“DESIGNING AN INTERCONNECTION NETWORK USING GENETIC ALGORITHM”

Project Report submitted in partial fulfillment of the requirement for the degree of

Master of Technology

in

Computer Science &Engineering

under the Supervision of

Dr. NITIN

By:

Aradhya Saini

Enrollment no: 132226

Jaypee University of Information Technology

Waknaghat, Solan – 173234, Himachal Pradesh

Certificate

This is to certify that project report entitled "Designing an Interconnection Network Using Genetic Algorithm", submitted by Aradhya Saini in partial fulfilment for the award of degree of Master of Technology in Computer Science to Jaypee University of Information Technology, Waknaghat, Solan has been carried out under my supervision.

This work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Date: **Signature:**

Supervisor's Name: Dr. NITIN

Designation: Associate Professor
Department of CSE
JUIT, Waknaghat
Solan(H.P.)

Acknowledgement

Foremost, I would like to express my sincere gratitude to my advisor **Dr. NITIN,** Associate Professor, Department of Computer Science & Engineering, Jaypee University of Information Technology, Waknaghat, Solan (H.P.) , for the continuous support and guidance in my thesis study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance has helped me at all times of my research and writing of this thesis. I could not have imagined having a better advisor and mentor for my thesis study.

I would also like to thank him for sharing his vast expanse of knowledge in guiding me with the correct books, research issues to study without which I wouldn't have been able to proceed so much further in my topic on Interconnection Networks. He always spared his valuable time and helped me in striving to move forward at every point.

I would like to convey my thanks to **Mr.Akash Punhani** for his precious time and efforts that he has put in, in providing guidance to this research at every step.

Last but not the least, I would like to pay sincere respect and gratitude towards my family for staying by me through thick and thin and for showing their confidence in me that motivated me throughout this process.

Date: Name of the student: Aradhya Saini

Signature:

Table of Contents

List of Figures

List of Tables

S.No. **Title** **Page No.**

List of Abbreviations

AAB	All-to-All Broadcast
AAG	All-to-All Gather
AAR	All-to-All Reduce
AAS	All-to-All Scatter
ACO	Ant Colony Optimization
CC	Collective Communication
EA	Evolutionary Algorithms
EDA	Estimation of Distribution Algorithm
GA	Genetic Algorithm
HGSA	Hybrid parallel Genetic Simulated Annealing
MBOA	Mixed Bayesian Distribution Algorithm
MIN	Multistage Interconnection Network
OAB	One-to-All Broadcast
OAS	One-to-All Scatter
UMDA	Univariate Marginal Distribution Algorithm
WH	Wormhole Routing

Abstract

This thesis describes the technique of evolutionary design which is aimed at designing an interconnection network. An Interconnection network is the communication path used in any digital subsystem and it is through them that the data is transferred between any two nodes. There is always willingness for obtaining shortest path between any source-destination pair. There are various performance parameters that need to be looked upon while constructing such paths in the network. Taking well care of these parameters leads to enhancement in the performance of the network. The parameters range from bisection width to cost to throughput. Therefore for obtaining optimal results, we need to take the performance analysis into consideration. For this purpose, here we construct the desired interconnection network by maximizing its bisection bandwidth and minimizing the cost of the network. However we know cost requires degree of the node, degree is given by number of links and bisection width is also dependent on the number of links. In this chaining, maximizing one and minimizing another leads to a tradeoff between the bisection bandwidth and the cost of the interconnection network. Now as we have to solve this difficult optimization problem, we do so by using Genetic Algorithm (GA). It indeed helps a lot in solving this tradeoff and constructs the network constrained to the parameters mentioned. As we turn to use Genetic Algorithm for optimization, we consider it much capable to solve such optimization problems which are humanly not possible to solve. The various GA operators like selection, crossover and mutation are applied which leads to the formation of new population on each iteration. The selection is carried out on each one of the chromosomes of the population based on the fitness function and thereby the best amongst them is considered. The offspring's are obtained first through the crossover and then corresponding mutation operator is performed to rectify them. This is carried out for specified number of generations after which the fittest chromosome obtained is simulated on OMNeT++.This all leads to interpretation of that fittest chromosome as the interconnection network which is designed constrained to the performance parameters, optimized by the genetic algorithm and thus providing us with the optimized design results.

CHAPTER 1

INTRODUCTION

In this section we take into consideration the problem definition. This clearly defines the goal that will be worked upon in the rest of the thesis. Apart from this the discussion is carried out on the motivation behind choosing this topic is. This section also includes the thesis organization along with the description made regarding the contributions to the society.

1.1 Problem Definition

Designing an interconnection network with the following constrained parameters:

- Maximizing the bisection bandwidth
- Minimizing the cost

Based on the genetic algorithm which comprises of efficient encoding, crossover and mutation rate to ensure a better rate of convergence towards optimal solution .

1.2 Motivation

A recent trend has taken to make efficient usage of scarce interconnection bandwidth that strives to move forward in order to make interconnection networks which then serve as the universal solution for various problems in modern digital systems. This has lead to switching of more and more systems from using dedicated wiring and bus- based switch to interconnection networks every year. Buses lack behind as apart from not working well when scaled to large number of communication partners, the high bandwidth demands also haven't been met. Even in dedicated wiring the attenuation of 24-gauge copper wire gets no better, speed of light remains same and routing of packets is anyway better than routing wiring [1]. As interconnection networks are used for communication between subsystems so they need to be fast and pace up their performance in terms of network bandwidth and processor performance. This draws the inference that the interconnection networks have gradually hit huge success as the communication medium in digital systems.

The performance of an interconnection network can be measured using three parameters: cost, bandwidth and latency. Even the parameters are inter-related to each other. The cost depends on the diameter and degree and diameter is in turn dependent on the number of links. As we take bisection bandwidth we observe that this is also dependent on the number of links. Therefore optimizing the cost and bisection bandwidth concurrently is a task that yields to design of the constrained interconnection network. As managing both cost and bandwidth is a tedious task therefore this is done optimally using evolutionary algorithm. An evolutionary design suffices to produce optimal or for that matter approximate optimal communication schedules which are in accordance to the networks sizes which are of our interest. In this thesis work, we take into consideration the constrained parameters as maximization of bisection bandwidth and minimizing the cost in order to produce an interconnection network where optimization is carried out using genetic algorithm.

1.3 Thesis Organization

This thesis document comprises of five chapters. Chapter 1 states the objective of the thesis along with the motivation behind choosing this thesis topic. Chapter 2 comprises of the relevant history read to understand the basics along with an introduction to the basics required in relevancy to the topic on interconnection network and genetic algorithm. Chapter 3 highlights the proposed approach that has been carried out so as to design the required interconnection network using the genetic algorithm. Chapter 4 comprises of the results that have been obtained on applying the genetic algorithm on interconnection network and thus testing its efficiency. Chapter 5 follows the summary of the thesis and some future points jotted down on which the work can be carried on further.

1.4 Contributions

The interconnection networks are said to be used in delivering high – performance or carrying out parallel computation as a low- latency interconnection. They are used in various applications like in routing, sorting etc. This is basically because these require data to be transferred from one node to another quickly in a very short span of time or the numerous heavy computations being done in parallel really fast so as to produce the results quickly. For this purpose our objective works well. It provides with low latency interconnection network so to transport the data between any two node pairs of source-destination or the results being computed accordingly really quick.

Like for example interconnection network being used in digital computers which are used for management of large databases or simulation of the physical systems. The performance of such digital systems is affected by the communication channel used. Therefore for the efficiency to enhance we have designed such an interconnection network that provides with lower latency at any point in time so that the communication can be faster. This thus leads to the large amount of data being computed concurrently at a much faster pace. The performance is well taken care of when constrained parameters are considered. The optimization for the interconnection network is carried out using the genetic algorithm which is well considered to give good results when it comes to creation of the network. It is of great help at every point as the path traversal according to this determined interconnection network observes lesser latency.

CHAPTER 2

LITERATURE SURVEY

In this chapter we very briefly outline the topics that are prerequisites to our topic. The discussion regarding collective scheduling and evolutionary algorithm is carried out in order to familiarize with the methods that can be helpful enough to gain a powerful insight to our approach. The discussion made on cost based method is done so as to find out how the working is carried out. Apart from this comparative analysis helps to reach a decision of which algorithm is best used in bisection. This even takes the background into consideration so that we have an introduction to the topics and keywords that we are going to refer to in the rest of the work. This includes brief description of the terminologies used in interconnection network and genetic algorithm.

2.1 PRELIMINARIES

2.1.1 Collective Communication Scheduling Algorithm [3]

The objective that bounds is designing of optimized collective communication (CC) schedules which help in boosting the performance and avoiding of link contentions.

- The implementation of CC Scheduling algorithm is done using the method of Univariate Marginal Distribution Algorithm (UMDA) which is considered to be a very simple EDA.

- In the wormhole-switched network, the broadcast communication (OAB) is considered to be not less than s steps, where we have $s = \lceil \log_2 N \rceil$ to be the count of nodes that have been updated about in each of the steps.

- In case of OAS communication, as we have that each node has the property that no more than one message can be injected at one time where the lower bound is N-1 steps.

- An optimal OAB schedule is shown in Figure 1 designed for 8-node Omega. In case of OAB, each of the chromosomes considered comprise of N genes, one for each present destination node. Three items constitute individual genes: a source node index, an index of the used path, and a step number as shown in Figure 2.

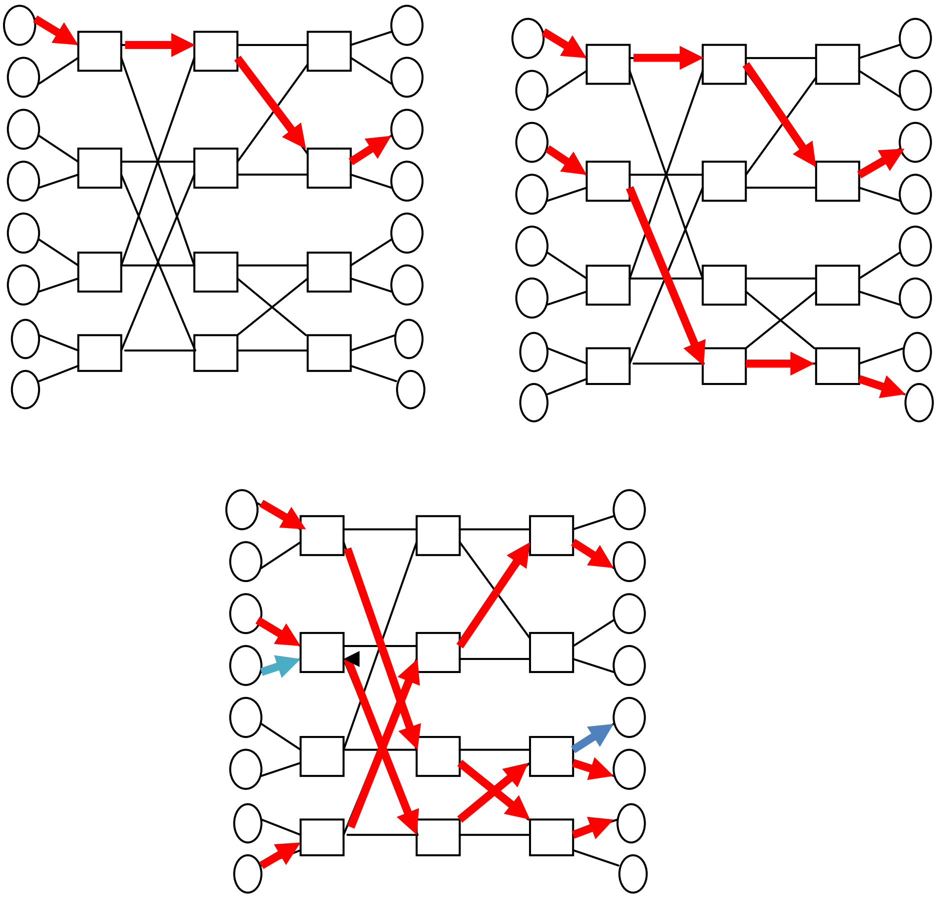


Figure 1 An OAB schedule reaching the lower bound on number of communication steps

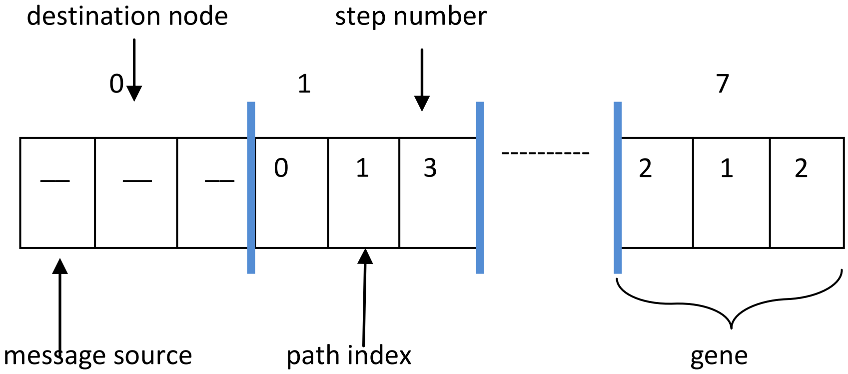


Figure 2 Structure of OAB chromosome

In OAS, the chromosome contains N genes. Here two items are present in a gene: a utilized path which is considered to be the first component and the second component is the used time step.

