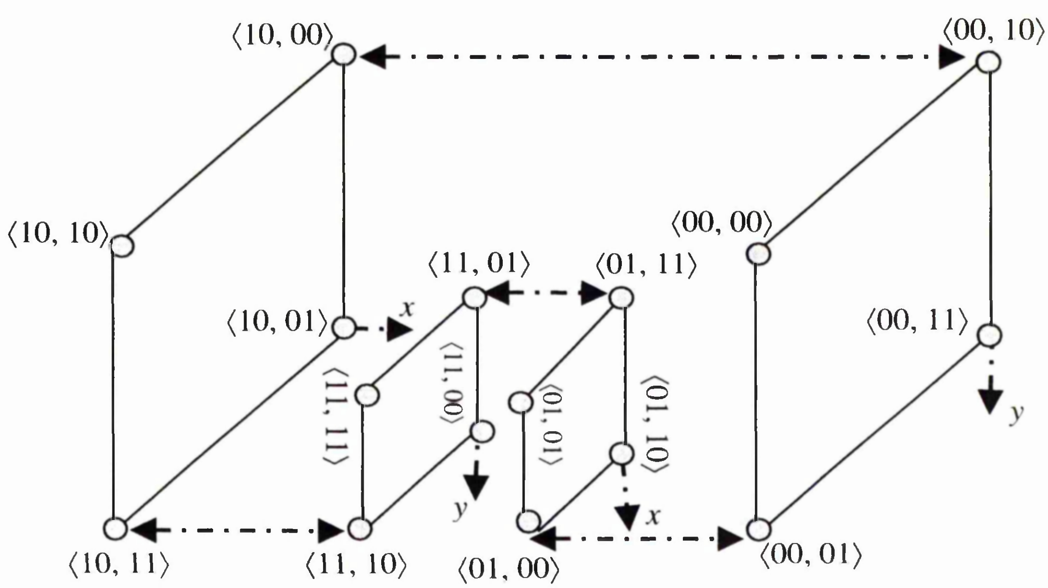


Figure 4.2: The OTIS-hypercube with 16 processors.

4.4 The Topological Properties of the OTIS-networks

This section discusses and derives some of the basic topological properties of the OTIS-networks including size, degree, diameter, number of links, broadcasting, and routing. Table 4.1 summarises the topological properties of the OTIS-star, OTIS-hypercube and OTIS-mesh along with their electronic counterparts star, hypercube and mesh. The symbols g and p in the table stands for the group and processor addresses respectively. Let us now discuss and derive the properties that are itemised in Table 4.1.

***Size*:** If G_0 is the factor network of size N, then the size of the OTIS-G_0 is N^2.

***Degree*:** Let $\langle g, p \rangle$ be any node in OTIS-G_0. Then the degree (or *deg*) of the OTIS-G_0 is as follows:

$$deg_{\text{OTIS-G}_0}(g, p) = \begin{cases} deg_{G_0}(p) & \text{if } g = p \\ deg_{G_0}(p) + 1 & \text{if } g \neq p \end{cases}$$

***Number of Links*:** Let N_0 be the number of links in the factor network and let M be the number of nodes in the G_0. The number of links in the OTIS-G_0 = $(M^2 - M)/2 + N_0 \cdot M$. For instance, the number of links in the OTIS-mesh consisting of 16 processors is $(4^2 - 4)/2 + 4 \cdot 4 = 12/2 + 16 = 22$.

***Diameter*:** Let $\langle g_1, p_1 \rangle$ and $\langle g_2, p_2 \rangle$ be two different processors in the OTIS-G_0. To transmit data originated in the source node $\langle g_1, p_1 \rangle$ to the destination node $\langle g_2, p_2 \rangle$ there are 3 possible paths to follow: a, b and c. These paths are in the following forms:

1- When $g_1 = g_2$, then the path employs only electronic moves.

Path *a*: $\langle g, p_1\rangle \xrightarrow{E} \langle g, p_2\rangle$

2- When $g_1 \neq g_2$ then the path employs 2 electronic moves and 2 optical moves.

Path *b*: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, p_2\rangle \xrightarrow{O} \langle p_2, g_1\rangle \xrightarrow{E} \langle p_2, g_2\rangle \xrightarrow{O} \langle g_2, p_2\rangle$.

3- Or when $g_1 \neq g_2$ then the path employs 2 electronic moves and one optical move.

Path *c*: $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, g_2\rangle \xrightarrow{O} \langle g_2, g_1\rangle \xrightarrow{E} \langle g_2, p_2\rangle$.

To send a message *M* from the source node $\langle g_1, p_1\rangle$ to the destination node $\langle g_2, p_2\rangle$ it must follow a route along one of the three possible paths *a, b,* and *c*. The length of the shortest path between the nodes $\langle g_1, p_1\rangle$ and $\langle g_2, p_2\rangle$ is one of the forms:

$$\text{Length} = \begin{cases} d(p_1, p_2) & \text{if } g_1 = g_2 \\ min(\, d(p_1, g_2) + d(g_1, p_2) + 1,\ d(p_1, p_2) + d(g_1, g_2) + 2) & o.w. \end{cases} \qquad \text{...(I)}$$

Where $d(p_1, p_2)$ is the length of the shortest path between any two processors $\langle g_1, p_1\rangle$ and $\langle g_1, p_2\rangle$. If δ_0 is the diameter of the factor network, G_0 then from (1) it follows that the diameter of the OTIS-G_0 is $2\delta_0 + 1$.

***Theorem 4.3*:** The diameter of OTIS-G_0 is the *Max* $(\delta_0, 2\delta_0 + 1)$ which is equal to $2\delta_0 + 1$.

The proof of the above theorem is a direct result from (I).

***Routing*:** is another important issue when considering a new interconnection network [28]. We assume that a routing algorithm in the factor network, G_0, is known and given with a function $Next_{G_0}(p_1, p_2)$; this function returns the forwarding node after p_1 in routing towards p_2. Also a function *Dist.* $_{G_0}(p_1, p_2)$ returns the minimum distance between the two processors $p_1$and p_2 in G_0. The message (*M*) is originated at the source node $\langle g_1, p_1\rangle$ and to be transmitted to the destination node $\langle g_3, p_3\rangle$. Figure 4.3 summarises the routing algorithm in the OTIS-G_0.

In Figure 4.3 the parameter M, stand for the message to be transmitted from the source node, while the parameters g_1, g_2, and g_3 are the source group, current group, and the destination group, respectively. Furthermore the parameters p_1, p_2, and p_3 stand for the source processor, current processor and the destination processor. The node $\langle g_1, p_1\rangle$ is defined to be the source node, $\langle g_2, p_2\rangle$, is the current node where the message has arrived and $\langle g_3, p_3\rangle$ is the destination node.

Network	Size	Degree	Diameter	Number of links
Star, S_n	$n!$	n-1	$\lfloor \frac{3}{2}(n-1) \rfloor$	$n!\cdot(n-1)/2$
Hypercube, Q_n	2^n	N	N	$(2^n\cdot n)/2$
Mesh(m,n)	$m\cdot n$	4	$2(\sqrt{n.m}-1)$	$(2mn - m - n)$
OTIS-star	$(n!)^2$	n-1 $g=p$ n $g\neq p$	$2\lfloor \frac{3}{2}(n-1)\rfloor+1$	$(n!\cdot(n-1)/2).n!+$ $((n!)^2-n!)/2$
OTIS-hypercube	$(2^n)^2$	N $g=p$ $n+1$ $g\neq p$	$2n+1$	$((2^n\cdot n)/2).2^n +$ $((2^n)^2 - 2^n)/2$
OTIS-mesh	$(m\cdot n)^2$	4 $g=p$ 5 $g\neq p$	$4\sqrt{n.m}-3$	$2\cdot(m\cdot n)^2$ $+((m\cdot n)^2- m\cdot n)/2$

Table 4.1: The topological properties of the star, cube and mesh and their OTIS counterparts.

When routing between the source node $\langle g_1, p_1\rangle$ and destination node $\langle g_3, p_3\rangle$, the distance between the source and destination nodes is in one of the forms: -

- If $g_1 = g_3$ then minimum distance between the source node and the destination node is $d = Dist_{G_0}(p_1, p_3)$. and the source node will call the function $Route_{\text{OTIS-}G_0}(M, \langle g_1, p_1\rangle, \langle g_2, p_2\rangle, \langle g_3, p_3\rangle, a)$ to route the message.

- If $g_1 \neq g_3$ then $d_1 = Dist_{G_0}(p_1, p_3) + Dist_{G_0}(g_1, g_3) + 2$ or $d_2 = Dist_{G_0}(p_1, g_3) + dist_{G_0}(g_1, p_3) + 1$.

  If $d_1 < d_2$ then the source node will call the function

  $Route_{\text{OTIS-}G_0}(M, \langle g_1, p_1\rangle, \langle g_2, p_2\rangle, \langle g_3, p_3\rangle, b)$ else

  $Route_{\text{OTIS-}G_0}(M, \langle g_1, p_1\rangle, \langle g_2, p_2\rangle, \langle g_3, p_3\rangle, c)$.

Function $Route_{\text{OTIS-}G_0}(M, \langle g_1, p_1\rangle, \langle g_2, p_2\rangle, \langle g_3, p_3\rangle, \text{path type})$

Begin **{function}**

if $\langle g_2, p_2\rangle \neq \langle g_3, p_3\rangle$ then

{where $\langle g_2, p_2\rangle$ is the current node and $\langle g_3, p_3\rangle$ is the destination node**}**

if path = a then send $(M, \langle g_1, p_1\rangle, Next_{G_0}(p_2, p_3), \langle g_3, p_3\rangle, a)$

else if path = b then

{path ***b*****:** $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, p_2\rangle \xrightarrow{O} \langle p_2, g_1\rangle \xrightarrow{E} \langle p_2, g_2\rangle \xrightarrow{O} \langle g_2, p_2\rangle$**}**

if $g_2 = g_1$ then

if $p_2 \neq p_3$ send $(M, \langle g_1, p_1\rangle, Next_{G_0}(p_2, p_3), \langle g_3, p_3\rangle, b)$

else send $(M, \langle g_1, p_1\rangle, \langle p_3, g_2\rangle, \langle g_3, p_3\rangle, b)$

else

if $g_2 \neq g_3$ then send $(M, \langle g_1, p_1\rangle, Next_{G_0}(g_2, g_3), \langle g_3, p_3\rangle, b)$

else send $(M, \langle g_1, p_1\rangle, \langle g_3, p_3\rangle, \langle g_3, p_3\rangle, b)$

else if path = c then **{path** ***c*****:** $\langle g_1, p_1\rangle \xrightarrow{E} \langle g_1, g_2\rangle \xrightarrow{O} \langle g_2, g_1\rangle \xrightarrow{E} \langle g_2, p_2\rangle$**}**

if $g_2 = g_1$ then

if $p_2 \neq g_3$ then send $(M, \langle g_1, p_1\rangle, Next_{G_0}(p_2, g_3), \langle p_3, g_3\rangle, c)$

else send $(M, \langle g_1, p_1\rangle, \langle g_3, g_2\rangle, \langle g_3, p_3\rangle, c)$ else

if $g_2 \neq p_3$ then send $(M, \langle g_1, p_1\rangle, Next_{G_0}(g_2, p_3), \langle g_3, p_3\rangle, c)$ else

stop, destination reached

End **{function}**.

Figure 4.3: The routing algorithm in the OTIS-G_0 using one of the paths *a*, *b* or *c*

Broadcasting: Data broadcasting is one of the most important operations required by many parallel applications [28]. Assume that the data is initially in the source node $\langle g_1, p_1\rangle$ where g_1 is the group index and p_1 is the processor index, and is to be broadcasted to all of the nodes N^2 in the OTIS-G_0. The data can be transmitted to all processors in the different groups using the following algorithm steps:

Step 1: The source processor $\langle g_1, p_1\rangle$ broadcasts the data to all the processors in its group using electronic moves.

Step 2: Each processor in the group then broadcasts the data to one processor in each group in the network using one OTIS move.

Step 3: Each processor in the group broadcasts the data again to the remaining processors using electronic moves.