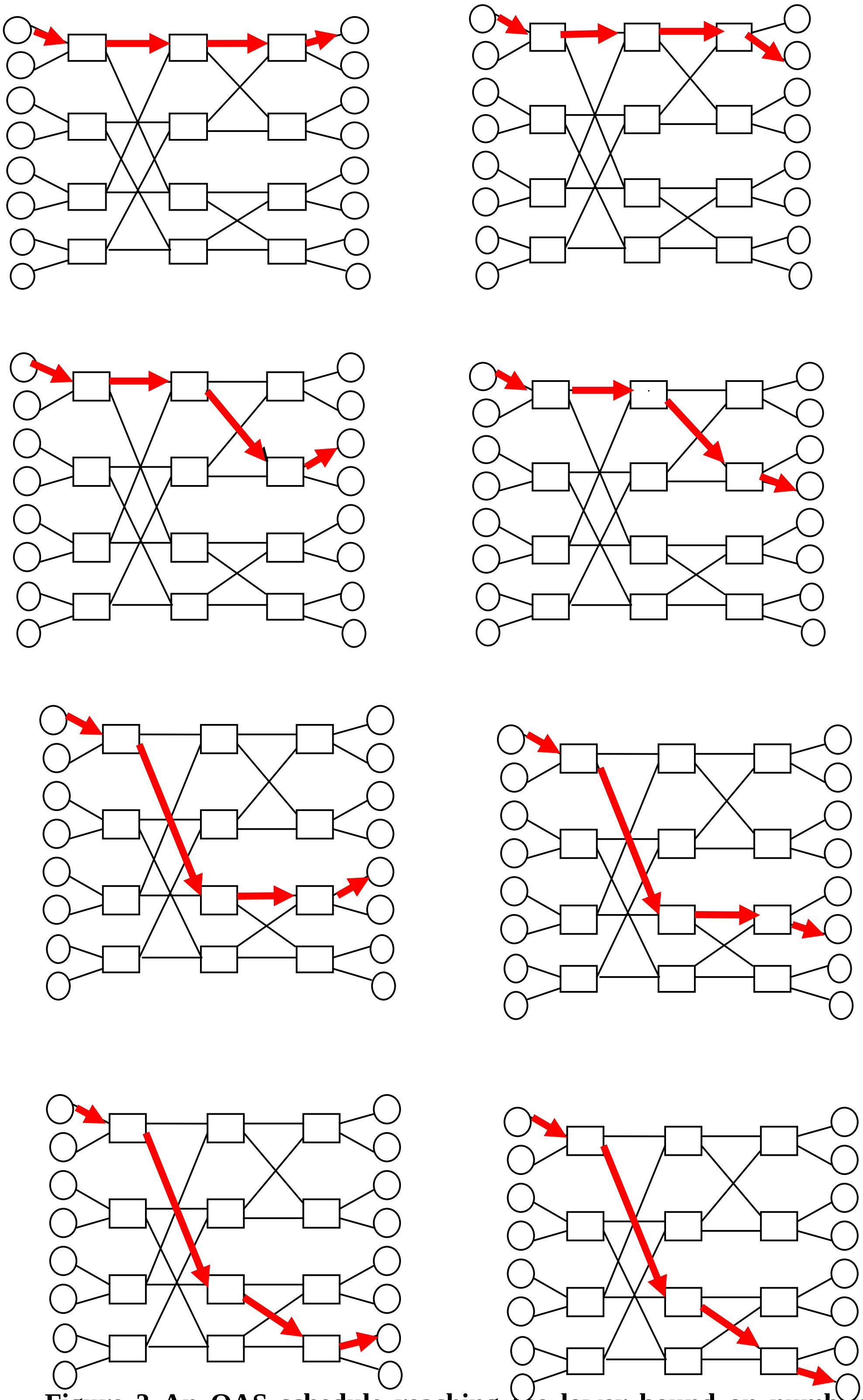


Figure 3 An OAS schedule reaching the lower bound on number of communication steps.

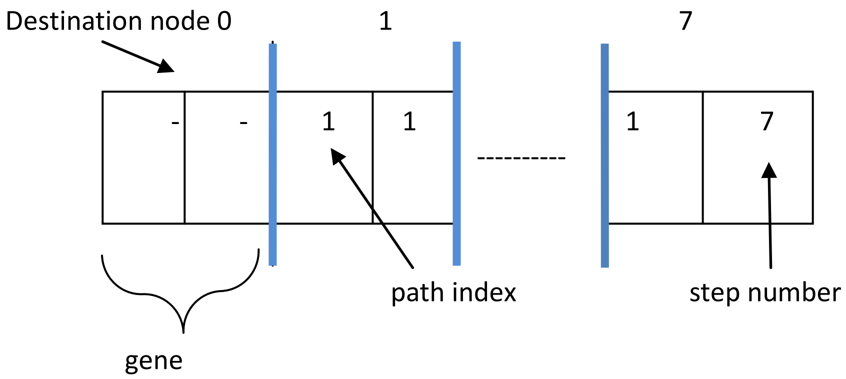


Figure 4 Structure of OAS chromosome

- Optimized communications schedules can be put up in switch routing tables and help provide good returns in case of many parallel algorithms.

2.1.2 Evolutionary Algorithm –HGSA and MBOA [4]

Two evolutionary algorithms aim at scheduling of collective communication on interconnection networks present in parallel computers.

2.1.2.1 Many-to-many broadcast by means of HGSA

- Hybrid parallel Genetic Simulated Annealing (HGSA) has been used to find out MNB schedules.

- Parallel Simulated Annealing (SA) along with the operations that are used in standard genetic algorithms is used by HGSA which is a hybrid method.

  The solution obtained by SA and from genetic manipulation helps in selecting a new solution for each process.

- T he selection is controlled by quite popular Metropolis criterion.

- A method of sampling a Boltzmann distribution is the well-known Metropolis algorithm.

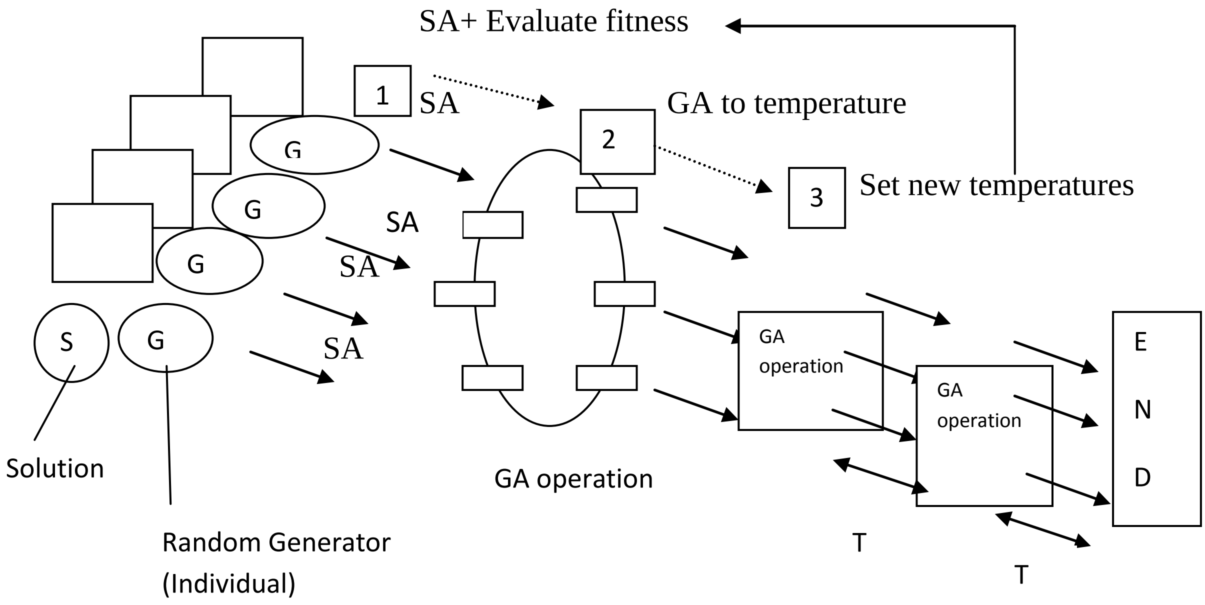


Figure 5 Hybrid parallel Genetic Simulated Annealing

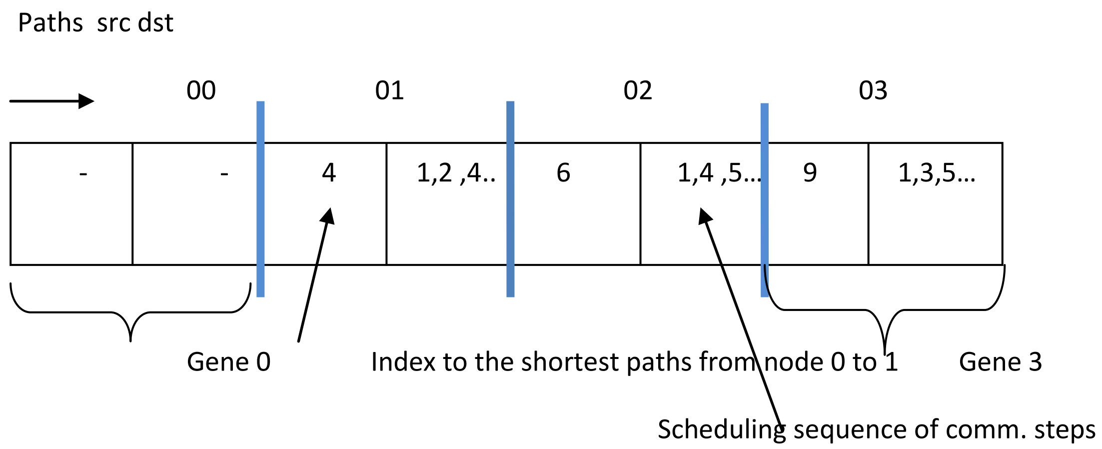


Figure 6 Structure of chromosome

2.1.2.2 Scheduling broadcast and scatter communication by means of MBOA

- Here we have a general procedure of MBOA that comes as an offspring under Estimation of Distribution Algorithm (EDA).
- Probability sampling follows probability estimation.
- These algorithms have to their benefit, the statistical information which is present in the collection of candidate solutions that are used to find out connection between genes.

- In the case of OAB with wormhole switching there is presence of each chromosome consisting of *P* genes, one present for each of the destination node.

- Individual genes are said to comprise of three things present within them: a source node index, the shortest path index, and a step number.

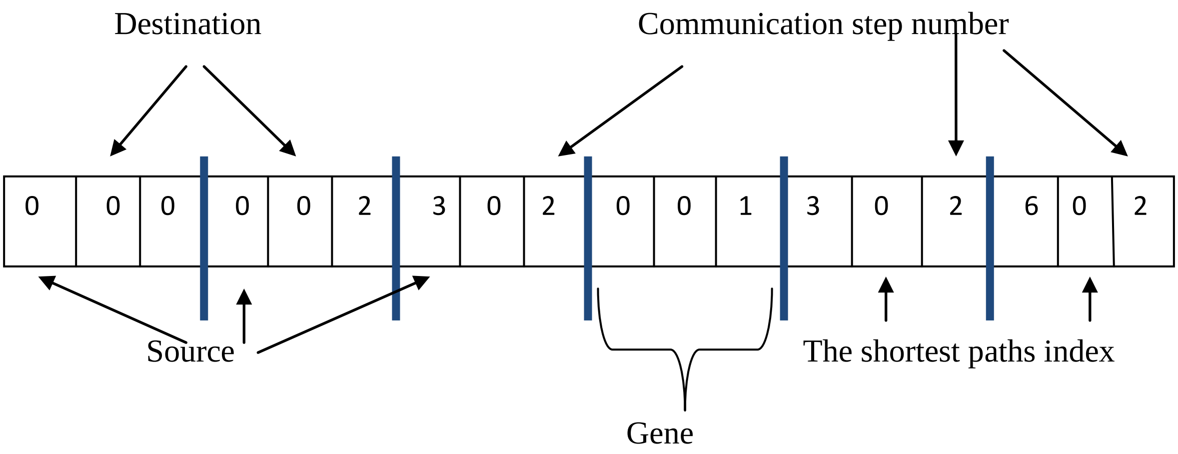


Figure 7 Structure of OAB WH chromosome

- The OAS chromosome uses P genes.

- Each gene consists of two items: an index which is meant for one of the shortest source destination path and a communication step number**.**

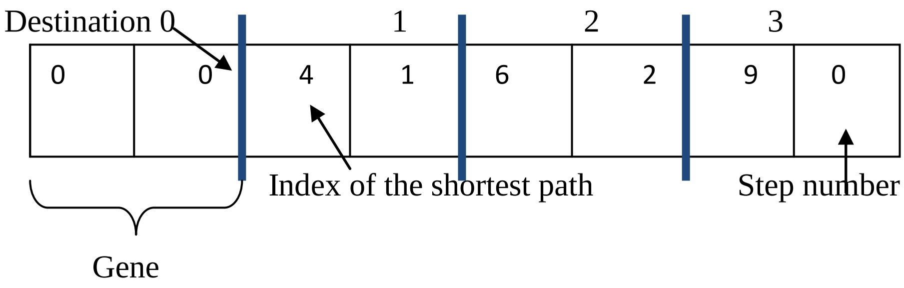


Figure 8 Structure of OAS WH chromosome

2.1.3 Genetic Algorithms for Switched Ethernet Industrial Networks [6]

- We aim to use the GA for the optimization of a switched Ethernet topology.
- To maximize the intra group exchanges and minimize the inter-group dialogues.

2.1.3.1 Graph Partitioning Problem:

Partitioning the vertices is carried out into disjoint subsets for a given graph with vertex and edge weights which gives that the summation of the vertex weights of each of the subset is much close to the average sum as well the total cost, of the edges used for the connection of vertices that are present in different domains is minimized.

- Each chromosome symbolizes a vector in which the i^{th} *element* of an individual is *j if the* i^{th} *vertex of the* graph is allocated to the partition labeled j .
- There are as many elements in the vector that has been encoded as vertices in the graph.

The proposed algorithm as follows [6]:

Create initial population

Do {

Choice of parent1 and parent2 is made from population

Computation of offspring=crossover (parent1, parent2)

Computation of mutation (offspring)

If adapted (offspring)

then replacement(population, offspring)

}

Until (stopping criterion)

Report the best result.

2.1.4 Topological Optimization of Interconnection Networks at focusing minimization of the overall cost [5]

- Presented a genetic algorithm to design telecommunication networks.
- Minimization of the network performance and design cost.
- The algorithm is studied as follows[5]:

Optimize network

Input: (ICN,G,Cost_mat,Rel_mat, $Cost_{max}$)

1. Generation of initial population

2. While (steady state has not been reached) do

   2.1 Apply selection process with crossover probability so as to select chromosomes into mating pool.

   2.2 Apply crossover in order to get a new solution network.

   2.3 In order to ensure two connected network the repair algorithm is applied.

   2.4 Find fitness using fitness function F(X) is computed.

   2.5 Select the fittest chromosome so as to communicate to the next generation.

3. End while

4. Declaration of the chromosome having minimum cost.

The abbreviations are considered as follows:

- F(X) Fitness of the network x
- $Cost_{max}$ Maximum permitted Cost constraint
- ICN Completely connected interconnection network
- G(N,L) Graph G with N number of nodes and L is the number

of links.