In the above algorithm, following Step 1 each processor of the group containing the source processor will receive a copy of the data from the source processor using electronic move. After Step 2, one processor from each group in the network will receive a copy using OTIS move. In Step 3, each processor of the OTIS-G_0 will obtain a copy of the data using electronic moves. Step 1 and 3 will take one electronic move and Step 2 will take one OTIS move. We can also conclude that the cost of broadcasting in the OTIS-G_0 is $2 \times \beta_{G_0} + 1$ (*i.e.* two electronic moves plus one optical move) where β_{G_0} is the cost of broadcasting in G_0.

4.5 Conclusions

This chapter has presented OTIS-networks and their two well-studied versions, notably the OTIS-mesh and OTIS-hypercube. Some of the basic topological properties such as the degree and diameter of the OTIS-mesh and OTIS-hypercube have been discussed. Moreover, some of the general properties of OTIS-networks have been derived including the size, degree, diameter, and number of links. Algorithms for broadcasting and routing in these networks have also been presented.

Some of discussed and derived basic topological properties in this chapter will be used in the next two chapters for proposing efficient structural outlooks for algorithm development on OTIS-networks.

Chapter 5

Efficient Structural Outlooks for Algorithm Development on OTIS-Networks

5.1 Introduction

A number of recent studies have revealed that *Optical Transpose Interconnection Systems* (OTIS) are promising candidates for future parallel computers. While existing studies have proposed algorithms for specific problems [38, 41, 62], this chapter presents two structural outlooks for algorithm development on the OTIS. The proposed outlooks are general in the sense that no specific factor network or problem domain is assumed. They also allow efficient mapping of a wide class of algorithms into the OTIS, and are based on grids and pipelines as popular structures that support vast body of applications including linear algebra, divide-and-conquer algorithms, sorting, and Fourier transforms. Timing models for measuring the performance of the proposed outlooks are also provided. These models allow the performance of various algorithms on the OTIS to be evaluated and compared with their counterparts on conventional electronic interconnection systems.

5.2 Notation and Definitions

The Optical Transpose Interconnection Systems (OTIS) are implemented using free-space optoelectronic technologies [34]. Processors are partitioned into groups where each group is realised on separate chip with electronic inter-processor connects, processors on separate chips are interconnected through free space interconnects. The philosophy behind this separation is to utilise the benefits of both the optical and the electronic technologies.

Recently there has been an increasing interest in the study of parallel algorithms for optoelectronic networks in general and for OTIS-networks in particular [15, 24, 38, 41, 61, 62]. Zane et al [36] have shown that the OTIS-mesh efficiently embeds four-dimensional meshes and hypercubes. Sahni and Wang [38, 41] have presented and evaluated various algorithms on specific OTIS-networks (OTIS-mesh) including basic data rearrangements, routing, selection and sorting. They have also developed algorithms for various matrix multiplication operations [62] and image processing [24].

Aside from the above-mentioned works, the study of algorithms on the OTIS has not yet fully matured. In this chapter, we contribute towards filling this gap by presenting two structural outlooks for developing algorithms on the OTIS. The structures are based on grids and pipelines, which support a large number of applications ranging from linear algebra to divide-and-conquer type of algorithms, sorting, and Fourier transforms. We also show that for some factor networks, such as the mesh and hypercube, the performance of the proposed structural outlooks outperforms the best-known results for these factor networks. The proposed outlooks are discussed in the sequel, but first we give the necessary notation and definitions.

An OTIS-based computer contains N^2 processors partitioned into N groups with N processors each. A processor is indexed by a pair $\langle x, y\rangle$, $0 \le x, y < N$, where x is the group index and y is the processor index. Processors within a group are connected by certain interconnecting topology while transposing group and processor indexes achieve inter-group links. For example a 16 processor OTIS-mesh, in this example, intra-group connects is mesh-based, hence the term OTIS-mesh is used to denote this network. In what follows, the terms OTIS-computer and OTIS-networks refer to parallel architectures based on OTIS and will be used interchangeably.

OTIS-networks are basically constructed by "multiplying" a known topology by itself. The vertices set is equal to the Cartesian product on the vertices set in the factor network. The edges set consists of edges from the factor network and new edges called the *transpose edges*. The formal definition of OTIS-networks is given below.

Definition 5.1: Let $G_0 = (V_0, E_0)$ be an undirected graph representing a factor network. The OTIS-$G_0 = (V, E)$ network is represented by an undirected graph obtained from G_0 as follows $V = \{\langle x, y\rangle \mid x, y \in V_0\}$ and $E = \{(\langle x, y\rangle, \langle x, z\rangle) \mid \text{if } (y, z) \in E_0\} \cup \{(\langle x, y\rangle, \langle y, x\rangle) \mid x, y \in V_0\}$.

The edges set E in OTIS-networks consists of two subsets, one is from G_0, called G_0-type edges, and the other subset contains the transpose edges. The OTIS approach suggests implementing G_0-type edges by electronic links since they involve intra-chip short links and implementing transpose edges by free space optics.

Definition 5.1 covers a wide class of OTIS-networks. In fact, for any known factor network G_0, a new OTIS-network can be obtained by the above definition. The OTIS-mesh [3], OTIS-hypercube [4] and OTIS-expander [10] are only few instances on such networks that have been investigated in the literature.

The address of a node $u = \langle x, y \rangle$ from V is composed of two components: the first, denoted by $\gamma(u)=x$, designates the group address and the second, denoted by $\rho(u)=y$, designates the processor address within that group.

The network OTIS-G_0 can be decomposed into $|V_0|$ disjoint copies of G_0. Fixing the group address and varying the processor address can achieve this decomposition. Another way of decomposing the OTIS-G_0 is by fixing the processors address and varying the group address. These two decomposition methods are given below.

Definition 5.2: Let Ψ_i, for all $i \in V_0$, be the subgraph induced by the set of nodes from V having the form $\langle i, x \rangle$ for all $x \in V_0$.

Definition 5.3: Let Φ_j, for all $j \in V_0$, be the subgraph induced by the set of nodes from V having the form $\langle x, j \rangle$ for all $x \in V_0$.

Figure 5.1 shows the decomposition of the OTIS-G_0 based on Ψ_i and Φ_j subgraphs. Each node in this decomposition is represented by (○), and each row and column of this decomposition has one node in common. In Ψ_i subgraph each processor is indexed by a pair $\langle x, y \rangle$, $0 \leq x, y < N$ where x is the group index and y is the processor index. In this decomposition, the group address is fixed while the processor address is variable. On the other hand, for Φ_j subgraph the group address is variable and the processor address is fixed. Each Ψ_i and Φ_j subgraphs represents a row or a column of the grid structure, respectively.

Any two subsets S_1 and S_2 share perfect matching if there is a bijective function between them. Given a graph G, for simplicity we denote by V_G the set of vertices, E_G the set of edges, $d_G(u, v)$ the length of a shortest path connecting u and v, and δ_G the diameter of G.

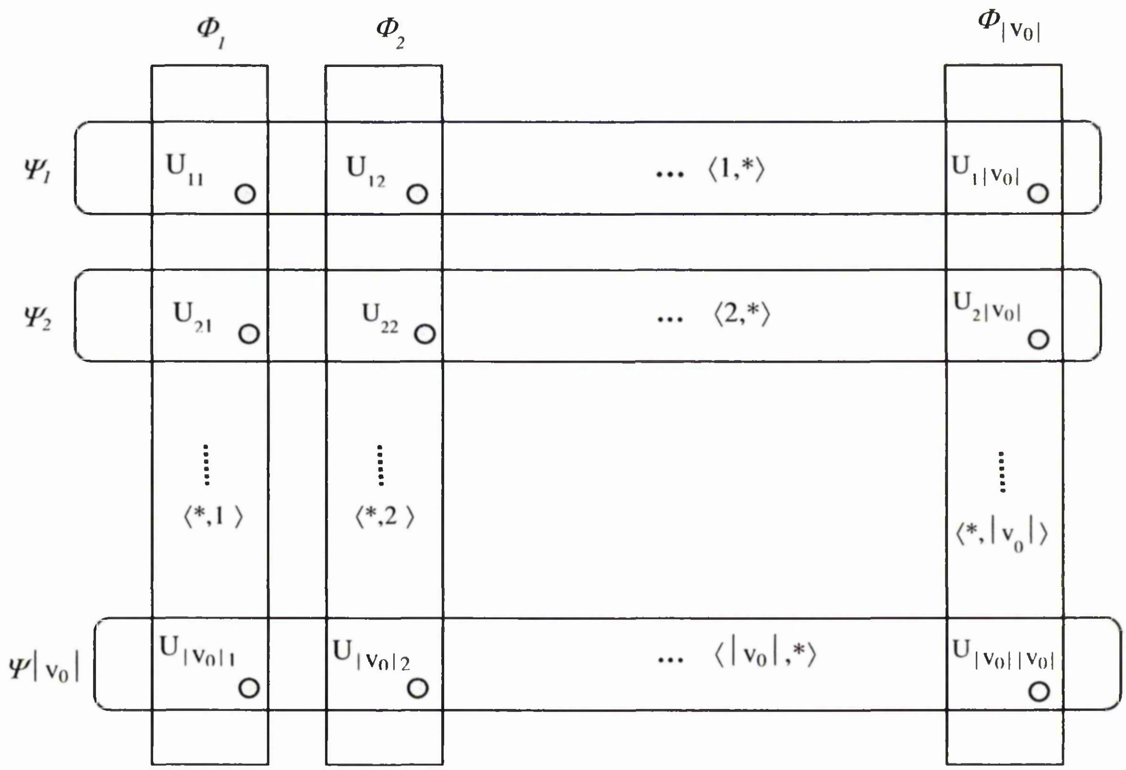


Figure 5.1: The grid structural outlook for OTIS-Networks.

Definition 5.4: Let $G_\Psi = (V_{G_\Psi}, E_{G_\Psi})$ be the graph obtained from OTIS-G_0 by clustering each of Ψ_i into a single vertex labelled by i and having a link between i and j if Ψ_i and Ψ_j share a prefect matching, i.e. $V_{G_\Psi} = V_0$ and $E_{G_\Psi} = \{(i, j) \mid$ for all i and j such that Ψ_i perfectly matches $\Psi_j\}$.

5.3 Hierarchical Decomposition for OTIS-Networks

In this section, the hierarchical structure of OTIS-networks is discussed. The properties of a new decomposition method for these networks are presented and proved. These properties are then used in the following sections to develop grids and pipelines as structural outlooks for developing various algorithms on OTIS-networks.

Theorem 5.1: The two Ψ and Φ decomposition methods of the OTIS-G_0 have the following properties:

1. Ψ_i is isomorphic to G_0.
2. $V_{\Psi_i} \cap V_{\Phi_j} = \{\langle i, j\rangle\}$.
3. Ψ_i and Φ_i share perfect matching for all i values.
4. Ψ_i and Ψ_j share perfect matching for all i and j values and hence G_Ψ is a complete graph.

Proof: Property 1 is a direct consequence of the definition of the OTIS-G_0. The function ρ maps nodes from V_{Ψ_i} to V_0. In fact, the set $\{\rho(u) \mid u \in \Psi_i\}$ is equal to V_0 for any i value. Since any two neighbouring nodes u and v in Ψ_i should have $\gamma(u)=\gamma(v)$ and $(\rho(u), \rho(v))$ is an edge in E_0; hence Ψ_i is isomorphic to G_0.

Property 2 states that for any two labels i and j from V_0, the two subgraphs Ψ_i and Φ_j have exactly one node in common. Since, $V_{\Psi_i} = \{\langle i, x\rangle \mid x \in V_0\}$ and $V_{\Phi_j} = \{\langle x, j\rangle \mid x \in V_0\}$, the intersection $V_{\Psi_i} \cap V_{\Phi_j}$ contains only the node $\langle i, j\rangle$.

Let $f_i : V_{\Psi_i} \rightarrow V_{\Phi_i}$ be a function that maps nodes form Ψ_i into Φ_i for all i values defined as follows: $f_i(\langle x, y\rangle) = \langle y, x\rangle$. First we have $|V_{\Psi_i}| = |V_{\Phi_j}|$ for all i and j. For any two distinct nodes u and v in V_{Ψ_I}, we have $f_i(\langle \gamma(u), \rho(u)\rangle) = \langle \rho(u), \gamma(u)\rangle \neq f_i(\langle \gamma(v), \rho(v)\rangle) = \langle \rho(v), \gamma(v)\rangle$; because $\rho(u) \neq \rho(v)$. Hence, the function f_i is one-to-one and onto. Thus property 3 follows.

Let t_{ij}: $V_{\Psi_i} \rightarrow V_{\Psi_j}$ be a function that maps nodes form Ψ_i into Ψ_j, for any i and j, as follows: $t_{ij}(\langle i, x\rangle)=\langle j, x\rangle$. For any two distinct nodes u and v from V_{Ψ_i} we have $t_{ij}(\langle i, \rho(u)\rangle) = \langle j, \rho(u)\rangle \neq t_{ij}(\langle i, \rho(v)\rangle) = \langle j, \rho(v)\rangle$. Since $|V_{\Psi_i}| = |V_{\Psi_j}|$ it follows that Ψ_i and Ψ_j share perfect matching for all i and j values and hence G_Ψ is a complete graph.

***Lemma 5.1*:** G_Ψ can be embedded into OTIS-G_0 with dilation $\delta_{G_0} + 2$.

***Proof*:** Since G_Ψ is complete graph, any two distinct nodes i and j in V_{G_Ψ} are neighbours. The "virtual" path between $\langle i, x\rangle$ and $\langle j, x\rangle$ in OTIS-G_0 that corresponds to the edge (i, j) in E_{G_Ψ} is constructed as follows: $\langle i, x\rangle \rightarrow \langle x, i\rangle \parallel \pi_{G_0}(i, j) \parallel \langle x, j\rangle \rightarrow \langle j, x\rangle$. An arrow represents an edge connecting the two nodes and the operation "||" means appending two paths (i.e. connecting the last node in the left path to first node in the right path). Notice that the choice of x from V_0 does not affect the construction of this path nor its length. The path segment $\pi_{G_0}(i, j)$ is an isomorphic copy to the optimal length path from i to j in G_0. It can be easily verified that the above constructed path is of optimal length equal to $d_{G_0}(i, j) + 2$. Hence, the longest such path cannot exceed δ_{G_0} + 2.

5.4 Structural Outlooks for OTIS-Networks

The grids and pipelines are popular computation models that have been employed in various parallel applications. They have been employed in early generations of parallel computers and they also continue to exist as convenient structural views for developing parallel algorithms [64-69]. In this section, we develop a two-dimensional grid structure and a pipeline structure for OTIS-networks.

***Theorem 5.2*:** OTIS-G_0 embeds a $|V_0|\times|V_0|$ grid with dilation $\delta_{G_0} + 2$, where rows and columns are G_0-configured.

***Proof*:** We arrange the nodes in OTIS-G_0 in a two-dimensional grid as follows: Let h: $V_0 \rightarrow \{1, 2, \ldots, |V_0|\}$ be a one-to-one and onto function that ranks the nodes of G_0. The node $\langle x, y\rangle$ in OTIS-G_0 is located at row $h(x)$ and column $h(y)$ in the $|V_0|\times|V_0|$ grid. With this mapping, the set of nodes in Ψ_x for any $x \in V_0$ are placed in the same row and the set of nodes in Φ_y for any $y \in V_0$ are placed in the same column. Since Ψ_x is isomorphic to

G_0 by property 1 in Theorem 5.1, then all rows in the grid are G_0-connected. The j^{th} column in the grid consists of $\langle x, h^{-1}(j)\rangle \mid x \in V_0\}$. Hence, all nodes in this set have the same processor address. By Lemma 5.1 any two nodes in this set are at a farthest distance of $\delta_{G_0} + 2$.

The above result can be used in two ways to facilitate algorithm development on OTIS-G_0. First, all algorithms that are known on G_0 can be ported to OTIS-G_0 at no additional cost. Second, the above $|V_0| \times |V_0|$ grid can be used to adapt mesh-based algorithms in a more efficient way. Divide-and-conquer and linear algebra [69] are few examples of such algorithms. The primary requirement in these algorithms is to be able to have simultaneous row or simultaneous column communication. The rows in the above grid structure are distinct copies of G_0 and hence processors across each row can communicate without interference from processors in other rows. Also columns are distinct copies of Φ_y, $y \in V_0$. Processors in each column can simultaneously communicate as follows: First the data in Φ_y is simultaneously transferred to Ψ_y through separate transpose channels connecting matching pairs from Φ_y and Ψ_y (by virtue of property 3 in Theorem 5.1). Then parallel communication is achieved in each row using the communication algorithms of G_0, and finally the data in Ψ_y is transferred back to Φ_y.

The following result along with its constructive proof gives a procedure to build generalised pipelines structural outlook in OTIS-G_0 networks. In pipelined processing, interstage communication should be order preserving in the sense that nodes with similar rank in successive stages should maintain a designated channel. In the ideal situation, these channels are node-disjoint so they don't interfere with each other and involve minimum intermediate nodes for fast stage-to-stage data transfer.

Theorem 5.3: If G_0 is Hamiltonian, OTIS-G_0 embeds a pipeline consisting of $|V_0|$ stages of size $|V_0|$ nodes each. Stages are G_0-configured and interstage distance is 3.

Proof: OTIS-G_0 network can be decomposed into $|V_0|$ disjoint copies of Ψ_x subnetworks. The Ψ_x subnetworks form the different pipeline stages. By Theorem 5.1 each of the Ψ_x's is isomorphic to G_0. We arrange the pipeline stages (the Ψ_x's) according the rank of x in a Hamiltonian cycle of G_0. Let h: $V_0 \rightarrow \{1, 2, \ldots, |V_0|\}$ be a function that defines the node's rank in the Hamiltonian cycle of G_0. So, the j^{th} stage consists of the set of nodes $\{\langle h^{-1}(j), y\rangle \mid y \in V_0\}$. The node $\langle h^{-1}(j), y\rangle$ in the j^{th} stage is coupled with the node $\langle h^{-1}(j+1), y\rangle$ in the $(j+1)^{st}$ stage of the pipeline. These two nodes are connected by the path $\langle h^{-1}(j), y\rangle \rightarrow \langle y, h^{-1}(j)\rangle \rightarrow \langle y, h^{-1}(j+1)\rangle \rightarrow \langle h^{-1}(j+1), y\rangle$. Notice that $h^{-1}(j)$ and $h^{-1}(j+1)$ are neighbours in G_0.

***Corollary* 5.1:** If G_0 is Hamiltonian, OTIS-G_0 can embed a two-dimensional wraparound mesh with dilation 3.

The above pipeline structural outlook exemplifies circular pipelines. Changing the number of stages or stage configuration (linear, circular, tree, etc.) in the above pipeline structure is straightforward. This can be done by characterising a path, cycle, or tree of size equal to the required number of stages in the new pipeline. Stages in the new pipeline are then ordered according to the ranks of the nodes in the characterised path, cycle, or tree. The interstage distance is 3 in all these cases. In fact, the result in Theorem 5.3 can be extended so that we have control over the stage structure as well. The stage can be G_0 or any network embedded in G_0.

5.5 Performance Evaluation

In this section, we evaluate the performance of the proposed structural outlooks for developing algorithms on grids and pipelines for two known OTIS-networks, the OTIS-hypercube [41] and OTIS-mesh [38]. The comparison is based on two standard measures of effectiveness: the network implementation cost and the network performance. In predicting the implementation cost we use the number of physical channels needed to achieve interconnects as a primary cost estimate. More involved estimates such as the required silicon area, system volume, channel length, power consumption rate, as well as optoelectronic receivers and transmitters can be used to give wider and precise image on the system's cost. However, we focus on the number of physical channels to stay within the scope of the thesis. This measure captures both of the wiring cost (electrical and/or optical) and the total I/O ports required for realising a network. More discussion on the OTIS cost and performance modelling can be found in [22, 58].

The network performance can be measured in different ways; the network bisection width, personalised communication cost and multicasts cost. We concentrate on a common form of communication as a measure to evaluate the performance of OTIS-networks. This form stems for the communication patterns in the proposed methods for developing algorithms on OTIS-networks, as explained below.

In the grid structural outlooks, the communication cost is measured by the amount of time needed to communicate a message across a row plus the time needed to communicate a message across a column. Usually, grid based applications emphasise simultaneous broadcast of uniform messages across rows and across columns [64-68, 69]. Hence, the communication time in a grid is estimated by the time required for a row broadcast followed by a column broadcast. A model used to estimate the cost of a multiple-port one-to-all broadcasting in a network of degree Δ and diameter δ is given by $\tau(m,\Delta,\delta) = \left(\sqrt{m \cdot a/(\beta \cdot \Delta)} + \sqrt{\delta - 1}\right)^2 \beta$, where β is the message latency, a is the unit transmission cost, and m is the message length [13]. This model determines the lower bound on the broadcasting cost in any vertex transitive network. The value of the parameters m, a, and β are are set to 1024 byte, 1 μs and 1000 μs, respectively [45].

In estimating the communication cost in the pipeline structural outlook, we consider a natural communication pattern in a pipeline processing system. This communication pattern involves a total message exchange at the stage level followed by a shifting phase to transfer data to peer nodes in the subsequent stage. The above model for multiple-port one-to-all broadcasting can be used to estimate the interstage communication time, however shifting the data between separate pairs of nodes in successive stages can be estimated by a simpler model. This model is given by $\beta+m\cdot\alpha$ [64, 68]. In this model we need β time units to set up the channel plus $m\cdot\alpha$ time units to shift the data.

Recent studies [20, 34] have revealed that free-space optical interconnects offer a speed advantage over electrical interconnect. Compared with the fastest off-chip electronically interconnects and for lengths up to a few centimetres electronically interconnects are faster than optical interconnects, optical interconnects provide as much as a twice better speed performance [58]. To simplify the analysis, there is no distinction between electronic and optical channels in the subsequent expressions. With this assumption we will still have a suitably accurate estimate on the system performance and a rough idea about the system cost.

Table 5.1 summarises the parameters needed to estimate the communication time and the implementation cost for the four networks: the hypercube, the mesh, the OTIS-hypercube, and the OTIS-mesh as functions on the network size (N). The third and fourth columns show the degree and the diameter, respectively for the sub-network given in the second column, which is connecting the set of nodes in a row (or a stage) of the grid structure (or the pipeline structure). Similarly, the sixth and seventh columns respectively show the degree and the diameter for the sub-network given in the fifth column, which is connecting the set of nodes in a column of the grid structure. The eighth column shows the inter-stage distance in the pipeline structure. The last column shows the number of links.

Applying the above models for estimating the communication time using the information in table 5.1, we obtain the expressions shown in table 5.2. Figure 5.2 shows the estimated communication cost based on the proposed grid structure. The figure shows that the communication cost in the OTIS-mesh is lower than its conventional counterpart, and the communication cost in the OTIS-hypercube is comparable to the hypercube performance. A similar performance for the pipeline structure can be observed in Figure 5.3. For the grid structure the communication cost is obtained by using Seidel formula [45]. Furthermore for the pipeline structure we used Seidel formula to calculate the communication cost within the stage plus the time needed to shift the data to the next stage.

A more balanced comparison should give an idea about the cost associated with a performance gain. Figures 5.4 and 5.5 depict these facts. These figures show that the proposed methods for algorithm development on OTIS-networks achieve better-cost performance ratio, which means the gain in performance comes at a lower cost. By noticing that an additional channel squares the size of OTIS-networks, its superiority in terms of implementation cost over the electronic alternatives becomes more evident. In this measure we multiplied the communication cost based on grid and pipeline structures by the number of the links of OTIS-networks and their counterparts [45].

Today's VLSI technology has serious limitations in terms of chip pin-out. For instance, the decreasing interest in the hypercube and similar logarithmic degree networks is attributed to the hardware limitations to support higher dimensionality. As the size of the system increases, it becomes exceedingly difficult to pack the required processing units within a single chip. The OTIS tackles these serious design concerns by utilising the combined strengths of 3-D chip packaging and optoelectronic interconnect technologies to bring a low-power ultra-compact hardware solution to systems requiring fast processing. The goal is to approach single chip performance using multi-chip devices. The figures demonstrate that OTIS-networks provides new horizons to implement large-scale systems that are not possible with today's conventional electronic technology, and yet perform efficiently in terms of supporting real applications.