- The fitness function is considered here as:

$$F(X) = Max\_ \ Cost\ (P) - Cost(X) \qquad (2.1)$$

where Max _Cost (P)= Maximum network cost of a population P.

Cost(X) = Cost of every network x.

- Graph G has N as the number of nodes which can be used to denote the terminals or the computers) and L is count of the links among them.
- The variable length integer encoding is considered having string representation.
- The described parameters are considered so as to describe how the network is constructed.
- The cost matrix is considered beforehand the computation.
- A repair algorithm is considered here as the mutation operator to which allows addition of some links in order to maintain the minimum 2-connectivity criteria [5].
- The reliability constraint that is considered here is :

$$\textbf{Minimize } Z= \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} C_{ij}\ x_{ij} \qquad (2.2)$$

Subject to $R(X) >= R_0$

Table 1 Literature Survey

Paper Details	Proposed Solution	Fitness Function
A New Genetic Algorithm based Method for Topological Optimization of Interconnection Networks , *IJCA*,2013	• **Encoding Technique** Variable length integer encoding string representation • **Selection** Roulette wheel • **Crossover** Single point crossover • **Mutation** Repair algorithm with 2 pair connectivity	F(X)=Max _Cost(P)-Cost(X)
Using Genetic Algorithms to Design Switched Ethernet Industrial Networks, IEEE,2002	• **Encoding Technique** N-string Representation • **Selection** Roulette wheel • **Crossover** Two point Cut crossover • **Mutation** Swap Mutation	$f(e, e_v, e_o) = (f-e_0)/(f+e_v)$
Evolutionary Optimization of Multistage Interconnection Networks Performance, ACM, 2009	• **Encoding Technique** Direct encoding • **Selection** Roulette wheel • **Crossover** Single point crossover • **Mutation** Repair algorithm with 2 pair connectivity	Basis is counting of conflicts present between all point-to-point communications which are carried out in the same steps.
An Evolutionary Approach to Collective Communication Scheduling, ACM,2007	• **Selection** Roulette wheel • **Crossover and Mutation(in HGSA)** As used in Genetic Algorithm • **Crossover and Mutation (In MBOA)**: Probability estimation followed by probability sampling	Computation is done taking into consideration the number of conflicts in the schedule (in HGSA).

Table 2 Comparative Study of Partitioning algorithms [24]

Partitioning algorithms	Basics	Description	Disadvantages
Constructive Partitioning Improvement	It uses seed growth or cluster growth. The objective needed to be fulfilled determines the use of gain functions.	1. It basically begins with the consideration of a new partition choosing seed logic cell. 2. It even takes the calculation of gain function into consideration thus measuring the profit of addition of a logic cell to the current partition.	1. There is a limitation to what can be achieved with a partitioning algorithm which is based on simple exchange. 2. Only one change is considered by simple exchange algorithm in case there is no improvement and then rejects it out rightly. 3. Such algorithms are greedy algorithms such that that move is accepted only if it provides an immediate profit . 4. This leads it to a local minimum from which it one cannot get away.
Iterative Partitioning Improvement	Based upon interchanges occur along-with group migration. In order to improve the partition the process of exchanging logic cells is carried out. The trail exchange is accepted in case the swap improves the	1. Comprises of swapping of sets of logic cells that are present connecting partitions. 2. Group migration algorithms are way better than simple exchange methods when it comes to improving a	All the more complex as swapping is done in terms of group swapping.

	partition, otherwise a new group of logic cells is selected to swap.	solution. 3. Based upon powerful Kernighan–Lin Algorithm.	
Kernighan–Lin Algorithm	Iterative as well as group migration is also considered.	1. Partitions must be equal size resulting in only two partitions. 2. Goal contains swapping nodes with goal of minimizing the count of edges that connect the two halves. . 3. Important feature is consideration of moves even though this leads to the tendency to make things worse.	1. Logic cells of different sizes are not allowed. 2. Unequal partitions are not allowed. 3. No more than two partitions. 4. N^2 dependency creates a problem for partition of large network.
The Ratio-Cut Algorithm	The constraint of steady partition size is removed.	1. Small, Highly connected networks are formed using ratio cuts. 2. Size of a partition formed equals the count of nodes that is present.	Formation of a reduced network is carried out from the highly connected groups (forms group of logic cells) which lead to the formation of a node to be present in the condensed network whereby using F-M algorithm. This can be used for improving the reduced network.

Look-ahead Algorithm	The first-level gain is the gain for the initial move, subsequent moves are second level gain and so we use a gain vector	Gain vector provides us using a look-ahead algorithm to be used in the choosing nodes that are have be swapped which thereby leads to reduction of both the mean and variation in the number of cuts which are present in the resulting partitions.	Requires before -hand look-ahead therefore increases computation.
Simulated Annealing	In Simulated annealing a series of random moves are successively used to change an existing solution.	Acceptance or rejection of an energy function is calculated for each of the new trial configuration. The best solution so provided is the global minimum. The probability of acceptance of a certain poorer configuration is said to be determined by the exponential expression: exp (–D E / T), where we have D E as the resulting increase that occurs in the energy function.	1. Requires waiting a long amount of time to find the optimum result. 2. In interchange strategy, only when there is a decrease in the energy function then only the new trial configuration is said to be accepted. However, in case of the simulated-annealing algorithm, there is acceptance of the new configuration in case the energy function increases which means things are getting worse.
Fiduccia–Mattheyses algorithm	Used to implement a net-cut partitioning and does reduce the computational effort.	The F–M algorithm takes logic-cell size into consideration and so it does select a logic cell so as to swap based upon maintenance of balance to be taken care of amongst the total cell size taking	F–M algorithms takes into consideration only the immediate gain that is to be made by movement of a node.

		into account each of the partitions.	

Table 3 Analysis for regular topologies

Network	Degree	Diameter	Cost	Bisection Width
Mesh	4	2√n-1	8√n-1	√n
Torus	4	√n-1	4√n-1	2√n
Hypercube	Log n	Log n	Log^2n	n/2

<u>PARAMETERS:</u>

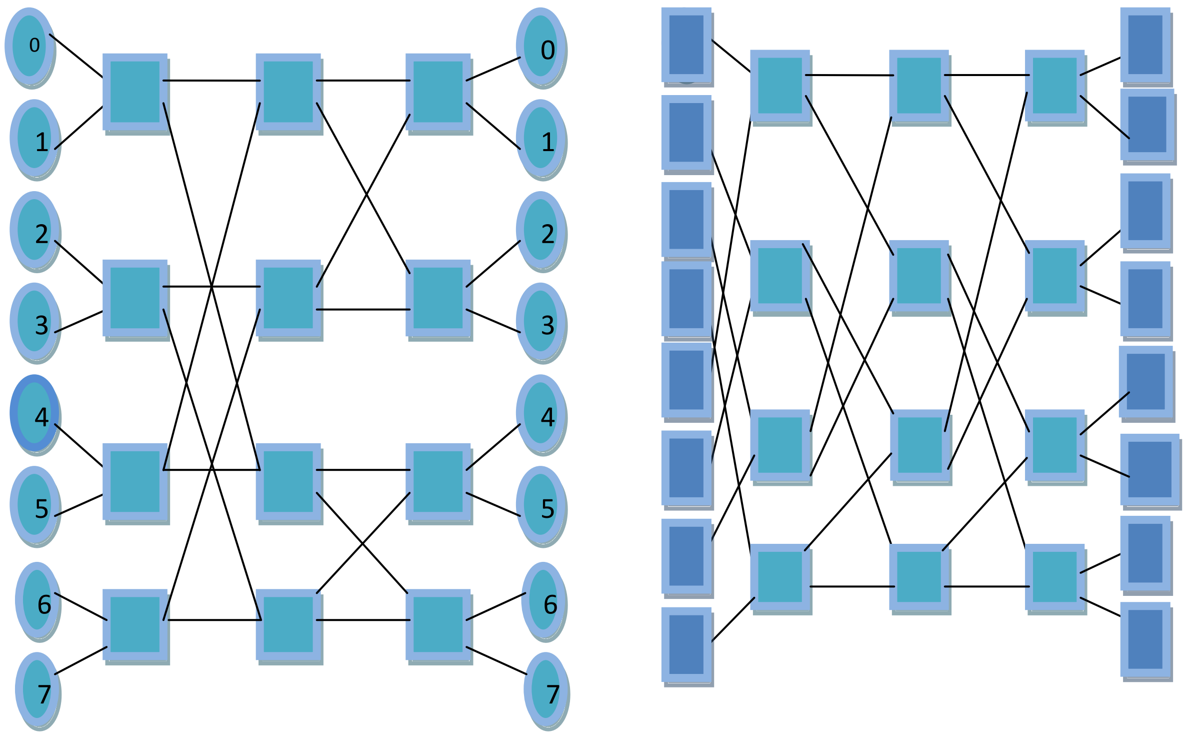
- Degree [1] - No of channels entering and leaving each node.
- Diameter - the minimum distance between two farthest node that are source and destination.
- Cost [9] - defined as the product of degree and diameter
- Bisection width[7] - smallest number of links which have to be cut so as to divide the network into two equal parts.

The discussion is on the regular topologies as we can see in Table 3. However the properties need to be looked after in case of irregular topologies also. The properties for each of the interconnection network present determine the way they can be used for analysis and eventually for construction of the network. For this very purpose we take into consideration the following Table 4 as there is need for evaluation.

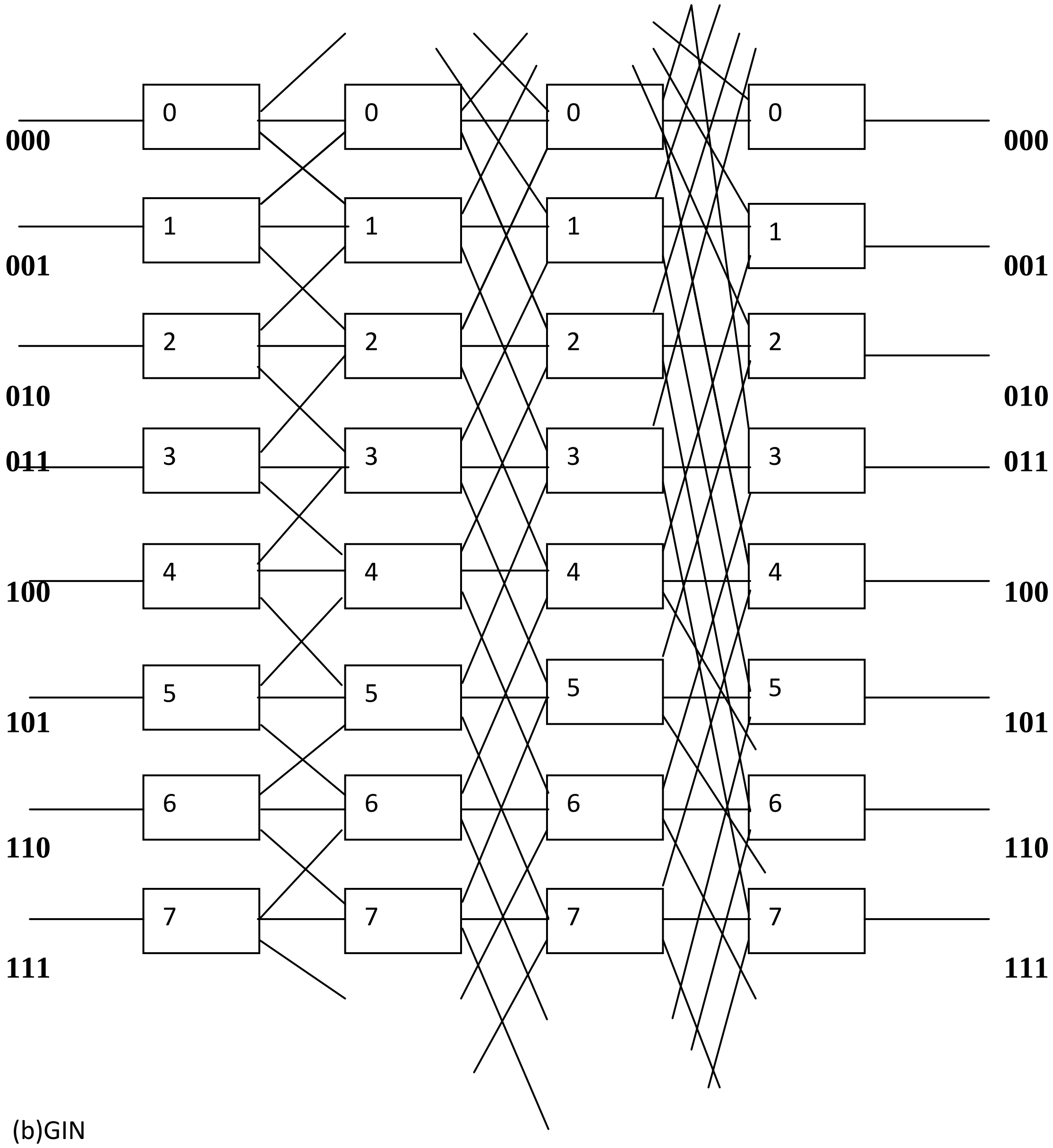
Table 4 Analysis for regular topologies [10]

Network	Stage 0	Stage 1	Stage 2	Stage 3	Total Cost= $\Sigma(I \times O \times S)$
Butterfly (2-ary 3-fly)	2*2*4	2*2*4	2*2*4	-	48
GIN	1*3*8	3*3*8	3*3*8	3*1*8	192
Omega	2*2*4	2*2*4	2*2*4	-	48

- I - input pins
- O - output pins
- S - switches



(a) A 2-ary 3-fly (c) Omega



(b)GIN

Figure 9 Different Topologies under analysis

2.2 Background

There is need of dedicated path between any two terminal pairs that are considered as a source-destination pair and so as to serve this purpose, interconnection network is implemented using shared router nodes along with shared channels. The path according to which they have been arranged is the network's topology. Here we move forward with the basics of network topology. We also take into account the prerequisites for evolutionary and genetic algorithm.

2.2.1 Network Basics

The designer implements topology, routing and flow control working within the technology constraints. In interconnection networks we can even consider connections between nodes with each node having its own processor, memory and associated router.