	Row/stage parameters			Column parameters			Inter-stage distance	Number of links
	Ψ_l	*Degree*	*Diameter*	Φ_l	*Degree*	*Diameter*		
Hypercube	$Q_{(\log N)/2}$	$\frac{1}{2}\log N$	$\frac{1}{2}\log N$	$Q_{(\log N)/2}$	$\frac{1}{2}\log N$	$\frac{1}{2}\log N$	1	$\frac{1}{2}N\log N$
Mesh(m, n)	m	2	$m-1$	N	2	n-1	1	$(2m.n - m - n)$
OTIS-hypercube	$Q_{(\log N)/2}$	$\frac{1}{2}\log N$	$\frac{1}{2}\log N$	G_Ψ	$\frac{1}{2}\log N$	$\frac{1}{2}\log N+2$	3	$\frac{1}{2}(N-\sqrt{N}+N\log N)$
OTIS-mesh	$M(\sqrt[4]{n.m}, \sqrt[4]{n.m})$	4	$\sqrt[4]{n.m}$	G_Ψ	4	$\sqrt[4]{n.m}+2$	3	$(n^2.m^2-n.m)/2+2(n^2.m^2)$

Table 5.1: Cost and performance parameters in the hypercube, mesh and their OTIS counterparts.

	Grid	Pipeline
Hypercube	$2\times\tau(m, \frac{1}{2}\log N, \frac{1}{2}\log N)$	$\tau(m, \frac{1}{2}\log N, \frac{1}{2}\log N) + (\beta+m\cdot\alpha)$
Mesh	$2\times\tau(m, 2, \sqrt{N})$	$\tau(m, 2, \sqrt{N}) + (\beta+m\cdot\alpha)$
OTIS-Hypercube	$2\times\{\tau(m, \frac{1}{2}\log N, \frac{1}{2}\log N)+(\beta+m\cdot\alpha)\}$	$\tau(m, \frac{1}{2}\log N, \frac{1}{2}\log N)+3\times(\beta+m\cdot\alpha)$
OTIS-Mesh	$2\times\{\tau(m, 4, \sqrt[4]{N}) + (\beta+m\cdot\alpha)\}$	$\tau(m, 4, \sqrt[4]{N}) + 3\times(\beta+m\cdot\alpha)$

Table 5.2: Communication time in the grid and the pipeline structures.

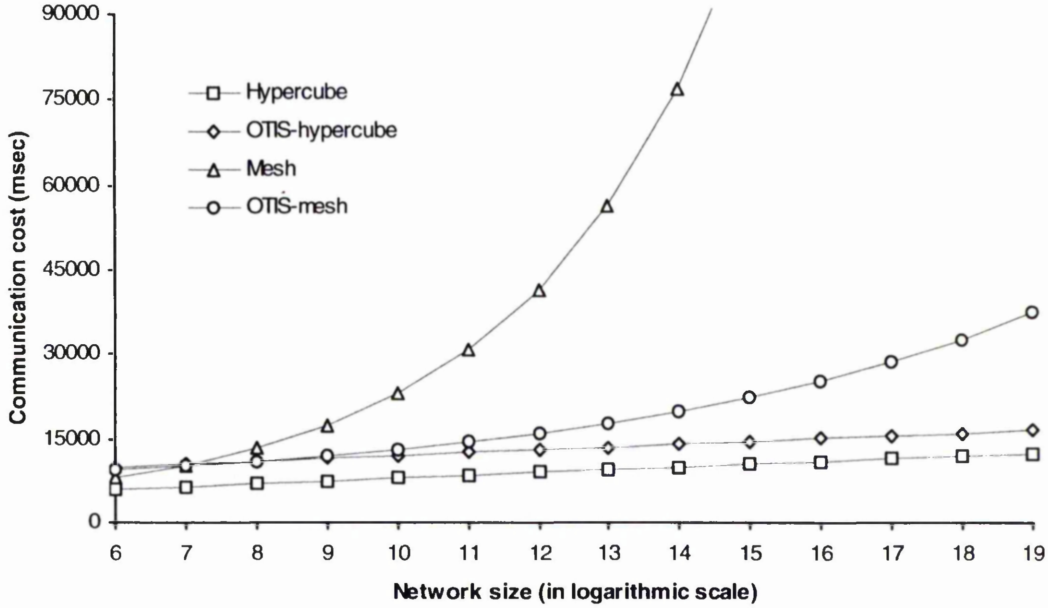


Figure 5.2: The communication cost in the grid structure.

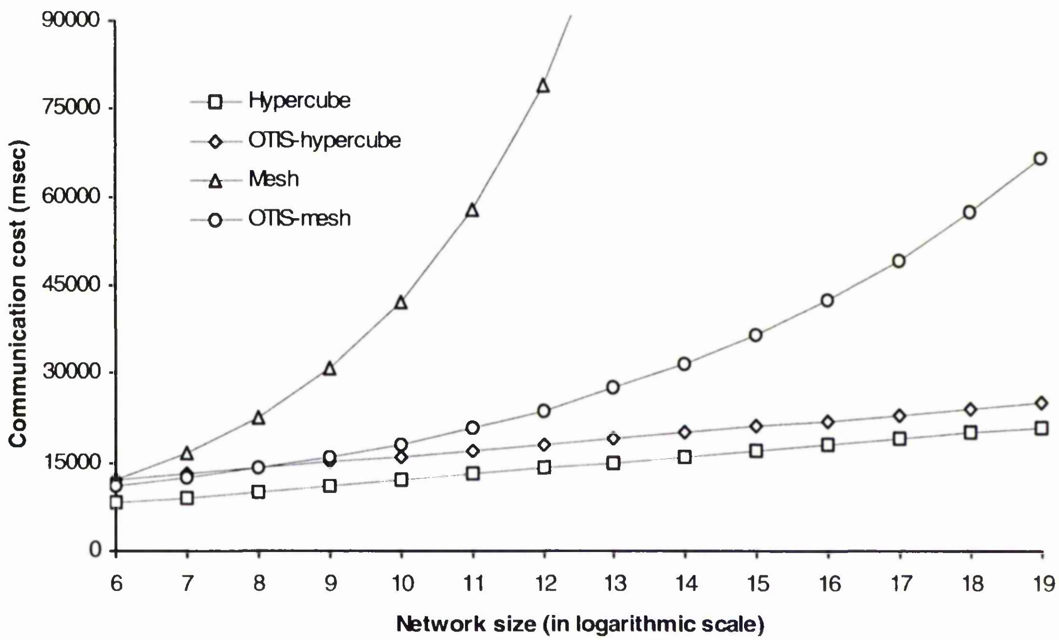


Figure 5.3: The communication cost in the pipeline structure.

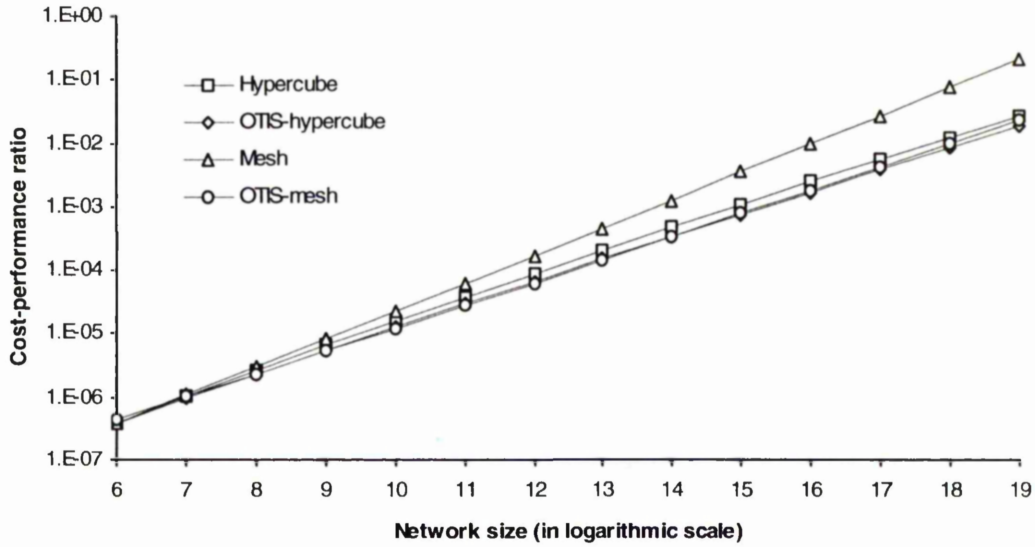


Figure 5.4: The cost-performance in the grid structure.

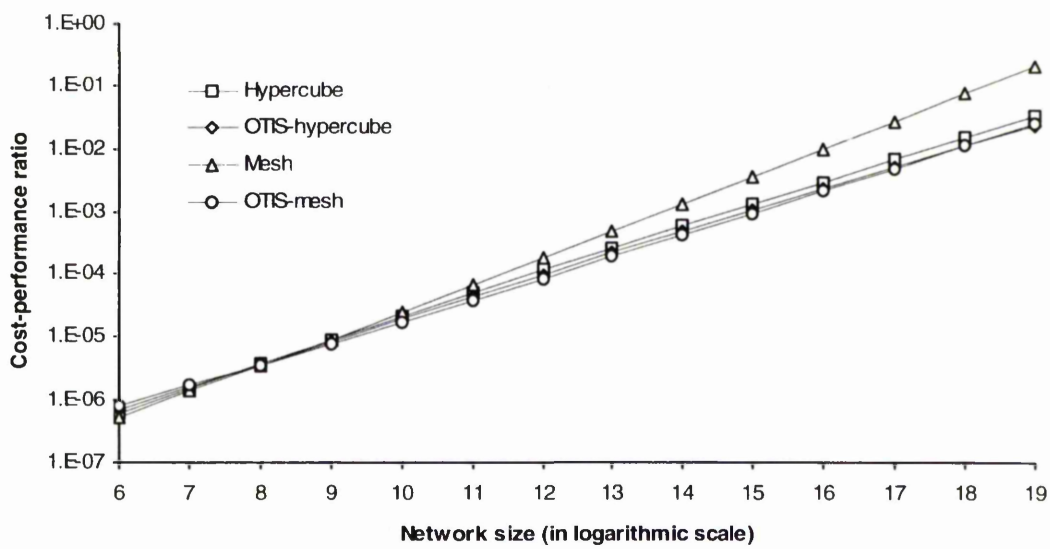


Figure 5.5: The cost-performance in the pipeline structure.

5.6 Conclusions

This chapter has proposed and evaluated two structural outlooks that allow the development of efficient algorithms for a wide class of problems on OTIS-networks. These outlooks are based on grids and pipelines as popular structures that support a large number of applications including linear algebra, divide-and-conquer algorithms, sorting, and Fourier transforms. Timing models for measuring the performance of the proposed outlooks have also been provided. These models have been used to evaluate and compare the performance of various algorithms on the OTIS-networks against their counterparts on conventional electronic interconnection systems. The results presented have revealed that the OTIS-mesh and OTIS-hypercube provide better cost-performance characteristics than the traditional mesh and hypercube. These obtained results indicate that OTIS-networks have the potential to be efficient candidates for future large-scale computers. The next chapter will show how the structural outlook based on the grid view is used to design efficient parallel algorithms for solving linear algebra problems, e.g. systems of linear equations, on OTIS-networks.

Chapter 6

Solving Systems of Linear Equations on OTIS-Networks

6.1 Introduction

In OTIS-networks, shorter (intra-chip) communication is realised by electronic interconnects while longer (inter-chip) communication is realised by free space interconnects. A number of recent studies have been conducted in this direction [24, 38, 41, 61-63]. Zane *et al* [36] have shown that the OTIS-mesh efficiently embeds 4-dimensional meshes and hypercubes. Sahni and Wang [38, 41] have presented and evaluated various algorithms on OTIS-networks, such as basic data rearrangements, routing, selection and sorting. They have also developed algorithms for various matrix multiplication operations [62] and image processing [24].

A side from the above research studies, there has been hardly any work that has proposed efficient parallel algorithms for other well-known problems on OTIS-networks. In an effort to address this imbalance we apply the grid structural outlook proposed in Chapter 5 to tackle a important problem from linear algebra that arises in many areas of science and engineering [70, 74]. In this chapter, we develop parallel algorithms for solving systems of linear equations, $A\vec{x}=\vec{b}$, on OTIS-networks. This study will demonstrate that OTIS-networks are potential candidates for high-performance parallel computers.

An OTIS-network contains N^2 processors partitioned into N groups containing N processors each. Processors in the same group are connected to each other through a certain topology using electronic interconnects, while transposing group and processor indexes attains the links in processors in different groups using optical interconnects. The address of any node $u = \langle x, y\rangle$ from V (set of vertices) is consists of two parts: the first, denoted by $\gamma(u)=x$, designates the group address and the second, denoted by $\rho(u)=y$, designates the processor address within that group; see Definition 5.1 in Chapter 5 for a formal definition of OTIS-networks.

It has been shown in the previous chapter that the network, OTIS-G_0, for a factor network, G_0, can be decomposed into $|V_0|$ disjoint copies of G_0 [70]. Fixing the group address and varying the processor address can achieve this decomposition [70, 74]. Another way of decomposing the OTIS-G_0 is achieved by fixing the processor address and varying the group address.

Two ranking for the nodes of rows and columns are $h(x)$ and $h(y)$ are defined as follows:

Let h: $V_0 \rightarrow \{1, 2, \ldots, |V_0|\}$ be a one-to-one and onto function that ranks the nodes of G_0. The node $\langle x, y\rangle$ in OTIS-G_0 is located at row $h(x)$ and column $h(y)$ in the $|V_0|\times|V_0|$ grid structural outlook.

The properties of the decomposition method for OTIS-networks have been presented in Chapter 5. These properties will be used in the subsequent section to develop various parallel algorithms on OTIS-networks for solving system of linear equations based on grid structural outlook. The two Ψ and Φ decomposition methods of the OTIS-G_0 have the following properties

1. Ψ_i is isomorphic to G_0.

2. $V_{\Psi_i} \cap V_{\Phi_j} = \{\langle i, j \rangle\}$.

3. Ψ_i and Φ_i share perfect matching for all i values.

4. Ψ_i and Ψ_j share perfect matching for all i and j values and hence G_Ψ is a complete graph.

The formal Proofs of these properties have been presented in [70]. It has also been shown in [70] that G_Ψ can be embedded into OTIS-G_0 with dilation δ_{G_0} + 2.

The grids are popular computation models that have been employed in various parallel applications in linear algebra, divide-and-conquer algorithms, sorting, and Fourier transforms, among others [32, 44]. They have been employed in early generations of parallel computers and they also continue to exist as convenient structural views for developing parallel algorithms [64-69].

OTIS-G_0 embeds a $|V_0|\times|V_0|$ grid with dilation δ_{G_0}+2 [70], where rows and columns are G_0-configured. This result can be used in two ways to facilitate algorithm development on OTIS- G_0. First, all algorithms that are known on G_0 can be ported to OTIS- G_0 at no additional cost. Second, the $|V_0|\times|V_0|$ grid can be used to adapt mesh-based algorithms in a more efficient way. Divide-and-conquer and linear algebra [69] are a few examples of such algorithms. The primary requirement in these algorithms is to be able to have simultaneous row or simultaneous column communication. The rows in the grid structural outlook are distinct copies of V_0 and hence processors across each row can communicate without interference from processors in other rows [70]. Also columns are distinct copies of Φ_y, $y \in V_0$. Processors in each column can simultaneously communicate as follows: First the data in Φ_y is simultaneously transferred to Ψ_y through separate transpose channels connecting matching pairs from Φ_y and Ψ_y . Then parallel communication is achieved in each row using the communication algorithms of G_0, and finally the data in Ψ_y is transferred back to Φ_y.

6.2 $A\vec{x} = \vec{b}$ with a Method of Solution

In this section, we apply the proposed grid structural outlook for solving systems of linear equations, $A\vec{x}=\vec{b}$, a problem that arises in many areas of science and engineering [17, 64, 65, 68, 69]. A direct method of solution transforms the system $A\vec{x}=\vec{b}$ into $U\vec{x}=\vec{c}$, where U is an upper triangular matrix. The solution vector $\vec{x}$ is then obtained by back substitution. The standard procedure to carry out LU decomposition of A is Gaussian elimination (GE). The GE procedure generates a sequence of $n \times n$ matrices $A^{(1)}$, $A^{(2)}$, ..., $A^{(n)}$ where $A^{(1)}$ is the initial matrix, A, and $A^{(n)}$ is the desired triangular matrix, U. The matrix $A^{(k)}$ (k=2, ..., n) represents the equivalent linear system for which the variable x_{k-1} has just been eliminated.

In this implementation of matrix triangulation, we use the standard 2-D matrix distribution method that is characterized by "combining" two functions; the first partitions the augmented matrix $A\vec{x} \mid \vec{b}$ into sets of elements and the second assigns these parts to the processors [11, 16]. In the sequel, we present a practical version of this distribution method.

In the proposed $|V_0| \times |V_0|$ grid structure, rows are the disjoint Ψ_x's and columns are the disjoint Φ_x's. Following the notation of Theorem 5.2, presented in Chapter 5, the processor at the r^{th} row and the c^{th} column of the grid (denoted by p_{rc}) is laid down at the intersection of Ψ_x and Φ_y, where $r = h(x)$ and $c = h(y)$. Recall that h is a function that ranks nodes of V_0.

For any positive integer ι, we denote by $\langle\iota\rangle$ the set of integer numbers from 1 to ι. Let $\vartheta: \langle|V_0|\rangle \rightarrow 2^{\langle n\rangle}$ be a surjective function that maps processor row indexes into subsets of matrix row indexes. Similarly, let $\chi: \langle|V_0|\rangle \rightarrow 2^{\langle n+1\rangle}$ be a surjective function that maps processor column indexes into subsets of matrix column indexes. Our matrix distribution method is characterized by the function $\xi = \vartheta \times \chi$. The function ξ maps ordered pairs from $\langle|V_0|\rangle \times \langle|V_0|\rangle$ into sets of ordered pairs from $2^{\langle n\rangle \times \langle n+1\rangle}$. The set $\xi(r, c)$ induces the set of elements from $A \mid \vec{b}$ to be assigned to p_{rc}.

The function ξ covers the class of matrix distribution methods that map elements of the same row (respectively column) to a processor in Ψ_x for some x (resp. Φ_y for some y). These are called block-distribution methods. In what follows, we consider only those instances of ξ that are based on block-distribution and map the matrix evenly to the set of processors. Furthermore, we assume $\vartheta(r_1)\cap\vartheta(r_2)=\phi$ for $r_1 \neq r_2$ and $\chi(c_1)\cap\chi(c_2)=\phi$ for $c_1 \neq c_2$, where r_1 and r_2 are any two different rows and c_1 and c_2 are any two different columns. Figure 6.1 shows example of block matrix distribution and the relationship between rows and columns in a 16-node OTIS-mesh.

The broadcast-based parallel GE algorithm for OTIS-G_0, abbreviated *b*_OTIS, reduces a matrix A of order n to triangular form requires $(n-1)$ steps. At any step, k, the following tasks have to be performed by p_{rc} in the order given below.

Task $\pi_p \equiv$ { *pivoting* } transform the system so that

$$|a_{kk}| = max\{|a_{ik}| \text{ for } k \le i \le n\} \quad \text{– partial pivoting}$$

$$|a_{kk}| = max\{|a_{ij}| \text{ for } k \le i,j \le n\} \quad \text{– complete pivoting}$$

Task $\pi_m^r \equiv$ { *compute multipliers* } $a_{ik} = a_{ik} / a_{kk}$ for all $i \in \vartheta(r)$ such that $k+1<i \le n$

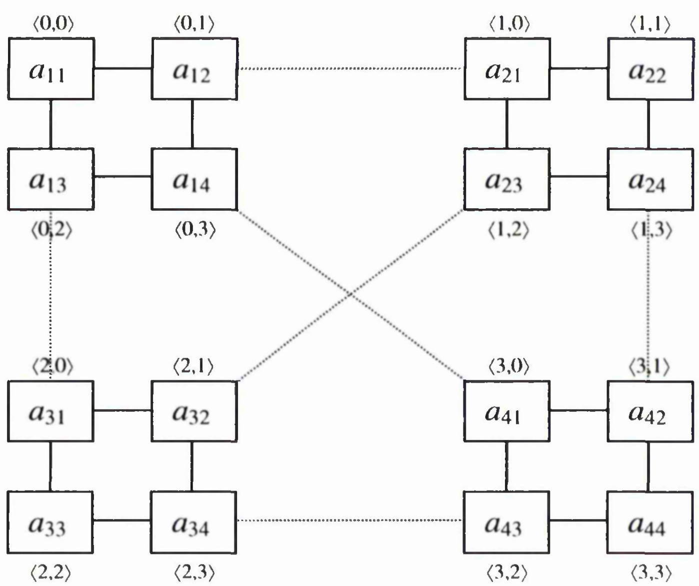
Task $\pi_e^{r,c} \equiv$ { *eliminate* } $a_{ij} = a_{ij} - a_{ik}\, a_{kj}$ for all $i \in \vartheta(r)$ and $j \in \chi(c)$ such that $k+1<i$, $j \le n$,

Let π_y^M denote the task of broadcasting a pivot subrow of M elements in Φ_y, and let π_x^M denote the task of broadcasting a multiplier subcolumn of M elements in Ψ_x. Notice that broadcasting in Φ_y use the usual communication algorithms of G_0 except that two data shift operations have to "bracket" communication algorithms to transfer data from Φ_y to Ψ_y and back by using the disjoint transpose edges.

The algorithm *b*_OTIS executed by each node p_{rc} in the $|V_0| \times |V_0|$ grid is shown in Figure 6.2. The steps of *b*_OTIS can be summarised as follows: If p_{rc} holds a pivot, it performs π_p followed by $\pi_y^{(n+1)/|V_0|}$. Otherwise it waits until it receives a pivot. Next, if p_{rc} holds a multiplier, it performs π_m^r and then $\pi_x^{n/|V_0|}$, otherwise it waits until it receives a multiplier. Finally, p_{rc} performs $\pi_e^{r,c}$ and enters the next iteration.

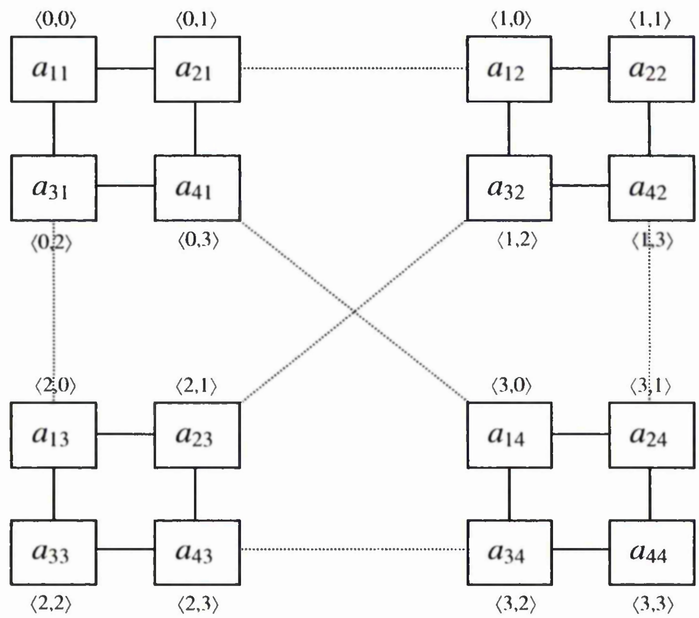
For partial pivoting, the k^{th} pivot row is determined by processors in $\Phi_{h^{-1}(c)}$, where $k \in \chi(c)$. These processors perform an "exchange-max" operation. At the end of this operation each processor will have a copy of the index of the pivot row, say l. If both l and k are in $\vartheta(r)$ then locally swap the two subrows k and l; otherwise interchange the two subrows k and l between all pairs of corresponding processors in $\Psi_{h^{-1}(r)}$ and $\Psi_{h^{-1}(v)}$ for $l \in \vartheta(v)$. The algorithm for task π_p with partial pivoting is given in Figure 6.3.