Take a simple analogy of cars being driven on a road. The intersections and the traffic signals are the shared routers. The connection pattern between these nodes and channels is known as the network's *topology*. There can be many ways to reach the destination. *Routing* determines the so many possible paths that can be chosen in order to route the packets. The channels considered to be like roads are used to carry packets which are like cars driven from one router node (i.e. intersection) to another. While balancing is carried out of the demand on shared resources that are present in the network, a good choice of paths leads to minimization of the length i.e. the count of the nodes that are present on the hops.

Flow control does up to some point dominate the priority in terms of which message will be getting access of the resources of the particular network over the given time period. As the utilization of resource moves upwards flow control becomes more crucial whereby flow control is also influenced. We even have the fact jotted down that packets with minimum delay are forwarded in case of good flow control and therefore helps in avoiding idling resources under high loads.

2.2.1.1 What is an interconnection network?

An interconnection network can be well defined as group of connections between the various channels and routers. This can be found to span from system level to network level. It is found connecting processors to memory, I/O devices to I/O controllers and Input ports to Output ports[2]. The digital systems consider interconnection network for transporting data between subsystems, which is besides the memory and logic being performed.

2.2.1.2 Why are interconnection networks important?

Interconnection networks have been very rapidly taking over the space of dedicated wiring and buses. Driving of wires is done using the power that is present along with the gate delay not being considered, whereas clock cycle is said to be consumed on wire delay [2]. Even though the memories and processors are becoming fast, there is no change in the speed of light. Scaling of pin density and wiring density is at s lower rate than the system components whereby frequency is still way far behind the rates of the clock. This all leads to interconnection networks playing a pivot role in the success of systems.

2.2.2 Topology

Topology can be considered as the arrangement of the nodes and the channels. The following figure shows the major two categories of topologies .They are *direct* and *indirect* topology. In direct topology every node can be a terminal as well as a switch node whereas in indirect topology every node can either be a terminal or a switch node [2]. Figure 10 depicts it well.

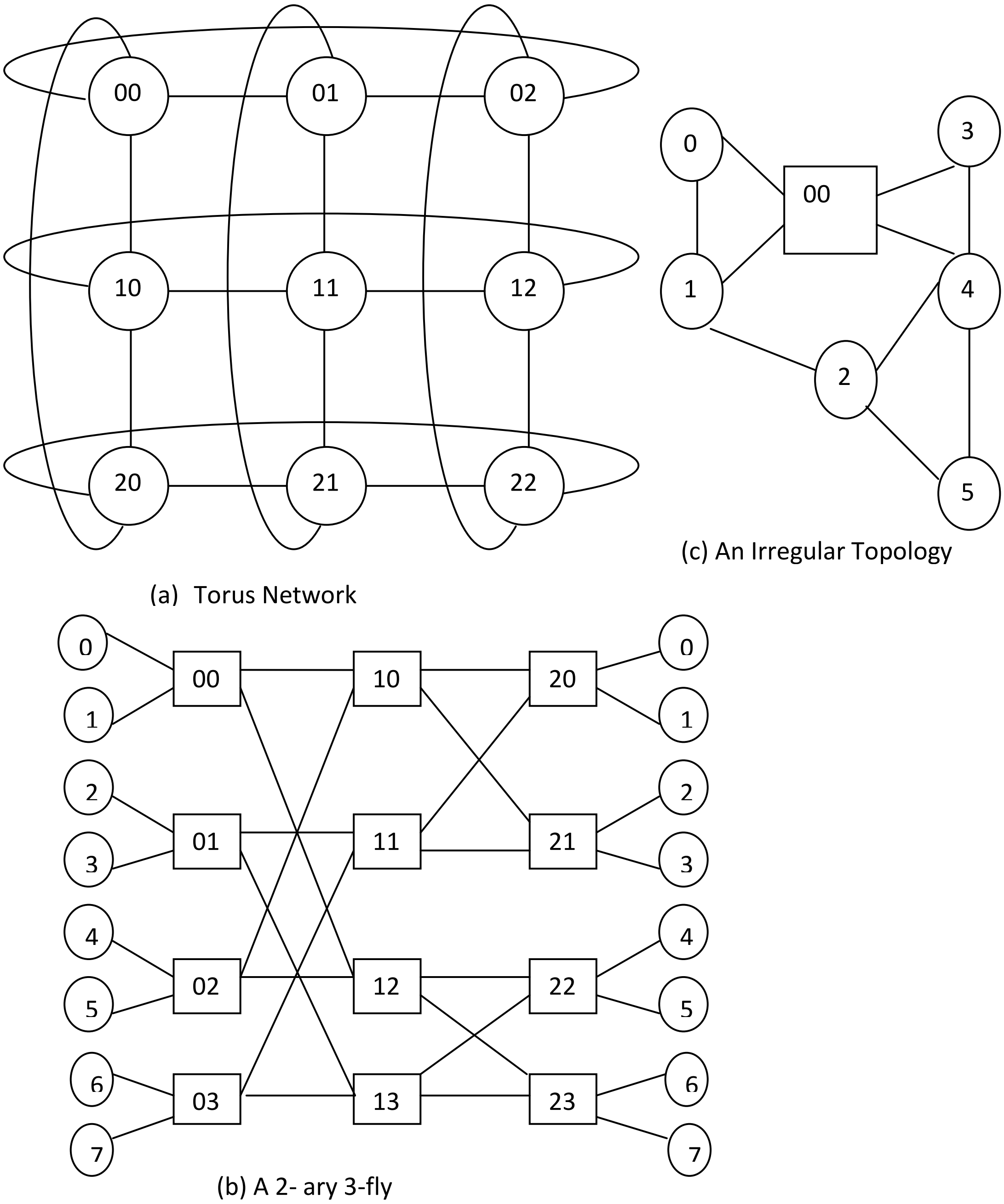


Figure 10 Network Topologies

Taking another example, the network shown in Figure 11 is a *torus (direct)* topology consisting of 16 nodes, which comprises each node connected to 8 channels, 1 to each neighbor and 1 from each neighbor. In each direction, the nodes are denoted by circles and each pair of channels is denoted by a line joining two nodes.

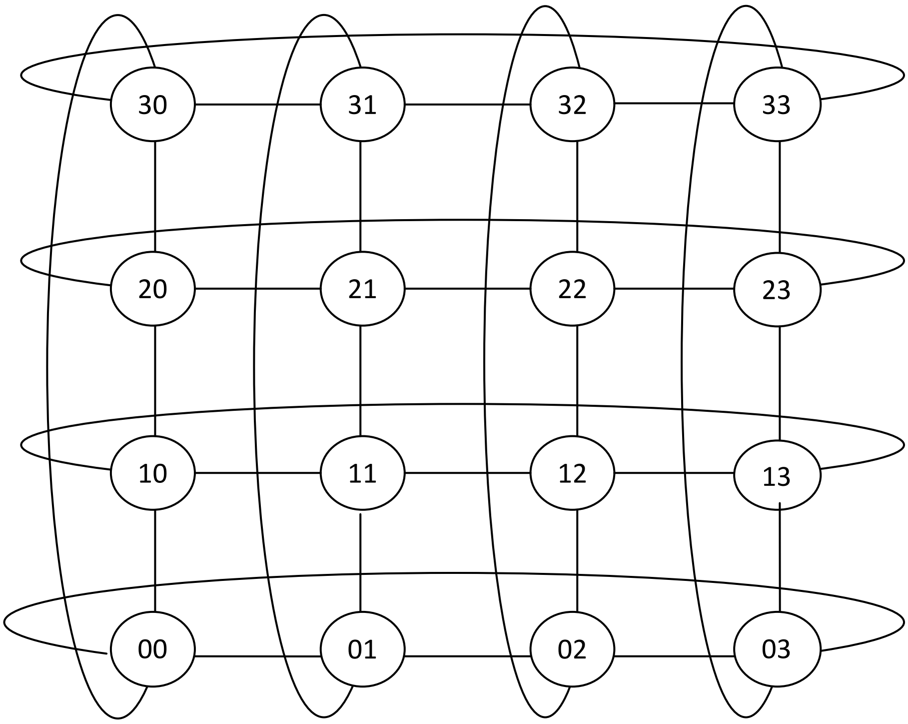


Figure 11 A network topology is the arrangement of nodes

2.2.3 Routing

The path that the packet will follow to transfer the data from the source to destination .There can be innumerable of them and one is chosen to be followed till the destination. In all the routing algorithms, the source-destination pair so chosen is single fixed path even though alternatives are also present. This is a *deterministic routing algorithm* which does get subjected to low bandwidth.

2.2.4 Flow Control

A fair strategy that avoids deadlock is a good flow control strategy. Buffers are present within the nodes like memories and registers and are implemented as storage to hold the packets in intermediate stages. For performance potential realization of the topology and routing method for control flow strategy, there needs to be avoidance of resource conflicts that holds a channel idle. There can be blockage of a packet that can use an idle channel just because it is waiting on a buffer that is being held by a packet that has been blocked on a busy channel.

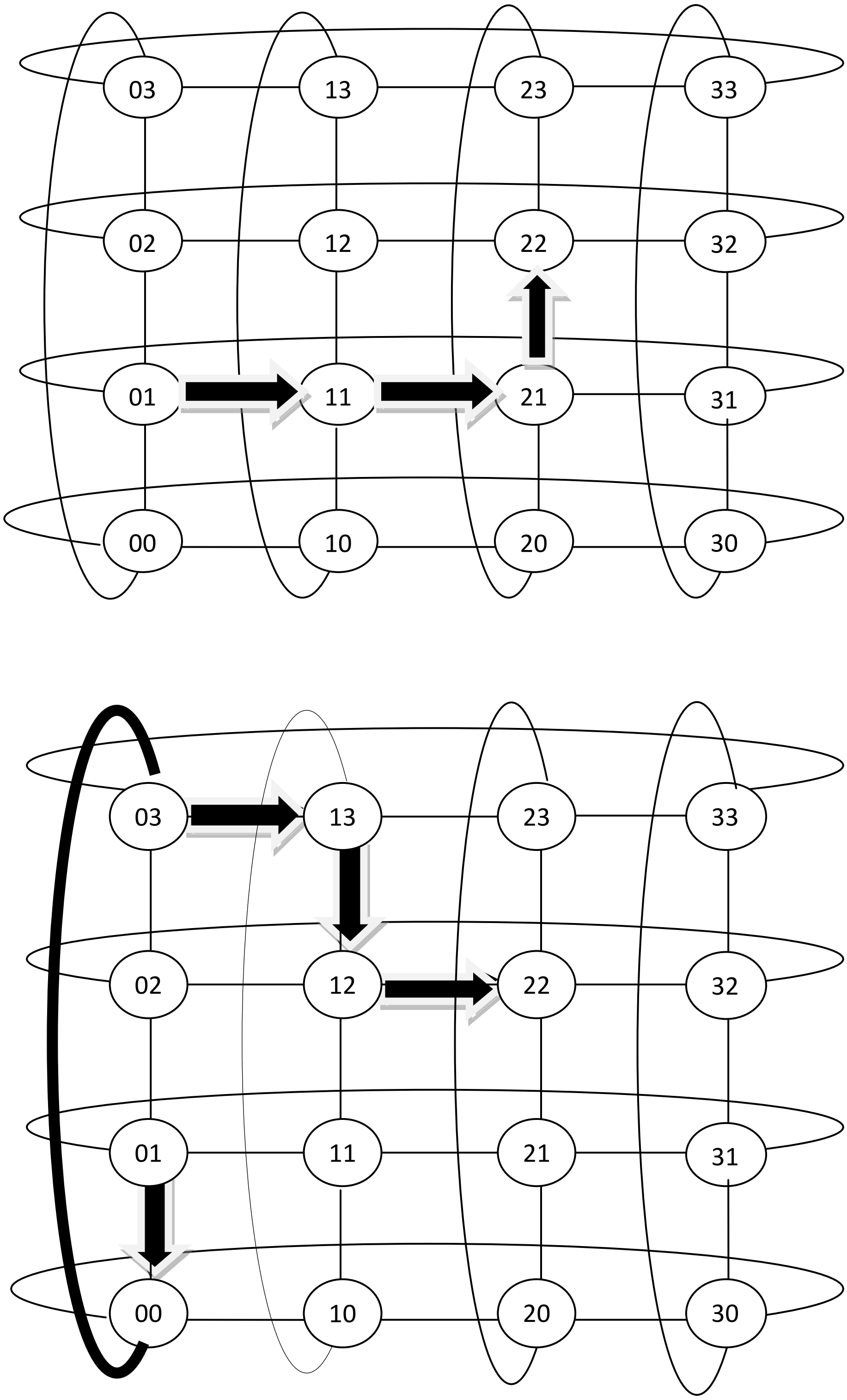


Figure 12 (a) Dimension-order routing moves the packet first in the x dimension, then it moves in the y dimension. (b) A non-minimal route requires more than the minimum path length. [1]

We notice that the path taken in Figure 12(a) has no of hops equal to 3, however in figure 12(b) there is no minimal routing depicting the no of hops equal to 5.

2.2.5 Evolutionary Algorithm

- Due to the increasing interest in the development of algorithms that are powerful enough to solve difficult optimization problems we have initiated to use the evolution of living beings present.
- The algorithms belonging to the class of algorithms which stochastic and are named as *evolutionary algorithm* (EA). The few efficient algorithms that fall under this category include *genetic algorithms* (GA), evolutionary strategies, evolutionary programming.