To carry our complete pivoting all processors in the grid perform exchange-max operation to find l and m such that $|a_{l,m}| = max\{|a_{ij}| \text{ for } k \leq i,j \leq n\}$. Then, the k^{th} pivot row is located by swapping/interchanging the rows k and l and swapping/interchanging the columns k and m. A complete pivoting algorithm for b_OTIS is given in Figure 6.4.



(a) Each row is in separate Ψ_x network.

Figure 6.1: $A_{4\times4}$ matrix distribution over 16 node OTIS-mesh.



(b) Each column is in separate Ψ_x network.

Figure 6.1: $A_{4\times4}$ matrix distribution over 16 node OTIS-mesh.

Algorithm *b*_OTIS

{ executed by each node p_{rc} in the $|V_0|\times|V_0|$ grid }

for k=1 to $n-1$ do

execute π_p

if $k\in\vartheta(r)$ execute $\pi_y^{(n+1)/|V_0|}$ to broadcast a pivot subrow in $\Phi_{h^{-1}(c)}$

else participate in the broadcast initiated in $\Phi_{h^{-1}(c)}$

if $k\in\chi(c)$ execute π_m^r then $\pi_x^{n/|V_0|}$ to broadcast a multiplier subcolumn in $\Psi_{h^{-1}(r)}$

else participate in the broadcast initiated in $\Psi_{h^{-1}(r)}$

execute $\pi_e^{r,c}$

end for

end *b*_OTIS

Figure 6.2: The algorithm *b*_OTIS.

Algorithm Task π_p: Partial pivoting

if $k \in \chi(c)$

find l such that $|a_{l,k}| = max\{|a_{i,k}|$ for all $i \in \vartheta(r)$ such that $k \leq i \leq n$ }

perform exchange-max operation to determine $a_{l,k}$ on all processors in $\Phi_{h^{-1}(c)}$

broadcast l in $\Psi_{h^{-1}(r)}$

else participate in the broadcast initiated in $\Psi_{h^{-1}(r)}$

if $l \in \vartheta(r)$ and $k \in \vartheta(r)$ swap subrows k and l

else interchange subrows k and l between $\Psi_{h^{-1}(r)}$ and $\Psi_{h^{-1}(v)}$ for $l \in \vartheta(v)$

end Task π_p

Figure 6.3: Partial pivoting algorithm for *b*_OTIS.

Algorithm Task π_p: Complete pivoting

find l and m such that $|a_{l,m}| = max\{|a_{ij}|$ for all $i \in \vartheta(r)$ and $j \in \chi(c)$ such that $k \leq i,j \leq N\}$

perform exchange-max operation to exchange $a_{l,m}$ with all processors in the grid

if $l \in \vartheta(r)$ and $k \in \vartheta(r)$ swap subrows l and k

else interchange subrows l and k between $\Psi_{h^{-1}(r)}$ and $\Psi_{h^{-1}(v)}$ for $l \in \vartheta(v)$

if $m \in \chi(c)$ and $k \in \chi(c)$ swap subcolumns m and k

else interchange subcolumns m and k between $\Phi_{h^{-1}(c)}$ and $\Phi_{h^{-1}(v)}$ for $m \in \chi(v)$

end Task π_p

Figure 6.4: Complete pivoting algorithm for *b*_OTIS.

The algorithm *b*_OTIS transforms the linear system $A\vec{x} = \vec{b}$ into an upper triangular system $U\vec{x} = \vec{c}$. Next we develop a broadcast-based backward substitution algorithm, abbreviated *bs*_OTIS, to obtain the final solution. In *bs*_OTIS we assume that elements of the reduced upper triangular matrix $U \mid \vec{c}$ are still in their respective processors. A processor p_{rc} in the algorithm performs n steps. A step, k, consists of the following simple tasks: (1) solve and broadcast x_k and (2) update and broadcast $\vec{c}$. To simplify the algorithm design we introduce the following two new tasks:

Task π_s $\equiv$ { *solve* } $x_k = c_k / u_{kk}$

Task π_u^r $\equiv$ { *update* $\vec{c}$ } $c_i^{\cdot} = c_i - u_{ik}\, x_k$ for all $i \in \vartheta(r)$ such that $1 \le i \le k$

At the beginning of the backward substitution algorithm, a processor p_{rc} in $\Phi_{h\text{-}1(c)}$ that holds the set of elements $\{c_i \mid i \in \vartheta(r)\}$ from $\vec{c}$ (denote this set by Cr) should broadcast these elements in $\Psi_{h\text{-}1(r)}$. Notice that initially the known vector $\vec{c}$ is stored in $\Phi_{h\text{-}1(c)}$ where $n+1 \in \chi(c)$.

The backward substitution algorithm then proceeds by performing the n substitution steps. At the k^{th} step, the processor p_{rc} holding u_{kk} performs π_s and then broadcast x_k in $\Phi_{h\text{-}1(c)}$. All other peers of p_{rc} in $\Phi_{h\text{-}1(c)}$ update their portion of $\vec{c}$ vector and broadcast it in their respective $\Psi_{h\text{-}1(r)}$'s. Figure 6.5 outlines this algorithm.

```
Algorithm bs_OTIS      { executed by the node p_rc in the |V0|×|V0| grid }
        if n+1 ∈ χ(c) execute π_x^{n/|V0|} to broadcast Cr in Ψ_{h-1(r)}
        else participate in the broadcast initiated in Ψ_{h-1(r)}
        for k = n downto 1 do
        if k∈χ(c) and k∈ϑ(r) execute π_s and then execute π_y^1 to broadcast x_k in Φ_{h-1(c)}
                else if k∈χ(c)
                        participate in the broadcast initiated in Φ_{h-1(c)}
                          execute π_u^r
                          execute π_x^{n/|V0|} to broadcast Cr in Ψ_{h-1(r)}
                  else participate in the broadcast initiated in Ψ_{h-1(r)}
        end for
end bs_OTIS
```

Figure 6.5: The algorithm *bs*_OTIS.

6.3 Conclusions

The advantage of using OTIS-network as optoelectronic architectures lies in their ability to manoeuvre the fact that free space optical links are superior in terms of speed and power consumption when the connect distance is more than few millimetres. Except from the research work of Wang and Sahni [24, 38, 62], there has been hardly any study that has proposed parallel algorithms on OTIS-networks. In this chapter, we have contributed towards filling this gap by designing parallel algorithms to solve systems of linear equations, $A\vec{x}=\vec{b}$, on these networks. The proposed algorithms are based on the grid structural outlook developed in Chapter 5. By presenting such parallel algorithms, this study has highlighted the suitability of OTIS-networks for solving one of the most important problems in Linear Algebra, and which arises in various research and practical areas.

Chapter 7

Conclusions and Future Directions

Over the past two decades many interconnection networks have been proposed in the literature, including the star, hypercube, arrangement and OTIS-networks. Most of the existing research on these networks has focused on analysing their topological properties. As a result, there has been relatively little activity committed to developing efficient algorithmic structural outlooks for important parallel applications. In an attempt to fill this gap, this research work has introduced two general structural outlooks for developing efficient algorithmic frameworks on vertex product networks, namely the arrangement-star network and OTIS-networks. These structural outlooks are based on grids and pipelines as popular structures that support a vast body of real-world parallel applications such as matrix computation, linear algebra, divide-and-conquer type of algorithms, sorting, and Fourier transforms.

7.1 Outline of the Results

The performance of various algorithms on the arrangement-star network and OTIS-networks can be evaluated and compared with their counterparts using the proposed structural outlooks. This study has provided new means for further testing the viability of the arrangement-star and OTIS-networks as attractive alternatives for future high-performance parallel computers. The main contributions in the thesis are itemised as follows:

- First, a new topology has been proposed, referred to as the *arrangement-star* network, which is constructed from the product of the star and arrangement graphs. A comparative analysis has revealed that the proposed network has superior topological properties over its factors: the star and arrangement graphs. Besides possessing a smaller diameter, node degree, and number of links, the arrangement-star has a lower broadcasting cost and more flexibility in choosing the desired network size. A new measure, called the *degree of accuracy*, has been used to compare the three networks. The arrangement-star maintains a higher degree of accuracy in terms of fitness to the desired network size, which is a great improvement over the star and arrangement networks. Furthermore, the proposed arrangement-star has been shown to be vertex symmetric. Such a property enables this network to support important applications such as matrix computations and divide-and-conquer algorithms.

- Second, a comparative study of the arrangement-star against the star and hypercube has been conducted. The comparison used the proposed structural outlooks, based on the grid and pipeline structures, and considered different network sizes and fixed message length. The results have shown that the new network outperforms both the star and hypercube due to its lower communication cost. These results add another evidence on the viability on the proposed arrangement-star network.

- Third, vertex product networks have been proposed as a unified groundwork covering a wide class of interconnection networks including important sub-classes such as the *Optical Transpose Interconnection Systems* (OTIS). This unification allows the development of generic algorithms which makes it possible to save lots of efforts that would be needed to study each network in this class individually.

- Fourth, two general structural outlooks for algorithm development on OTIS-networks have been proposed and evaluated. These structures are based on grids and pipelines as common outlooks often used in the design of parallel algorithms.

- Fifth, timing models for measuring the performance of the proposed outlooks on OTIS-networks have been provided. The results presented in this thesis have shown that the performance of algorithms on OTIS-networks, based on the grid and pipeline outlooks, outperform their counterparts on conventional electronic interconnection systems in terms of the communication cost and cost/ performance criteria. For instance, the OTIS-mesh and OTIS-hypercube have been shown to provide better communication cost and cost-performance characteristics over the traditional mesh and hypercube.

- Sixth, the structural outlooks for algorithm development proposed for OTIS-networks have been used to new parallel algorithms for solving systems of linear equations, which is one of the important problems in Linear Algebra that arises in many fields of science and engineering.

7.2 Future Research Directions

There are several interesting issues that require further investigation. These are summarised below.

- A natural follow-up for this work would be to develop a complete theory that supports ready-made results for the "comprehensive" class of vertex product networks. With this theory the various topological properties of any vertex product networks can be derived in straightforward manner. In addition to enabling meaningful comparative studies of vertex product networks, this would significantly reduce the efforts that would be undertaken to study these networks individually.

- Another extension of the work presented in this thesis could be the generalisation of structural outlooks for the class of vertex product networks. Again using this generalisation several real algorithms can be ported into arbitrary vertex product networks and also the resulting algorithms can be compared with each other and with the original versions in the constitute networks from which the vertex product networks are built.

- Irregular networks have recently received a lot of attention from the research community due to the emergence of "cluster of workstations" as a cost-effective method for achieving parallel processing. A new direction of research along the broad lines of this thesis would be to investigate the design of efficient parallel algorithms on these networks.

References

[1] S. B. Akers, and B. Krishnamurthy, "A Group Theoretic Model for Symmetric Interconnection Networks," *Proc. Intl. Conf. Parallel Processing*, 1986, pp. 216-223.

[2] K. Day and A. Tripathi, "A Comparative Study of Topological Properties of Hypercubes and Star Graphs," *IEEE Trans. Parallel & Distributed Systems*, vol. 5, no. 1, 1994, pp. 31-38.

[3] S. B. Akers, D. Harel and B. Krishnamurthy, "The Star Graph: An Attractive Alternative to the n-Cube," *Proc. Intl. Conf. Parallel Processing*, 1987, pp. 393-400.

[4] K. Day and A. Al-Ayyoub, "The Cross Product of Interconnection Networks", *IEEE Trans. Parallel and Distributed Systems*, vol. 8, no. 2, Feb. 1997, pp. 109-118.

[5] Stone H. S., "Parallel Processing with the Perfect Shuffle," *IEEE Trans. Computers*, C-20, 1971, pp.153-161.