- Genetic Algorithms follows the underlying sequential steps:

- **Encoding –** The key issue in GA is how we encode the solution of the problem in hand and this is done into the chromosome **[15].**
- **Selection –** The genetic search is directed towards promising regions that are present in the search space **[15].**
- **Crossover-** helps to create new chromosome by combining the parts of the two chromosomes in hand **[15].**
- **Mutation –** small changes are made in one single chromosome which leads to the derivation of a new chromosome which is also stored for further use **[15].**

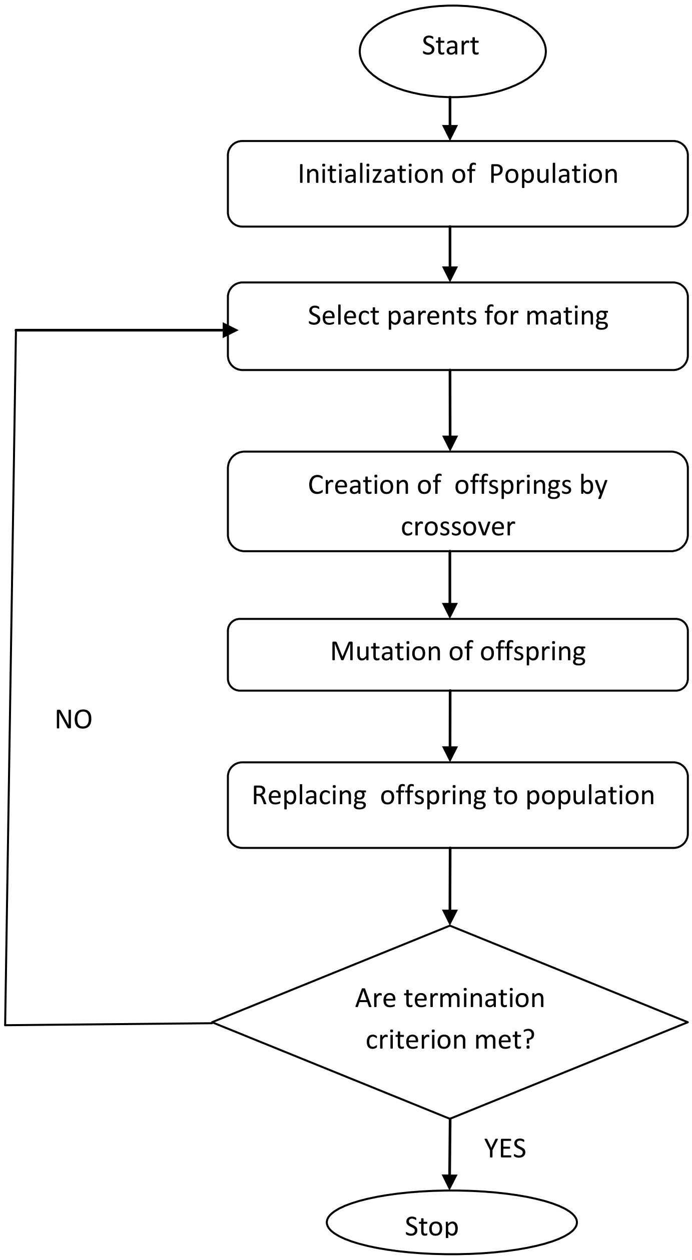


Figure 13 Flowchart of Standard Genetic Algorithm [15]

- There can be usage of other algorithms also like Simulated Annealing for solving difficult combinatorial problems.
- In Table 5 we can see the study of the various algorithms, depicting their characteristics.

Table 5 Study of algorithms used to solve optimization problems

Methodology Used	Definition	Strategy Used	Advantages	Disadvantages
Genetic Algorithm[16]	The GA mechanism is considered to be parallel when performed on a set of solutions and the information is exchanged using the crossover operation.	Consideration of fitness function at random in the selection strategy, in each runs of the algorithm.	The multi-point crossover and mutation does help in providing an output as the optimal path and also alternate path if required.	1. There is a possibility of it trapping at local minima, or finding of an optimal solution can be time consuming. 2. GA does rapidly discover the search space, but there is difficulty in finding the exact minima.
Simulated Annealing[16]	It takes one solution at any particular instance of time.	It emphasizes upon regulation of the parameter of temperature which is used to evaluate the solution.	A generic metaheuristic for finding a good approximation in order to locate the global optimum of a given function (a global optimization problem) considered in a large search space.	1. Only one candidate solution is used and no conclusive overview of the search space. 2. SA is sequential in nature and therefore it is slow.

Particle Swarm Optimization[17]	A candidate solution is iteratively in a process of improvement which optimizes a problem. Also a one -way information sharing mechanism.	Consideration of a population of candidate solutions (dubbed particles), while moving these solutions over the particle's position and velocity, around in the search space, in accoradance to simple mathematic -al formulae.	1. No overlapping and mutation. 2.The solution decides a real number code which is adopted. 3.Count of the dimension equals to the constant which is present in the solution.	1. Suffers from the partial optimism, leading to less precision at the guideline of its speed and the direction. 2. Cannot work out the problems of scattering, optimization and non-coordinate system,
Differential Evolution	A candidate solution is iteratively being improved which optimizes a problem which is with regards to a measure quality	Population of candidate solutions is maintained and creates candidate solution by combining existing ones, then keeping the candidate solution whichever gives the best score in hand.	1.Does not require a differentiable optimization problem. 2. Can also be considered on optimization problems that fall under the category as being even continuous, are noisy, change over time.	1. An optimal solution is ever found is not guaranteed.
Ant Colony Optimization (ACO)	Use graphs to find a good path thus converging to solving computational problems through this probabilistic technique.	Based on the ant searching of a good link present between the colony and a source of food and this is done finding an optimal path through the graph.	Pheromone evaporation provides with this advantage of avoidance of convergence to a locally optimal solution thus not constraining the solution space.	Reduction in the attractive strength as the pheromone trail begins to evaporate.

2.2.6 Interconnection Network Using Evolutionary Algorithm

Solving an optimization problem, which here caters to designing an interconnection network using EA, requires specification of:

(1) A collection of all the present candidate solutions to the problem and
(2) A required measure for evaluating the performance of each of the candidate solution in accordance to the required objective.

The goal requires finding a solution or a set of solutions for obtaining the best performance and provide with an optimal solution with respect to the specified performance measure.

2.2.6.1 Integration of Genetic Algorithm for Designing Interconnection Networks

- The genetic algorithm does maintain a population of *individuals or chromosomes.*

- Each chromosome here represents the number of links or nodes present in the topology.

- Each chromosome is evaluated to give as output some measure of its *fitness.*

- Then some of the chromosomes undergo genetic operations – mutation and crossover to form new individuals as output.

After several generations, we obtain the convergence to the best chromosome or individual, which hopefully provides with an optimal or for that matter a suboptimal solution to the optimization problem.
The encoding of this solution and afterwards the decoding of the corresponding solution is considered for the particular topology. Conventional

GA parameters are defined such as *population size* (pop_{size}), *crossover* (p_c) and *mutation* (p_m) *probability* [6] which is known to influence the performance of the algorithm.

The encoding of solution for designing of the interconnection network is carried out by encoding the links between the vertices present in the network into the chromosome. The '1' denotes the presence of a link between the corresponding two vertices and '0'denotes the absence. Say 1001011…01 is of N^2 bit length so this is decoded into a N×N 2-D matrix.

CHAPTER 3

PROPOSED APPROACH:

A GENETIC ALGORITHM BASED OPTIMIZED INTERCONNECTION NETWORK

This chapter focuses on our proposed approach. As our goal is designing an interconnection network using constrained parameters we need to take well care of the parameters. In this we very well explain the permissible limits that have been set on the performance parameters and the properties defining them. The algorithms used for obtaining the same are well explained here. The major focus is on the proposed approach of designing an interconnection network based on the criteria of performance parameters for which the optimization is carried out using the Genetic Algorithm. This leads to applying GA on the interconnection network so as to design one tied to the constraints.

3.1 Introduction

On-chip networks have always played a very crucial role in the determinacy of the performance of computing systems present which include high-speed network routers, embedded devices etc. On moving further, the communication architecture that binds them plays a significant role in the performance of the chip taken as a whole. For this very purpose interconnection networks are used & showcased better than bus and dedicated wiring wherein they provide high speed transfer of data or packets. Interconnection networks help with the provision of the processor performance concurrency as well as high bandwidth demands are also well met.

The parameters on which the topology is based upon are of great importance. The designing of interconnection network is based upon parameters of cost, bandwidth and latency. Therefore we can well see that both cost and bandwidth affect the topology of the interconnection network.

On taking into consideration the properties that these parameters use we conclude that cost is dependent upon the degree of the nodes and diameter of the network. The diameter is in turn dependent on the number of links. The second parameter that is considered here is the bisection bandwidth. For the bisection bandwidth we just need to take into consideration the bisection width which is finding the minimum number of links that have to be removed in order to get the network to be divided into two equal parts. As both the bisection bandwidth and cost are dependent on number of links as a result they are dependent on each other too therefore maximizing bisection bandwidth and minimizing the cost would lead a trade off. Therefore for optimization of this criterion we require genetic algorithm that can be used to optimize the network based on certain performance criteria. For this purpose genetic algorithm is applied and this is how the constrained optimized network is obtained and used between any source-destination pair so as to get the shortest path between them that is to be traversed.

3.2 Proposed Approach

3.2.1Optimization Criteria

The topology of the interconnection network is hugely dependent on the parameters considered. As we can see from Equation 3.2 in Sec. 3.2.7 the cost is dependent on the product of the degree and diameter. The diameter is in turn dependent on the number of links. The second parameter that is considered is the bisection bandwidth which needs the consideration of the bisection width where the cutest is calculated from Sec. 3.2.3. When the minimum number of links is removed from the network then this divides it into two equal halves and the cutest is obtained. The bisection bandwidth is also therefore dependent on links. On consideration of both cost and bisection bandwidth we get the tradeoff as both the parameters are dependent on the number of links. Therefore the maximization of bisection bandwidth and minimization of cost requires optimization in order to fulfill the objective. This optimization criterion is achieved by using genetic algorithm which is the technique applied for getting constrained and optimized interconnection network.

3.2.2 Coding Parameters

The encoding for the solution is done in the form of a chromosome as described in Sec 2.2.6.1. The network is represented in the form of a graph. The graph is then encoded as a N×N 2-D binary matrix M as shown in Figure 16 where the entries are of '0' and '1'. Each of the rows and columns denote one of the N vertices denoted as V_0, V_1, $V_2 \ldots V_N$. '0' denotes absence of a link and '1' denotes presence of a link between the corresponding two vertices. Say we consider a 3×3 network then for this there is a 9×9 2-D binary matrix and N^2 bits length chromosome. The chromosome is read from the 2-D binary matrix in a row wise linear form. Taking a permutation of a chromosome as [0101010100……..00101] it is of 81 bits. The population size(p_{size}) is =10 so the total population size is given by the formulae:

$$\boldsymbol{Total\ population\ size(in\ bits)} = \boldsymbol{N^2} \times \boldsymbol{p_{size}} \qquad \textbf{(3.1)}$$

This Equation 1 is obtained for each of the generations taken where the no of generations (no_of_gen) =10.

3.2.3 Graph Partitioning Algorithm: Kernighan–Lin Algorithm

The graph partitioning algorithm requires division of the network into two equal halves. For this very purpose the K–L algorithm helps us search for a group of node pairs which are to be swapped. This leads to increase in the gain even though swapping of the individual node pairs from that group might provide us with the decrease in the gain. It is iterative as well as group migration is also considered. Partitions must be equal sized resulting in only two partitions i.e. this doesn't allow for partitions to be of unequal size and the count of partitions not being more than two.

The goal of this algorithm contains partitioning into two equal halves as well as swapping of the nodes with the aim of minimizing the number of seen edges that connect the two partitions [41]. This helps in dividing the network as V_0, V_1, $V_2 \ldots V_{N/2}$ in partition 0 and $V_{N/2 \ldots} V_N$ in partition 1. This coincides with our objective of dividing the network into two partitions and then minimizing the external links. It can well be inferred from the analysis in Table 2 that to help fulfill our goal we can use the *Kernighan–Lin Algorithm* as the graph partitioning algorithm. One pass of the Kernighan-Lin graph bisection algorithm is as follows [18]:

1. Values of g_a, g_b are calculated for each $a \in A$,$b \in B$.
2. $Q_A = 0$, $Q_B = 0$
3. for $i = 1$to $n - 1$ do

   Begin
4. We choose $a_i \in A - Q_A$ and $b_i \in B - Q_B$ such that we have g_{a_i,b_i} for maximal value computed over all choices of *a* and *b*.
5. Set $Q_A = Q_A \cup \{a_i\}, Q_B = Q_B \cup \{b_i\}$
6. for each $a \in A - Q_A$do

   $g_a = g_a + 2\delta(a, a_i) - 2\delta(a, b_i)$
7. for each $b \in B - Q_B$ do

   $g_b = g_b + 2\delta(b, a_i) - 2\delta(b, b_i)$

   End
8. We choose $k \in \{1 \dots \dots \dots n - 1\}$in order to maximize $\sum_{i=1}^{k} g_{a_i b_i}$
9. Subsets $\{a_1, \dots \dots, a_k\}$ and $\{b_1, \dots \dots, b_k\}$are interchanged to obtain a new bisection.

End

3.2.4 Selection and Fitness criterion

This forms a part of the GA engine. The selection is carried out to so as to discover the best solution to the problem. Each of the candidate solutions need to be evaluated to the fitness function. The fitness function comprises of evaluation on the basis of the performance metrics which here are cost and bisection bandwidth. The Roulette Wheel method has been considered for selection. In this the wheel is spun equal to the size of the population. The fitness formulae can be seen in Equation 3.3 in Sec. 3.2.7. Our objective is to maximize the fitness function pertaining to the value obtained for each of the chromosome inp_{size}.

In this we consider a 9×9 matrix and eventually the maximum node degree comes out to four and so we get the values of 36(9×4).

The values in Equation 3.3 can also be obtained from Equation 3.2 and step no 2.2 of the algorithm in Sec. 3.2.7. The values are then substituted and fitness values calculated for each of the N^2 bit length chromosome in the population size. This gets a total of p_{size} values. Maximization of these values over the population and eventually the generation leads to the fittest individual being selected as the solution. As the last generation gives the best chromosome of N^2 bit length which can be decoded as the symmetric 2-D binary matrix K as shown in Figure 17 and this can eventually be decoded as the graph which helps in forming the network.

3.2.5 Crossover and Mutation Operators

These are the two important GA operators that are used in the process. On exploration of the search space new candidates are formed. For the purpose of crossover the pair of any two chromosomes is taken and then they are evaluated by taking a single point and then cutting both the chromosomes from that point. Single point is taken as we are using one-point crossover here as shown in Figure 14. This can well be seen from the Equation 3.4. The tail part of first chromosome is copied into second and the second chromosome copied into the first chromosome. The chromosome still does remain of the same N^2 bit length. The new child chromosomes are stored so that they can be used as in further. This is carried out for each of the pair of the chromosomes in each of the generations for the population size. The value of the probability of the p_c is kept 1 here.

The mutation operator is considered to be the secondary operator of the GA algorithm. The mutation probability value also needs to be kept close to 1. The application of the operator on the matrix is done according to the need. This is one prerequisite that needs to be taken care of. One example for this can be seen in Sec 2.1.4 where the repair algorithm is considered to be the mutation operator. So now here based on the application in hand and the output so obtained from the crossover operator for the 2-D N×N matrix we consider the symmetric property to be taken into consideration. The mutation operator required to make the matrix symmetric. The values of chromosome change accordingly when read from symmetric 2-

D matrix. As seen in Equation 3.5 the symmetric matrix is obtained by copying the values of a row to the corresponding column and vice versa.

This is carried out for each of the chromosome where the 2-D matrix is made symmetric for each of the candidate solutions. This is carried on till the last generation and for whole of the population size. The mutation is the last step in the GA algorithm being used. This thus leads to the formation of the symmetric 2-D matrix and so the network obtained is constrained to the parameters so as to meet the objective of maximizing the bisection bandwidth and minimizing the cost. This network is then simulated in the further step so as to look into the performance analysis. This is very well depicted in Figure 17 which includes the symmetric matrix as well as the network required for simulation. This simulation is required so for the construction of the network that has so been obtained after this experimental setup [38].

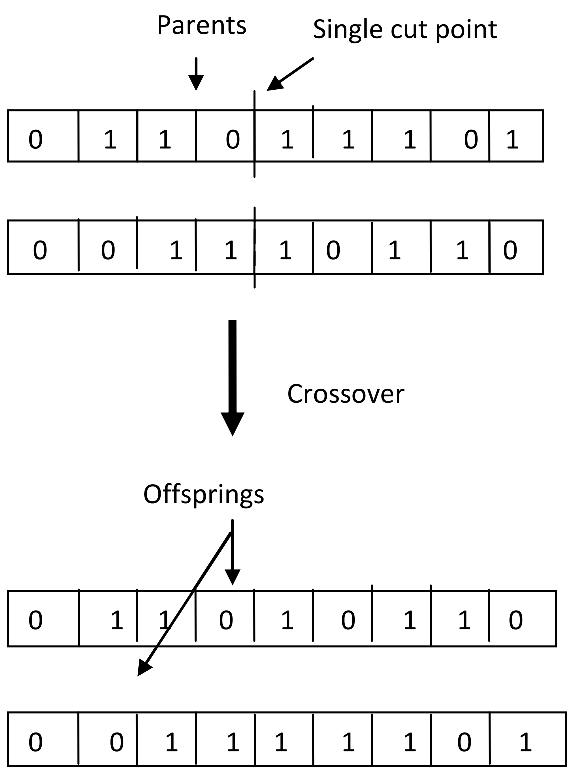


Figure 14 One - point Crossover operator

3.2.6 System Design

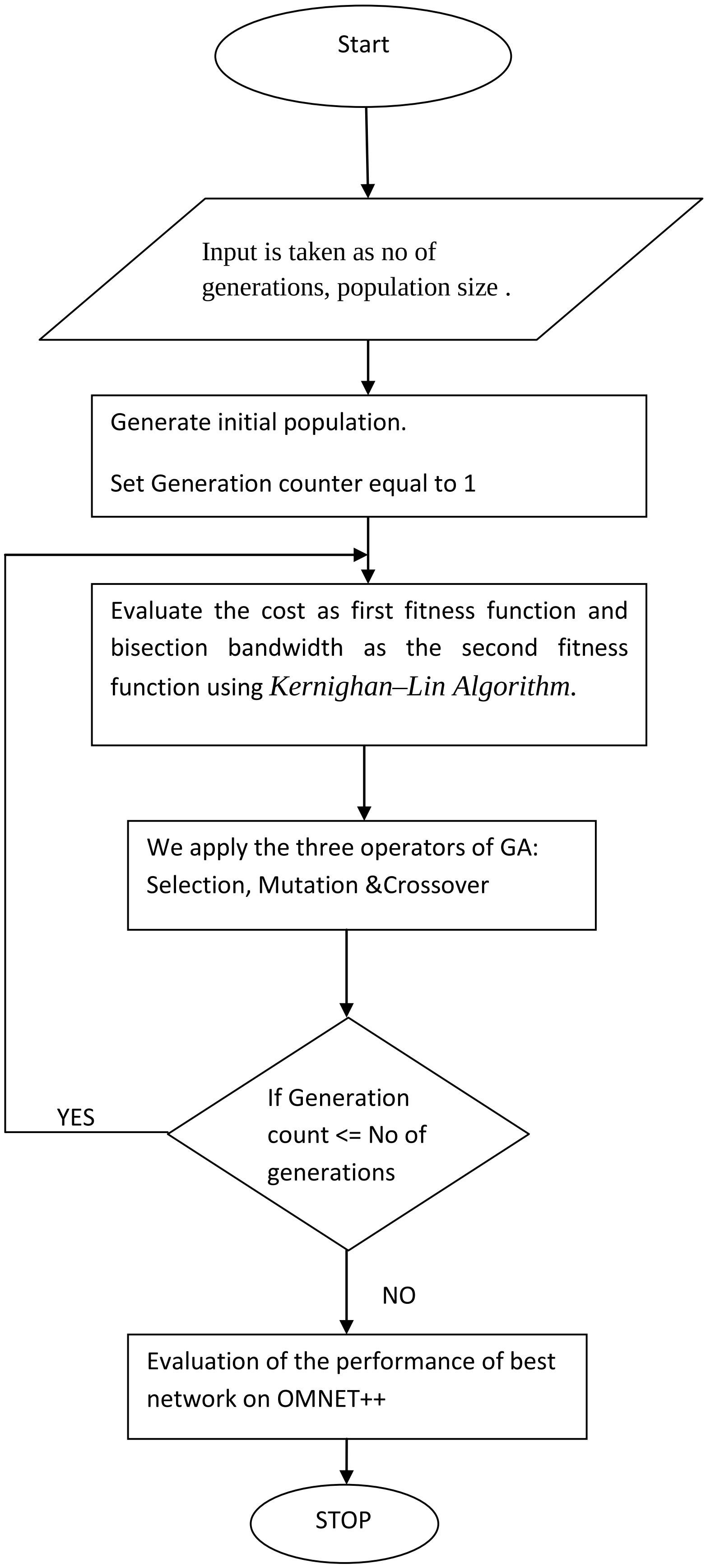


Figure 15 Flowchart depicting system design

3.2.7 Network Design Using Genetic Algorithm

Input: (pop_{size}, p_c, p_m, no_of_gen)

1. Generate initial population

2. While (stopping criterion not obtained)

2.1 Calculate the cost C (first fitness parameter):

$$\boldsymbol{C = Deg \times Diam} \qquad \textbf{(3.2)}$$

2.2 Calculate the bisection width (second fitness parameter) using Kernighan-Lin algorithm.

2.3 Form an Objective function *f(X)* and evaluate fitness of each chromosome upon it.

$$\boldsymbol{f(X) = Bisection\ width + (36 - C)} \qquad \textbf{(3.3)}$$

2.4 Apply the selection procedure to select chromosomes into the new generation.

2.5 Apply crossover GA operator on two individuals:

$$\boldsymbol{Offspring = crossover(parent\ 1, parent\ 2)} \qquad \textbf{(3.4)}$$

2.2 Apply mutation operator

$$\boldsymbol{M_{ij} = M_{ji}} \qquad \textbf{(3.5)}$$

3. End while loop

4. Declare the chromosome with minimum cost and maximum bisection bandwidth.

---

pop_{size} – size of the population

p_c - probability of crossover

p_m – probability of mutation

no_of_gen – number of generations

Deg = $\max\{deg_i | \, i \in 0 \ldots . N-1\}$

where deg_i = Number of nodes incident on node i.

Diam = $\max\{diam_i | \, i \in 0 \ldots . N-1\}$

where $\text{diam}_\text{i} = \max\{dist_r | r \in 0 \ldots\ldots N-2\}$

and $dist_{r} =$ number of links on path from from i th node to r th node.

3.2.8 OMNeT++: The Simulation Tool [23]

The OMNeT++ simulation IDE which is being used here is based upon Eclipse platform. The extension of this platform is carried out by using new views, editors etc. OMNet++ helps addition in functionality through

- The creation and configuration of models(NED AND INI files)
- Batch executions performed along with the simulation results being analyzed.

Eclipse provides with:

- C++ editing
- Modeling UML
- Bug-tracker integration
- Database access

This is done using the various plug-ins which are commercial and open source. OMNeT++ comprises of the following components:

3.2.8.1 NED files

In the OMNet++ IDE , on double clicking the .ned file , it opens up in the NED editor. As it is dual mode so in the graphical mode of editor we can make the required changes to the network whereas in the textual mode we are permitted to work straight away on the NED source.

3.2.8.2 INI files

The configuration or the INI files are the text files present with the .ini extension. They are required for parameterization and configuration of simulation models for execution. INI file editor too is dual mode.

3.2.8.3 Message Files

Choosing *File ->New -> Message Definition* (msg) brings up a wizard where one can provide with the specification of the target directory along with the file name for the message definition. There can be an empty MSG file too.

3.2.8.4 C++ Development

C++ files are opened in the IDE using the C++ source editor. The C++ code is assisted by the OMNeT++ IDE which also helps in configuring the build along with starting of the build process and then launching of simulations. The OMNeT++ IDE even contains the editors, views etc also. The C++ source editor support in tooltips of documentation, highlighting syntax as well as automatic indentation, formatting of code and some others too.

Creation of an OMNeT++ project that supports C++ development can be configured as:

File -> New -> OMNeT++ project

3.2.8.4 Launching and Debugging

OMNeT++ IDE helps us in executing single simulations as well those that are in batches. When running batches we can very easily gain from multiple processors or processor cores. The Eclipse as well as the IDE already comes in handy for storing of the particulars respective to the program that will be launched which includes the list of arguments, variables etc. This thus basically includes launching the project properly with the following steps used:

Run As -> OMNeT++ Simulation

Debugging support is also obtained from the Eclipse CDT which helps in provision of a large number of functionality which is additional. This includes inspection of the tooltips, conditional breakpoints etc.

This setup is very useful in simulation of the routing so as to evaluate the performance of the constructed network.

CHAPTER 4

IMPLEMENTATION DETAILS

This chapter involves taking the implementation of the proposed approach into consideration. This takes into account evaluation of algorithm as well as that of topology.

4.1 Evaluation of Algorithm

The evaluation of the algorithm comes before the evaluation of the topology. For this purpose we take into hand the 2-D matrix that has been constructed. One example can be seen is the way this 2-D binary matrix is read from the torus network shown in Figure 10(a). The sequence is formed as the network is encoded as the graph. This graph is subsequently formed into the 2-D matrix. The matrix has the entries being filled as ‘0’ and ‘1’. Here ‘1’ indicates a connection present between the vertices in the corresponding row and column and ‘0’ symbolizes absence of the connection. This 2-D matrix gets read in the linear form into a chromosome.

	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
V_0	0	1	1	1	0	0	1	0	0
V_1	1	0	1	0	1	0	0	1	0
V_2	1	1	0	0	0	1	0	0	1
V_3	1	0	0	0	1	1	1	0	0
V_4	0	1	0	1	0	1	0	1	0
V_5	0	0	1	1	1	0	0	0	1
V_6	1	0	0	1	0	0	0	1	1
V_7	0	1	0	0	1	0	1	0	1
V_8	0	0	1	0	0	1	1	1	0

Figure 16 2-D binary matrix M [6]

In Figure 16 9×9 matrix is taken and read linearly as 81 bits and therefore taken in the form of chromosome. The chromosome for the Figure 16 can be decoded as 011100100……1110. The generation of one population requires each of the chromosomes in the population size (p_{size}) = 10 to combine and form the population. This accounts to 810 bits in this case. The initial population is generated as follows:

(Written to **new_bit81.txt** file)

0010000000000010101101110100010100101000011011100100011111000001
0001111010001011100100001111001000101101011000000011010001011 10
1011100101101010110110010000000001000110011011001110110000111 01
1110010011010011111010100111011110001010100001110111000000101 00
0010101000000011101100001000100000011001101001010010011110110 01
1101100100010110110011011000001111111010000001101000101000101 01
1000100100011000011100010100101000011000001010101100001000110 11
0110001000111101100111000110111010001011000000111010000000111 01
0101000010110101101111001010110101110101100001011100001001110 00
0001011100001101101000100100100100101000001110110000000011110 01
1111110011000011100011101011010011111000001000010011010001010 00
0101000010010010100100000010111001100101110111011001111001101 01
00000100010100101001000001000001001110000110001011 1100

After storing this in the text file we calculate the first fitness parameter for each of the chromosomes present in the population which is cost. This is done equal to number of times of the population size=10.The results shown further are for the first generation

(Calculated for each of the chromosome of 81 bits and written to **outputcost.txt**)

18

18

18

18

18

18

12

18

18

Again corresponding to each of the chromosomes present in the population the second fitness parameter is also calculated which is taken as the bisection width.

(Cut set is computed and written to **output.txt**)

15

15

20

12

16

15

18

16

14

15

After this the calculation is carried out for evaluation of objective function f(X) for each of the chromosome.

f(X) = bisection width+ (36-C) where $X \in 0\ldots\ldots p_{size}$ which can be seen in Equation 3.3.

(Written to **fitnessfunc.txt**)

33

33

38

30

34

33

42

34

32

27

After this the crossover operation is applied on any pair of chromosomes.

(Written to **outputcross.txt)**

0010000000000010101101110100010100101000011011100100011111100001

000110000111011100

0110011011001110110000111011110010011010011111010100111011111000

101011101000101110

0001011100001101101101011000000011010001011101011100101101010110

11001000000000100

0100001111001000101000100100100100101000001110110000000001111001

111111001100001110

0011101011010011111000001000010011010001010000101000010010010100

10000001011100000

0101000011000001010101100001000110110110001000111101100111000110

111010001011001100

0111010000000111010101000010110101101111001010110101110101100010

00110000111011100

0010000000000010101101110100010100101000011011100100011111100000

101110000100111000

0111010000000111010000000100001001101000101000010100001001001010

010000001011100000

0011101011010011111101000010110101101111001010110101110101100010

00110000111011100

0100001111001000101000100100100100101000001110110000000001111001

111111001100001110

0010110110011011000001111111010000001101000101000101011000100100
001100001110001010

0101000011000001010101100001000110110110000101000101011000100100
001100001110001010

0010110110011011000001111111010000001101001000111101100111000110
11101000101100110

0111010000000111010000000100001001101000101000010100001001001010
010000001011100000

0111010000000111010000000100001001101000101000010100001001001010
010000001011100000

0101000011000001010101100001000110110110000101000101011000100100
001100001110001010

0101000011000001010101100001000110110110000101000101011000100100
001100001110001010

0000101000010101000000011101100001000100000011001101001010010011
11011001110110010

0000101000010101000000011101100001000100000011001101001010010011
11011001110110010

After this the mutation operator is applied so as to make the 2-D binary matrix symmetric for each of the chromosomes. The value to be carried on to the 1st generation and it is stored as follows:

0010000000000010101001110100010100100011011010110100010000100010
11100000000011100

0011101010010011111100001001000011011000010000101100011111000100
10000101110101010

01110100010001110110010000110101011001010010111000011101011100100010100000110111 00

0111010001000111011000001001000011010100010001101100010111000100000000101010101010

0100001111001000100000100100100100100011001110000000000100010001111110001100010110

0101000011000001010001100001010110110011000100001001010100010100001101011101010 10

0101000011000001010001100001010110110011000100001001010100010100001101011101010 10

0101000011000001010001100001010110110011000100001001010100010100001101011101010 10

0101000011000001010001100001010110110011000100001001010100010100001101011101010 10

0000101000010101000100011100000001001100000001001000100111101001001000001000010110

This is carried on for the no of generations (no_of_gen) =10.The values for the cost, bisection width, crossover and mutation are saved in the text file for each of the generations.