[6] A. Al-Ayyoub and K. Day, "A Comparative Study of Cartesian Product Networks", *Proc. of the Intl. Conf. on Parallel and Distributed Processing: Techniques and Applications*, vol. I, August 9-11, 1996, Sunnyvale, CA, USA, pp. 387-390

[7] A. Awwad, A. Al-Ayyoub, M. Ould-Khaoua, "On the Topological Properties of the Arrangement-star Network," *Submitted to the Journal of Systems Architecture.*

[8] Y. Y. Saad and M. H. Schultz, "Topological Properties of Hypercubes," *IEEE Trans. Computers,* vol. 37, no. 7, July 1988, pp. 867-872.

[9] C. L. Wu and T. Y. Feng, "The Reverse Exchange Interconnection Network." *IEEE Trans. Computers*, Vol. 29, 1980, pp. 801-811.

[10] A. Menn and A.K. Somani, "An Efficient Sorting Algorithm for the Star Graph Interconnection Network," *Proc. Intl. Conf. on Parallel Processing*, 1990, pp.1-8.

[11] S. Lakshmivarahan, J. S. Jwo, and S. K. Dhall, "Analysis of Symmetry in Interconnection Networks Based on Cayley Graphs of Permutation Groups: A Survey," Tech. Report School of Electrical Engineering and Computer Science, The University of Oklahoma, August 1991.

[12] A. L. Rosenberg, "Issues in the Study of Graph Embedding," Graph Theoretic concepts in Computer Science, Lecture Notes in Computer Science, Vol. 100, Springer-Verlag, New York, 1981, pp. 150-176.

[13] S. L. Johnson and C-T Ho, "Optimum Broadcasting and Personalised Communication in Hypercubes," *IEEE Trans. Computers*, Vol. 38, No 9, September 1989, pp. 1249-1268.

[14] F. Harary, "Graph Theory," Addison Wesley, 1969, pp. 47-48.

[15] S. Sahni and C. Wang, "BPC Permutations on the OTIS-mesh Optoelectronic Computer," *Proceedings of the IEEE Conference on Massively Parallel Programming using Optical Interconnect*, 1997, pp. 130-135.

[16] W.C. Athas and C.L. Seitz, "Multicomputers: Message-Passing Concurrent computers," *Computer* 21, 1988, pp. 9-24.

[17] D. Bertsekas and J. Tsitsiklis, "Parallel and Distributed Computation: Numerical Methods," Prentice-Hall, 1989.

[18] D. Saika and R. K. Sen, "Two Ranking Schemes for Efficient Computation on the Star Interconnection Network," *IEEE Trans. Parallel & Distributed Systems*, vol. 7, 1996, pp. 321-327.

[19] D. Saika and R. K. Sen, "Order Preserving Communication on a Star Network," *Parallel Computing* 21, 1995, pp. 1292-1300.

[20] W. Hendrick , O. Kibar, P. Marchand, C. Fan, D. Blerkom, F. McCormick, I. Cokgor, M. Hansen, and S. Esener, "Modeling and Optimisation of the Optical Transpose Interconnection System," *In Optoelectronic Technology Centre*, Program Review, Cornell University, Sept. 1995.

[21] S. Lakshmivarahan, and S.K. Dhall, "Analysis and Design of Parallel Algorithms Arithmetic and Matrix Problems," McGraw-Hill Publishing Company, 1990.

[22] F. Leighton, "Introduction to Parallel Algorithms and Architectures: Arrays, Trees, and Hypercubes," Morgan Kaufmann Publishers, San Mateo, California, 1992, pp. 399-438.

[23] M.J. Quinn, "Parallel Computing: Theory and Practice," McGraw-Hill, Inc., 1994.

[24] C. Wang and S. Sahni, "Image Processing on the OTIS-mesh Optoelectronic Computer," *IEEE Trans. Parallel and Distributed Systems*, vol. 11, no. 2, 2000, pp. 97-109.

[25] A. Al-Ayyoub and K. Day, "The Hyperstar Interconnection Network," *J. Parallel & Distributed Computing*, vol. 48, no. 2, 1998, pp. 175-199.

[26] S. Das and A. Banerjee, "Hyper Petersen Network Yet Another Hypercube-Like Topology," *Proc. Frontiers '92,* McLean, Virginia, 1992, pp. 270-277.

[27] E. Ganesan and D. Pradhan, "The Hyper-DeBruijn Networks: Scalable Versatile Architecture," *IEEE Trans. Parallel & Distributed Computing*, vol. 4, no. 9, 1993, pp. 962-978.

[28] K. Day and A. Tripathi, "Arrangement Graphs: A Class of Generalised Star Graphs," *Information Processing Letters,* vol. 42, 1992, pp. 235-241.

[29] I. Jung and J. Chang, "Embedding Complete Binary Trees in Star Graphs," *Journal of the Korea Information Science Society*, vol. 21, no. 2, 1994, pp. 407-415.

[30] K. Day and A. Tripathi, "Embedding of Grids and Hypercubes and Characterisation of Spanning Trees in Arrangement Graph," *Proc. Int. Conference on Parallel Processing*, 1993, pp. 56-72.

[31] K. Day and A. Tripathi, "Embedding of Cycles in Arrangement Graphs," Tech. Report TR 91-58, Computer science Dept., Univ. of Minnesota, October 1992.

[32] A. Al-Ayyoub and K. Day, "Matrix Decomposition on the Star Graph," *IEEE Trans. Parallel & Distributed Systems*, vol. 8, no. 8, 1997, pp. 803-812.

[33] A. Youssef, "Design and Analysis of Product Networks," *Proc. 5th Symposium Frontiers of Massively Parallel Computation (Frontiers '95),* 1995, pp. 521-528.

[34] G. Marsden, P. Marchand, P. Harvey, and S. Esener, "Optical Transpose Interconnection System Architecture," *Optics Letters*, 18(13), 1993, pp. 1083-1085.

[35] W. Hendrick , O. Kibar, P. Marchand, C. Fan, D. Blerkom, F. McCormick, I. Cokgor, M. Hansen, and S. Esener, "Modelling and Optimisation of the Optical

Transpose Interconnection System," *In Optoelectronic Technology Centre*, Program Review, Cornell University, Sept. 1995.

[36] F. Zane, P. Marchand, R. Paturi, and S. Esener, "Scalable Network Architecture Using the Optical Transpose Interconnection System (OTIS)," *Journal of Parallel and Distributed Computing* 60(2000), pp. 521-538.

[37] A. Awwad, A. Al-Ayyoub, M. Ould-Khaoua, "On the Topological Properties of the Arrangement-star Network," *The international conference of Parallel and Distributed Processing Techniques applications (PDPTA'99)*, pp. 2165-2170.

[38] C. Wang and S. Sahni, "Basic Operations on the OTIS-mesh Optoelectronic Computer," *IEEE Trans. Parallel and Distributed Systems*, vol. 9, no. 12, 1998, pp.1226-1236.

[39] H. J. Siegel *et al,* "PASM: A Patitionable SIMD/MIMD System for Image Processing and Pattern Recognition," *IEEE Trans. Computers*, C-30, 1981, pp. 934-947.

[40] H. J. Siegel and C.B. Stunkel, "Trends in Parallel Machine Interconnection Networks," *IEEE Comput. Sci. & Eng.*, 1996, pp. 69-71.

[41] S. Sahni and C. Wang, "BPC Permutations on the OTIS-hypercube Optoelectronic Computer," *Informatica*, vol. 22, 1998, pp. 263-269.

[42] A. Al-Ayyoub and K. Day, "Efficient Algorithms on the Hyperstar," *Parallel Algorithms and Applications,* vol. 14, 1998, pp. 79-88

[43] K. Day and A. Al-Ayyoub, "The graph product of Interconnection Networks, A Case Study: Merging the Properties of the Star Graph and the Hypercube," *Journal of Mathematical Modelling and Scientific Computing*, vol. 6, 1996.

[44] P. Fragopoulou and S. Akl, "A Parallel Algorithm for Computing Fourier Transforms on the Star Graph," *IEEE Trans. Parallel & Distributed Systems*, vol. 5, no. 5, 1994, pp. 525-31.

[45] S. Graham and S. Seidel, "The Cost of Broadcasting on Star Graphs and *k*-ary Hypercubes," *IEEE Trans. Computers*, vol. 42, no. 6, 1993, pp. 756-759.

[46] Berthome, P., A. Ferreira, and S. Perennes, "Optimal Information Dissemination in Star and Panckae Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 7, no. 12, Aug. 1996, pp. 1292-1300.

[47] Mendia V. and D. Sarkar, "Optimal Broadcasting on the Star Graph," *IEEE Trans. Parallel and Distributed Systems,* Vo;. 3, No. 4, 1992, pp. 389-396.

[48] S. Rajasekaran and D. Wei, "Selection, Routing, and Sorting on the Star Graph," *J. Parallel & Distributed Computing,* vol. 41, 1997, pp. 225-33.

[49] Efe K. and A. Fernandez, "Computational Properties of Mesh Connected Trees: Versatile Architecture for Parallel Computations," *Proceedings of the 1994 International Conference on Parallel Processing,* vol. 1, 1994, pp. 72-74.

[50] J. Jwo, S. Lakshmivarahan and S. Dhall, "Embedding of Cycles and Grids in Star Graphs," *Journal of Circuits, Systems, and Computers*, vol. 1, no. 1, 1991, pp. 43-74.

[51] Wei-Kuo Chiang and R.-J. Chen, "On the Arrangement Graph," *Information processing Letters*, vol. 66, 1998, pp. 215-219.

[52] K. Day and A. Tripathi, "Characterisation of Node-disjoint Paths in Arrangement Graphs," *Kuwait Journal of Science and Engineering*, vol. 25, 1998, pp. 35-50.

[53] A. Al-Ayyoub and K. Day, "Node-ranking Schemes for the Star Graph," *Personal Communication, 2001.*

[54] Sanjay Ranka, Jhychun Wang and Nangkang Yeh, "Embedding Meshes on the Star Graph," *Proceedings on Supercomputing* '90, November 12 - 16, 1990, New York, USA, pp. 476-485

[55] M. Nigam, S. Sahni and B. Krishnamurthy, "Embedding of Hamiltonians and Hypercubes in Star Interconnection Networks," *Proc. Intl. Conf. Parallel Processing*, 1990, pp. 340-343.

[56] A. Agarwal, "Limits on Interconnection Network Performance," *IEEE Trans. Parallel & Distributed Systems*, vol. 2, 1991, pp. 398-412.

[57] W.J. Dally, "Performance Analysis of *k*-ary *n*-cubes Interconnection Networks," *IEEE Trans. Computers*, vol. 39, no. 6, 1990, pp. 775-785.

[58] G. Yayla, P. Marchand, and S. Esener, "Speed and Energy Analysis of Digital Interconnections: Comparison of on-chip, off-chip, and Free-space Technologies," *Applied Optics*, vol. 37, no. 2, 1998, pp. 205-227.

[59] A. Krishnamoorthy, P. Marchand, F. Kiamilev, and S. Esener, "Grain-size Considerations for Optoelectronic Multistage Interconnection Networks," *Applied Optics*, vol. 31, no. 26, 1992, pp. 5480- 5507.

[60] S. Chatterjee and S. Pawlowski, "Enlightening the Effects and Implications of Nearly Infinite Bandwidth," *Communications of the ACM*, vol. 42, no.6, 1999, pp. 75-83.

[61] S. Sahni, "Models and Algorithms for Optical and Optoelectronic Parallel Computers," *Proceedings of the International Symposium on Parallel Algorithms and Networks*, IEEE Computer Society Press, 1999, pp. 2-7.

[62] C. Wang and S. Sahni, "Matrix Multiplication on the OTIS-mesh Optoelectronic Computer," *Proceedings of the 6th IEEE International Conference on Parallel Interconnects*, 1999, pp. 131-138.

[63] S. Sahni and C. Wang, "BPC Permutations on the OTIS-mesh Optoelectronic Computer," Technical Report 97-008, CISE department, University of Florida, 1997.

[64] M. Angelacci and M. Colajanni, "Unifying and Optimising Parallel Linear Algebra Algorithms," *IEEE Trans. Parallel and Distributed Systems*, vol. 4, no. 12, 1993, pp. 1382-1397.

[65] P. Cappelo, "A Mesh Automaton for Solving Dense Linear Systems," *Proceedings of the International Conference on Parallel Processing*, 1985, pp. 418-425.