For the 9th generation also we need to take into consideration the values that are stored after the crossover as they will be required for mutation operator. Once the mutation is applied the modified values will be passed on further on to the last generation.

(Written to **outputcross.txt)**
0010000000000010101001110100010100100011011010110100010000100010111001010100 01010

0111010001000111011000100101000100100111010001100100010100000100011000010000111 10

0111010001000111011001011001110011010100011101011100101111110011
110011100000100100

0001011100001101100000100101000100100111010001100100010100000010
001100101010001010

0101000011000001010000000100100001101000000010000100101011101010
000010101110101010

0100001111001000100001100000110110110011000100001001011000001010
110110101100101010

0101000011000001010000000100100001101000000010000100101011101001
011100101010001010

0010000000000010101001110100010100100011011010110100010000010010
000010101110101010

0101000011000001010000000100100001101000000010000100101011101010
000010101110101010

0101000011000001010000000100100001101000000010000100101011101001
011100101010001010

0001011100001101100000100101000100100111010001100100010100000010
001100101010001010

0111010001000111011000100101000100100111010001100100010100000010
001100101010001010

0100001111001000100001100000110110110011010001100100010100000010
001100101010001010

0111010001000111011000100101000100100111000100001001011000001010
110110101100101010

0101000011000001010000000100100001101000000010000100101011101010
000010101110101010

0101000011000001010000000100100001101000000001000010010101110101000001010111010101O

0100001111001000100001100000110110110011010001100100010100000100011001010100 01010

0100001111001000100001100000110110110011010001100100010100000100011001010100 01010

01110100010001110110001001010001001001110100011001000101000000100110000101000 1110

01110100010001110110001001010001001001110100011001000101000000100110000101000 1110

The values of the 9th generation which is carried forward for computation in the last generation once the mutation has also been applied are as follows:

0010000000000010101001110100010100100011011010110100010000100010111000010000 11110

0111010001000111011001011001010011010100011101111100100111100110000111000101 00100

0101000011000001010000000100100001101000000001000010010101110101000001010111 01010

0101000011000001010000000100100001101000000001000010010101110100100000100011 10101110

0101000011000001010000000100100001101000000001000010010101110101000001010111 01010

0001011100001101100000100101100100100111010001000100011100000101111001010000 01010

0100001111001000100001100000110110110011010000001100011000000101101001011001 01010

0101000011000001010000000100100001101000000010000100101011101010000010101110101010

010000111100100010000110000011011011001101000000110001100000010110100101100101010

010000111100100010000110000011011011001101000000110001100000010110100101100101010

These values are considered to be the starting point for the last generation computations.

For the last generation also the values are calculated for bisection width, cost and evaluation of objective function f(X) for each of the chromosomes present as follows:

(Calculated for each of the chromosome of 81 bits and written to **outputcost.txt**)

18

12

18

24

18

12

18

18

18

18

Again corresponding to each of the chromosomes present in the population the second fitness parameter is also calculated which is taken as the bisection width.

(Cut set is computed and written to **output.txt**)

10

16

10

10

10

10

8

10

8

10

(Stored for the last generation in **fitnessfunc.txt)**

40

34

28

28

28

26

28

26

28

40

As we need to maximize the value of the fitness function. Therefore out of all the values obtained we choose 40 as it is the maximum of all the values obtained. This value has been chosen as it is the maximum and the fittest of all. Even the crossover values for the last generation are also stored which areas follows:

(Written to **outputcross.txt)**

0010000000000010101001110100010100100011011010110100010000100010111101011101010 10

0010000000000010101001110100010100100011011010110100010000100010

011110101110101010

010100001100000101000000100100001101000000010000100101011101010

000000001000011110

010100001100000101000101100101001101010001110111110010011110011

000011100010100100

011101000100011101100000100100001101000000010000100101011101010

000010101110101010

010000111100100010000110000011011011001101000000110001100000010

110100101100001010

00010111000011011000000100101100100100111010001000100011100000101111100101000101010

01000011110010001000011000001101101100110100000011000110000000101111010111010101010

00100000000000101010011101000101001000110110101101000100001001011010010110010101010

01000011110010001000011000001101101100110100000011000110000001011010010110000101010

01000011110010001000011000001101101100110100000011000110000000101111010111010101010

01110100010001110110000001001000011010000000100001001010111010100000101011101010101

01010000110000001010000000100100001101000000001000010010101110101000001010111010101010

00010111000011011000000100101100100100111000100001001010111010100000101011101010101

01010000110000001010000000100100001101000000100010001000111000001011111001010001010101

01000011110010001000011000001101101100110100000011000110000001011010010110000101010

01000011110010001000011000001101101100110100000011000110000001011010010110000101010

00010111000011011000000100101100100100111000100001001010111010100000101011101010101

00010111000011011000000100101100100100111000100001001010111010100000101011101010101

0101000011000000101000000010010000110100000000100001001010111010010000100011101011110

0101000011000000101000000010010000110100000000100001001010111010010000100011101011110

Moving further corresponding to the fittest value we have to extract the 81 bit chromosome.

Then the corresponding network is designed from the 81 bits. This is done in reverse order that means first of all the corresponding 81 bits are chosen. Then the corresponding 2-D matrix is formed and the graph is extracted. From the graph we form the corresponding network. So now here for the value of 40 we have the following 2-D matrix that has been extracted:

	V_0	V_1	V_2	V_3	V_4	V_5	V_6	V_7	V_8
V_0	0	1	1	1	0	1	0	0	0
V_1	1	0	0	0	1	1	1	0	1
V_2	1	0	0	1	0	1	1	0	0
V_3	1	0	1	0	0	1	1	0	1
V_4	0	1	0	0	0	1	1	1	0
V_5	1	1	1	1	1	0	0	1	0
V_6	0	1	1	1	1	0	0	1	1
V_7	0	0	0	0	1	1	1	0	0
V_8	0	1	0	1	0	0	1	0	0

Figure 17 2-D matrix K obtained for the topology

From this 2-D matrix the network is formed. This created network is then designed in OMNeT++ [23] in the form of the required topology. The evaluation of this topology is then carried out in order to determine the enhancement in the efficiency in this proposed design.

4.2 Simulation Results: Evaluation of topology

Once the network is formed then its evaluation is carried out on OMNeT++. The network is decoded as the topology. This topology comprises of 9 nodes. This is then simulated on OMNeT++ and the performance evaluation is then carried out. The simulation of routing is done so as to determine the performance and for this the following pre requisites have already been considered:

This depends upon:

- Simulation time limit
- Inter arrival time
- Packet Length
- Propagation Delay
- Data rate

The analysis of the topology is carried out taking into consideration the propagation delay to be 0.1ms. Apart from this the data rate is taken 1Gbps. For the simulation time limit the value is taken to be 0.5s. The packet length is 1024 bytes and the following figure 18 gives all these values present in the .ini file:

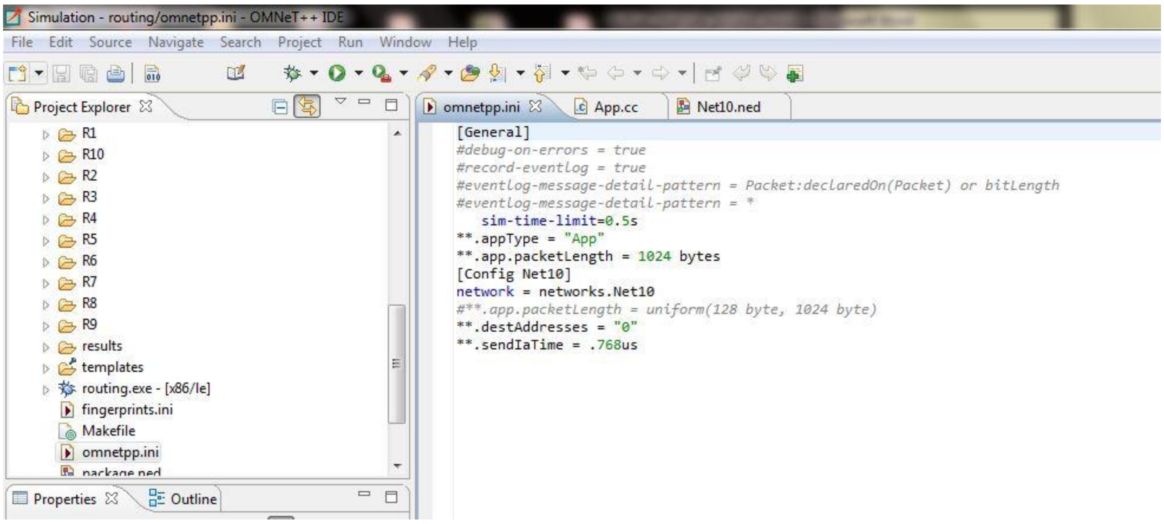


Figure 18 omnetpp.ini file

The values here are to be computed for packet inter arrival time equal to .768us. Figure 19 shows the network which is created and that has been coded in the .ned file. This is the creation of the topology that has been developed from the algorithm. This can very well be seen in Figure 19.

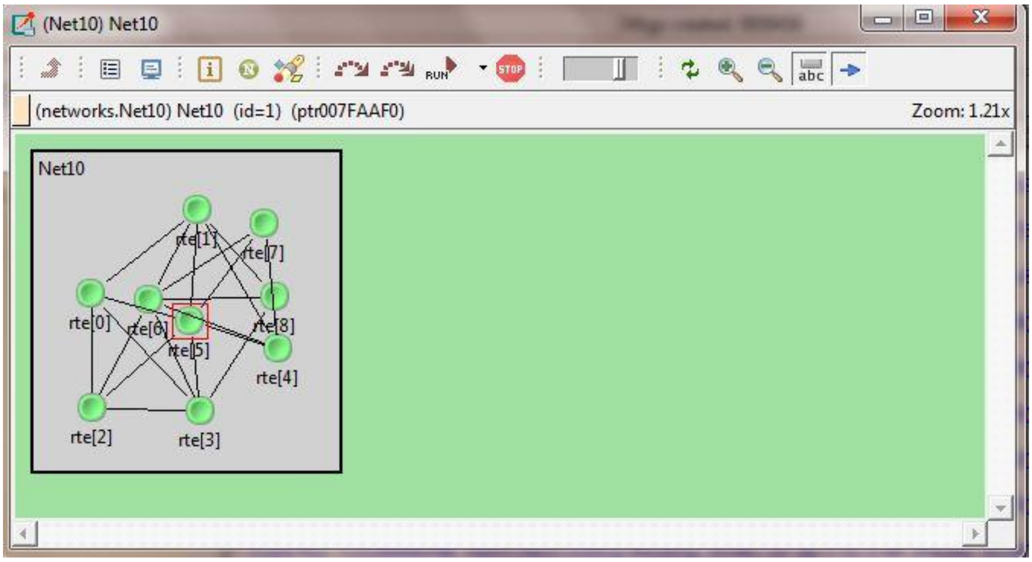


Figure 19 Created network topology

During simulation for Simulation time limit = 0.5s the values for events, messages generated are as seen in Figure 20.

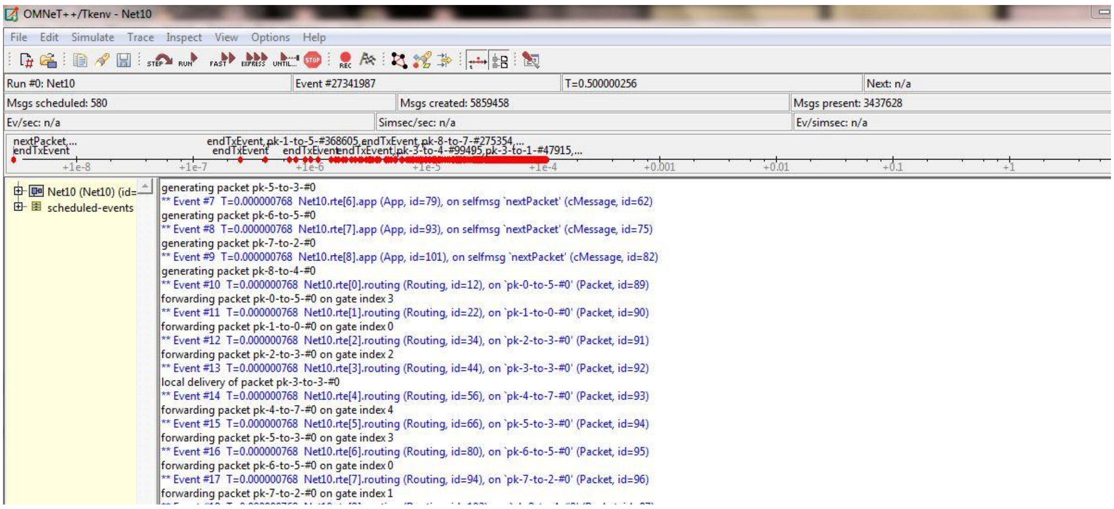


Figure 20 Simulation window

Figure 21 takes into account the mean latency values for .768us. These values are calculated for each of the 9 nodes and then the mean is taken.