[66] Z. Guo, R. Melhem, R. Hall, D. Chiarulli, and S. Levitan, "Pipelined Communications in Optically Interconnected Arrays," *Journal of Parallel and Distributed Computing*, vol. 12, 1991, pp. 269-282.

[67] T. Szymanski, "The Complexity of FFT and Related Butterfly Algorithms on Meshes and Hypermeshes," *Proceedings of the International Conference on Parallel Processing*, vol. III, 1992, pp. 77-81.

[68] M. Angelacci and M. Colajanni, "Subcube Matrix Decomposition: A Unifying View of LU Factorisation on Multicomputers," *Parallel Computing*, vol. 20, no. 2, 1994, pp. 257-270.

[69] A. Al-Ayyoub and K. Day, "Block-cyclic Matrix Triangulation on the Cartesian Product on Star Graphs," *Journal of Computers and Mathematics with Applications*, vol. 36, no. 5, 1998, pp. 113-126.

[70] A. M. Awwad, A. Al-Ayyoub, K. Day, M. Ould-Khaoua, "Generalised Methods for Algorithm Development on Optical Systems," *Proceedings of the 2000 Arab Conference on Information Technology* (*ACIT*'2000), Zarka Private University, Jordan-Zarka, October 31st - November 2nd. 2000, pp. 111-118.

[71] A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "The Topological and Performance Merits of Arrangement-star Networks," Technical Report, University of Glasgow, Scotland, TR-2001-93, June 2001.

[72] C. Wang and S. Sahni, " Matrix Multiplications on the OTIS-Mesh Optoelectronic Computer," *IEEE Trans. Computers*, vol. 40, no. 7, July 2001, pp. 635-646.

[73] D. H. Lawrie and D. A. Padua, "Analysis of Message Switching with Shuffle-exchanges in Multiprocessors," *Proceedings of the Workshop on Interconnection Networks for Parallel and Distributed Processing*, 1980, pp. 116-123.

[74] A. M. Awwad, A. Al-Ayyoub, M. Ould-Khaoua, and K. Day, "Solving Linear Systems Equations Using the Grid Structural Outlook," *Proceedings of the 13th IASTED Parallel and Distributed Computing and Systems (PDCS' 2001)*, Anaheim, USA, August 21-24, 2001, pp. 365-369.

Appendix A

The following program is written in Pascal Language. It will find the performance evaluation of broadcasting cost, for three graphs arrangement-star $AS_{n,m,k}$ = arrangement $A_{m,k}$ = star S_n.

Note that any comments in the program are written between the two parenthesis{ }

==

```
{$N+}

Program Plottings;

Uses WinCrt;

{ This program will find the broadcasting cost for the three graphs based on Seidel
formula }

Const

  e = 0.1; { This is the allowed error ratio which it is 10 percent of the required size }

   increment = 1;

   Max_nmk  = 16;

    MaxSize  = 100000000.0;   { Maximum possible size }

var

    NN, n, m, k, error : Double;

    OutFile          : Text;

Function f (n :double) : double;
```

```
Begin { The following function will find the factorial of any real number }

f := sqrt(2*pi*n) * exp( n*ln( n/exp(1) ) ) * ( 1+1/(12*n)+1/(288*n*n)+1/(n*n*n) );

End;

Function Cost(Size, Degree, Diameter : Double) : Double;

Const { The assumption of the Seidel formula parameters where M, a, and b are
        defined as the message length, unit transmission cost and message latency }

  M = 1024;

  a = 1;

  b = 1000;

var t :double;

Begin

  t := sqrt(a*M/(b*degree))+sqrt(diameter-1);

  Cost := trunc(t*t*b) { The Seidel formula }

End;

Function Star(NN : Double; var n : Double) : Double;

Var Size : Double;

Begin

  n := 1;

  Repeat
```

```
    n := n + increment;

    Size := f(n)

  Until  Size >= NN - error;

  Star := Cost(Size, n-1, trunc(1.5*(n-1))){ The broadcasting cost for the star graph }

End;

Function Arrangement(NN : Double; Var m, k : Double) : Double;

Var

  C, Size, min, minm, mink  : Double;

Begin

  min   := f(Max_nmk);

  m     := 1;

  Repeat

    m := m + increment;

    k := 0;

    Repeat

      Size := f(m) / f(m-k);   { Size of the arrangement graph }

      If Size >= NN - error Then

        Begin

        C:=Cost(Size, k*(m-k), Trunc(1.5*k) );
```

```
        If C < min then
          Begin
            min  := C;
            minm := m;
            mink := k;
          End
        End;
      k := k + increment
    Until k>=m
  Until m>Max_nmk;
m:=minm;
k:=mink;
Arrangement:=min
End;
Function StarArrangement(NN : Double; Var n, m, k : Double) : Double;
Var
  C, Size, min, minn, minm, mink : Double;
Begin
  min   := f(Max_nmk);
```

```
n := 1;

Repeat

 n := n + increment;

 m := 1;

 Repeat

  m := m + increment;

  k := 0;

  Repeat

   Size := f(n) * f(m) / f(m-k); { Size of the arrangement-star network}

   If Size >= NN - error Then { Issued size should be greater than or equal to the
                                required size minus 10 percent of the required
                                size}

     Begin

       C:=Cost(Size, (n-1) + k*(m-k), Trunc(1.5*n) + Trunc(1.5*k) );

       If C < min then

         Begin

           min  := C;

           minn := n;

           minm := m;

           mink := k
```

```
          End
       End;
     k := k + increment
    Until k>=m
   Until m>Max_nmk
  Until n>Max_nmk;
n:=minn;
m:=minm;
k:=mink;
StarArrangement:=min
End;
Function Log2(x:double):double;
Begin  { The above function convert the size to logarithmic form }
  Log2 := ln(x)/ln(2.0);
End;
Begin  {main}
Assign(OutFile, 'c:\all.dat'); Rewrite(OutFile); { The output will be assigned to the
file c:\all.dat }
NN := 1024 {1048576};  { The first size assumed to be 1024, then the next sizes will
                          be increased by double of the previous size }
```

```
While NN<=MaxSize do  { The maximum size is 100000000 nodes }

  Begin

    error := e * NN;

    Writeln(OutFile, Log2(NN):10:0, Star        (NN, n      ):15:0,

                           Arrangement   (NN, m, k  ):15:0,

                           StarArrangement(NN, n, m, k):15:0);

    NN:=NN*2;

  End;

{

      The parameters n, m, and k can be used to find other measures for the three
      networks (star, arrangement, and arrangement-star). For example, you can use
      them to find the network size, broadcasting cost, cost/performance ratio,
      number of links, etc.

      This program generates a text file named ALL.DAT on the root directory of
      your hard disk. Open this file using Excel (give the exact name c:\all.dat of
      the file name box) and try to generate the charts you desire.

}

Close(OutFile);

End.
```

Appendix *B*

The following program is written in Pascal language. It will find the matching probability, for three graphs arrangement-star $AS_{n,m,k}$ = arrangement $A_{m,k}$ = star S_n for a network of exactly NN nodes to exist in the star, arrangement, and arrangement-star.

Hints to understand the program:

All possible sizes for the star, arrangement, and arrangement-star are generated and stored in the 3 lists L1, L2 and L3, respectively. The procedure PUT inserts a given size into a give list only if it is not previously in the given list.

The procedure SORT arranges the sizes in the given list in a decreasing order.

The procedure PROBABILITY receives a list (L of length N) and a network size (NN).

It then finds the number of all sizes in L that are below NN (note that finding this number is easier when L is sorted). Dividing this number by NN will give us the required probability. }

```
{$N+}

Program Scalability_Plottings;

Const

   Maxnmk   = 12;

   MaxSize  = 40e6;
```

```
Type List = Array [1..100] of real;

Var

   NN          : real;

   OutFile      : Text;

   f           : Array[1..Maxnmk] of real;

   i, N1, N2, N3 : Integer;

   L1, L2, L3    : List;

Procedure SortList(Var L:List; N:Integer);

Var i, j : Integer;

     t : real;

Begin

For i:=1 to N-1 do

   For j:=i+1 to n do

      If L[j]<L[i] Then

         Begin

         t:= L[i]; L[i]:=L[j]; L[j]:=t

         End

End;

Procedure Put(Var L:List; Var N:Integer; Element:real);
```

```
Var i     : Integer;

   InList : Boolean;

begin

InList := False;

for i := 1 to N do if L[i]=Element Then InList := True;

If Not InList Then

  Begin

  N := N + 1;

  L[N] := Element;

  End;

End;

Procedure Star;

   Var n : Integer;

    Begin

      for n := 1 to Maxnmk do

           Put(L1,N1,f[n]);

      SortList(L1, N1)

    End;

Procedure Arrangement;
```

```
Var m, k : Integer;
  Begin
    for m := 1 to Maxnmk do
      for k:=0 to m-1 do
        Put(L2,N2,f[m]/f[m-k]);
    SortList(L2, N2)
  End;
Procedure ArrangementStar;
  Var n , m, k : Integer;
  Begin
    for n := 1 to Maxnmk do
      for m:=1 to Maxnmk do
        for k:=0 to m-1 do
          Put(L3, N3, f[n]*f[m]/f[m-k]);
    SortList(L3, N3)
  End;
Function Propability(L:List; N:Integer; NN:Real):Real;
Var i     : Integer;
Begin
```

```
i:=0;

Repeat

i:=i+1;

Until  L[i] >= NN;

Propability:=i/NN

End;

Begin  {main}

Assign(OutFile, 'c:\all.dat'); Rewrite(OutFile);

N1:=0; N2:=0; N3:=0;

f[1]:=1; for i:=2 to Maxnmk do f[i] := f[i-1] * i;

Star;

Arrangement;

ArrangementStar;

NN:=1024;        { This loop goes from 2^10 until 2^25        }

                 { In the figure we plotted only from 2^8 to 2^18 }

While NN < MaxSize do

   Begin

   Writeln(OutFile,ln(NN)/ln(2):20:0, Propability(L1,N1,NN):20:10,

                        Propability(L2,N2,NN):20:10,
```

```
                    Propability(L3,N3,NN):20:10);

  NN:=NN*2;

  End;

Close(OutFile);

End.
```

Publications During Work:

1- A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "On the Topological Properties of the Arrangement-star Network," *Proceedings of the 1999 International Conference on Parallel and Distributed Processing Techniques and Applications* (*PDPTA'* 99), pp. 2165-2170.

2- A. M. Awwad, A. Al-Ayyoub, K. Day, and M. Ould-Khaoua, "Generalised Methods for Algorithm Development on Optical Systems," *Proceedings of the 2000 Arab Conference on Information Technology* (*ACIT' 2000*), Zarka Private University, Jordan, October 31st - November 2nd, 2000, pp. 111-118.

3- A. M. Awwad, A. Al-Ayyoub, M. Ould-Khaoua, and K. Day, "Solving Linear Systems Equations Using the Grid Structural Outlook," *Proceeding of the 13th IASTED Parallel and Distributed Computing and Systems (PDCS' 2001)*, August 21-24, Anaheim, USA, 2001, pp. 365-369.

4- A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "On the Topological Properties of the Arrangement-star Network," *under review*, *Journal of Systems Architecture*, 2000.

5- A. Al-Ayyoub, A. M. Awwad, M. Ould-Khaoua, and K. Day, "Generalised Methods for Algorithm Development on Optical Systems," *under review, Journal of Parallel and Distributed Computing,* 2000.

6- A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "On the Algorithmic Issues of the Arrangement-Star Network," accepted, *The 2001 Arab Conference on Information Technology* (*ACIT'2001*), Jordan University of Science and Technology, Nov. 13–15, 2001, pp. 9-14.

7- A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "The Topological and Performance Merits of Arrangement-star Networks," Technical Report TR-2001 - 93, Department of Computing Science, University of Glasgow, June 2001.

8- A. Al-Ayyoub, A. M. Awwad, M. Ould-Khaoua, and K. Day, "Efficient Methods for Algorithm Development of Optoelectronic Networks," Technical Report TR-2001-95, Department of Computing Science, University of Glasgow, August 2001.

9- A. M. Awwad, A. Al-Ayyoub, and M. Ould-Khaoua, "Routing in OTIS-Networks," in preparation, 2002.