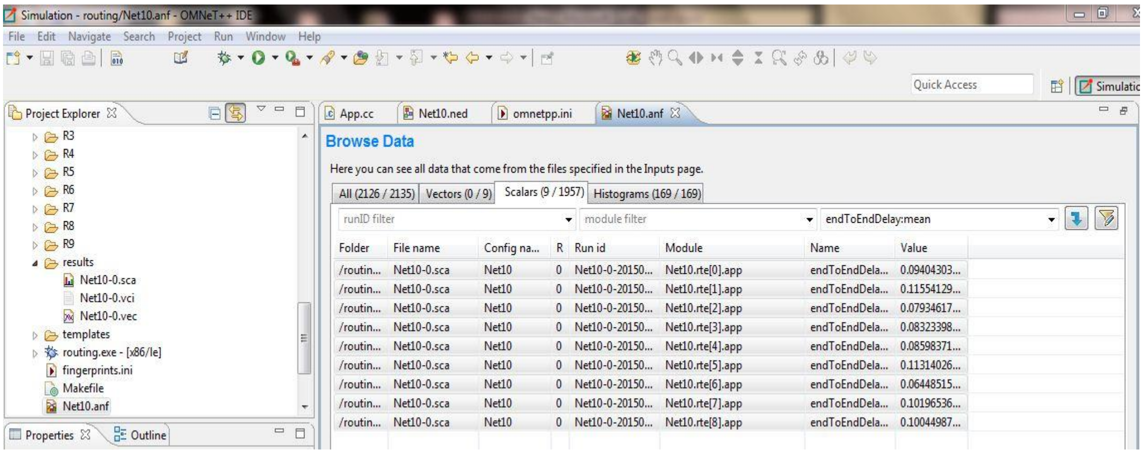


Figure 21 Mean latency values

Table 6 gives the latency values for the following defined packet inter arrival time:

Table 6 Inter arrival Time and Average Latency

Inter Arrival Time(us)	**Average Latency (Designed Network)**	**Average Latency(Torus Network)**
7.68	0.000141	0.000146
3.84	0.008562	0.013204
2.56	0.026911	0.037278
1.92	0.041558	0.065505
1.536	0.058685	0.079252
1.28	0.070029	0.0871
1.097	0.077601	0.090673
0.96	0.082576	0.091959
0.85	0.088137	0.09461
0.768	0.093132	0.097453

The Table 6 comprises of the comparison of the designed network and the Torus network in Figure 10(a). Through the comparative analysis we draw the infer that for every value of packet inter arrival time that has been taken in the Table 6, the average latency for the designed network is any time better than the existing network that is taken which is the Torus network. So in that case there is enhancement in the time to traversal of the designed interconnection network as shown by the graph in Figure 22. Each of the load values of the inter arrival time depict the fact that the designed interconnection network gives better results in terms of average latency.

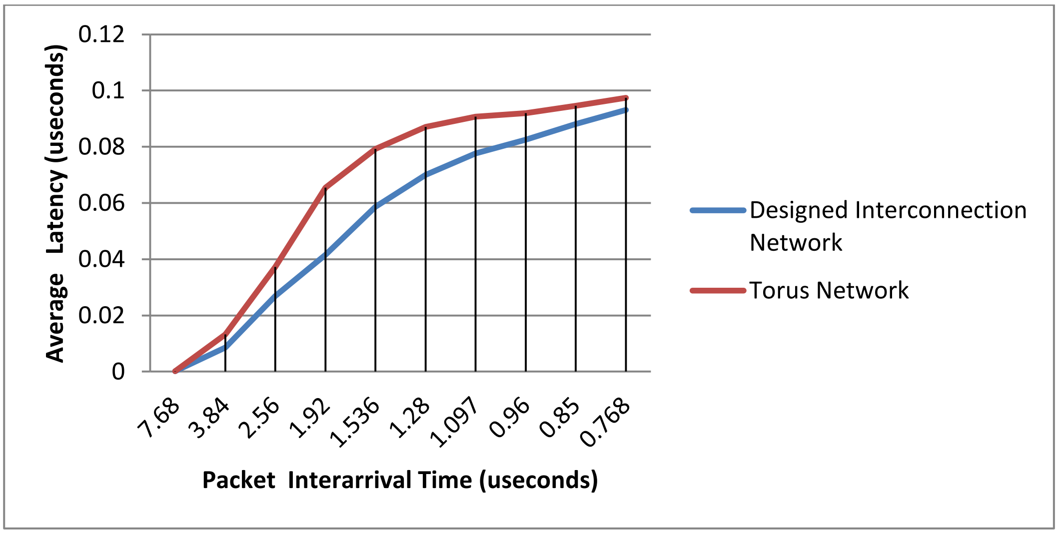


Figure 22 Average Latency versus packet inter arrival time

The average latency is computed on whole of the network once the end to end delay mean is found for each of the nodes. There is another set of analysis that can be drawn from Figure 22. This tells that with the decrease in the value of load on the packet inter arrival time the value of average latency increases. Therefore from this we infer that there is an increase in the average time taken for the evaluation of the network as the time of generation of packets decreases eventually.

CHAPTER 5
CONCLUSION AND FUTURE WORK

This chapter concludes by giving a brief overview of how the goal has been achieved and what steps have been taken for the same.

5.1 Conclusion

The design of an interconnection network is laid down constrained and pertaining to desired performance parameters. In this thesis the constraints in the parameters are determined by maximizing the bisection bandwidth and minimizing the cost [37]. There is a trade off between the two as cost is dependent on degree and degree is dependent on number of links whereas bisection width also depends upon number of links. This problem of obtaining an optimization has been solved using evolutionary algorithm. Genetic Algorithm (GA) has been used specifically. Then we work upon the various steps of GA. There is pre-determined population size and no of generations for which the each of the GA operators performs their functionality once the evaluation of the fitness function has been carried out. After this the fittest value is considered obtained over the number of runs equal to the no of generations. This is obtained after maximizing the value of fitness. This obtained value is decoded into the 2-D matrix and then as the network which finally leads to the formation of the required topology. This topology is then evaluated in OMNeT++ based on some parameters. The simulation of routing is done so as to provide with a decrease in the average latency for any packet inter arrival time. Therefore the designed interconnection network gives better results than the Torus network with which the comparison has been carried out.

5.2 Future Work

The work carried out in this thesis can be carried on further too. Here we have designed an interconnection network that is constrained to the parameters. It is optimized using the Genetic Algorithm. This procedure can be generalized further in future taking into consideration larger number of nodes where the value of N in N×N keeps on increasing as there is an increase in the total number of nodes N present in the network. Eventually the performance of the designed network can be tested as the number of nodes increase.

Apart from this we can also take into consideration varying the parameters or the properties affecting them. The properties may also be handled differently. As we know based upon them the interconnection network is constrained, designed therefore making changes will lead to evaluation of network accordingly.

References

[1] J. Siegel, "*Interconnection networks for large-scale parallel processing; Theory and case studies*," Lexington Books ,1985.

[2] Jose Duato," *Interconnection networks*," Morgan Kaufmann,1997.

[3] J. Jaros, "Evolutionary optimization of multistage interconnection network performance," in *Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, 2009, pp. 1537–1544.

[4] J. Jaros, M. Ohlidal, and V. Dvorak, "An evolutionary approach to collective communication scheduling," in *Proceedings of the 9th annual conference on Genetic and evolutionary computation*, 2007, pp. 2037–2044.

[5] P. K. Tripathy, R. K. Dash, and C. R. Tripathy, "A New Genetic Algorithm based Method for Topological Optimization of Interconnection Networks," Int. J. Comput. Appl.,vol. 63, no. 3, pp. 7–13, 2013.

[6] N. Krommenacker, T. Divoux, and E. Rondeau, "Using genetic algorithms to design switched Ethernet industrial networks," in Industrial Electronics, 2002. ISIE 2002. Proceedings of the 2002 IEEE International Symposium on, 2002, vol. 1, pp. 152–157

[7] J. Arjona Aroca and A. Fernández Anta, "Bisection (band) width of product networks with application to data centers," Parallel Distrib. Syst. IEEE Trans. On, vol. 25, no. 3, pp. 570–580, 2014.

[8] S. Sharma, K. S. Kahlon, P. K. Bansal, and K. Singh, "Improved Irregular Augmented Shuffle Multistage Interconnection Network," Int. J. Eng. IJE, vol. 2, no. 3, p. 27, 2008.

[9] I. Stojmenovic, “Honeycomb networks: Topological properties and communication algorithms,” Parallel Distrib. Syst. IEEE Trans. On, vol. 8, no. 10, pp. 1036–1042, 1997.

[10] S. Rajkumar and N. K. Goyal, “Design of 4-disjoint gamma interconnection network layouts and reliability analysis of gamma interconnection Networks,” *J. Supercomput.*, vol. 69, no. 1, pp. 468–491, 2014.

[11] I. Stojmenovic, “Direct Interconnection Networks,” *Parallel and Distributed Computing Handbook, A.Y. Zomaya, ed., pp. 537-567.* McGraw-Hill, 1996

[12] F. Thomson Leighton, “Introduction to Parallel Algorithms and Architectures: Arrays,” *Trees Hypercubes Morgan Kaufmann Publ.*, 1992.

[13] M. Ohlídal, J. Jaroš, J. Schwarz, and V. Dvořák, “Evolutionary design of OAB and AAB communication schedules for interconnection networks,” in *Applications of Evolutionary Computing*, Springer, 2006, pp. 267–278.

[14] J. Duato, S. Yalamanchili, and L. M. Ni, "*Interconnection networks: An engineering approach*," Morgan Kaufmann, 2003.

[15] J. Jaroš, “Evolutionary Design of Collective Communications on Wormhole Networks,” Ph. D. thesis, Brno, CZ, 2010.

[16] Y. R. Elhaddad, “Combined Simulated Annealing and Genetic Algorithm to Solve Optimization Problems,” *World Acad. Sci. Eng. Technol.*, vol. 68, 2012.

[17] D. P. Rini, S. M. Shamsuddin, and S. S. Yuhaniz, “Particle swarm optimization: technique, system and challenges,” *Int. J. Comput. Appl.*, vol. 14, no. 1, pp. 19–26, 2011

[18] T. Bui, C. Heigham, C. Jones, and T. Leighton, “Improving the Performance of the Kernighan-Lin and Simulated Annealing Graph Bisection Algorithms,” in *Proceedings of the 26th ACM/IEEE Design Automation Conference*, New York, NY, USA, 1989, pp. 775–778.

[19] A. J. Soper, C. Walshaw, and M. Cross, “A combined evolutionary search and multilevel optimisation approach to graph-partitioning,” *J. Glob. Optim.*, vol. 29, no. 2, pp. 225–241, 2004.

[20] J. Kennedy, J. F. Kennedy, and R. C. Eberhart, "*Swarm Intelligence,*" Morgan Kaufmann, 2001.

[21] R. Poli, “Analysis of the publications on the applications of particle swarm optimisation,” *J. Artif. Evol. Appl.*, vol. 2008, p. 3, 2008.

[22] A. Rexhepi, A. Maxhuni, and A. Dika, “Analysis of the impact of parameters values on the Genetic Algorithm for TSP,” *Int. J. Comput. Sci. Issues IJCSI*, vol. 10, no. 1, 2013.

[23] http://www.omnetpp.org/component/jdownloads/download/32-release-older versions/2273-omnet-4-4-1-win32-source-ide-mingw-zip.

[24] http://iroi.seu.edu.cn/books/asics/Book2/CH15/CH15.7.htm

[25] William J. Dally, “Performance analysis of k-ary n-cube interconnection networks,” *IEEE Trans. on Computers*, vol. 39, no. 6, pp. 775-785, June 1990.

[26] Mahsa Moazez , Farshad Safaei , Majid Rezazadeh, “Design and Implementation of Multistage Interconnection Networks for SoC Networks,” *International Journal of Computer Science, Engineering and Information Technology*, vol.2, no.5, October 2012.

[27] Manjunath E , Dhana Selvi D , “Design and Implementation of an On-Chip Permutation Network for Multiprocessor System-On-Chip,” *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, vol. 3, no. 6, pp. 173 – 177, June 2014.

[28] Kawalpreet Kaur, Sandeep Sharma , “Improved Meta-Flattened Butterfly Multistage Interconnection Network,” *International Journal of Computer Applications*, vol. 74, no. 9, pp. 34–37,July2013.

[29] Ravi Rastogi, Nitin, Durg Singh Chauhan and Mahesh Chandra Govil, “Disjoint Paths Multi–stage Interconnection Networks Stability Problem,” *International Journal of Computer Science Issues*,vol. 8, no. 1, 2011.

[30] C. R. Tripathy, N. Adhikari,” On A New Multicomputer Interconnection Topology For Massively Parallel Systems,” *International Journal of Distributed and Parallel Systems*, vol.2, no.4, July 2011.

[31] Akers, Sheldon B., and Balakrishnan Krishnamurthy, "A group-theoretic model for symmetric interconnection networks," *IEEE Transactions on Computers*, vol.38, no.4, pp. 555–566, April 1989.

[32] George B. Adams, Howard Jay Siegel, “The Extra Stage Cube: A Fault-Tolerant Interconnection Network for Supersystems,” *IEEE Transactions On Computers*, vol. C-31, no. 5, May 1982.

[33] Mahdi Nikdast, Jiang Xu, Luan H. K. Duong, Xiaowen Wu, Zhehui Wang, Xuan Wang, Zhe Wang,” Fat-Tree-Based Optical Interconnection Networks Under Crosstalk Noise Constraint,” *IEEE Transactions On Very Large Scale Integration Systems*, vol. 23, no. 1, January 2015.

[34] Iain A. Stewart,"Interconnection Networks of Degree Three Obtained by Pruning Two-Dimensional Tori," *IEEE Transactions On Computers*, vol. 63, no. 10, October 2014.

[35] Johnnie Chan, Student Member, Gilbert Hendry, Student Member, Keren Bergman, Fellow, and Luca P. Carloni," Physical-Layer Modeling and System-Level Design of Chip-Scale Photonic Interconnection Networks," *IEEE Transactions On Computer-Aided Design Of Integrated Circuits And Systems*, vol. 30, no. 10, October 2011.

[36] Aleksandr Biberman, Hugo L. R. Lira, Kishore Padmaraju, Noam Ophir, Johnnie Chan, Michal Lipson, and Keren Bergman, "Broadband Silicon Photonic Electrooptic Switch for Photonic Interconnection Networks," *Ieee Photonics Technology Letters*, vol. 23, no. 8, April 15, 2011.

[37] S. N. Dorogovtsev, J.F.F. Mendes, "Evolution of Networks: From Biological Nets to the Internet and WWW," Oxford University Press, 2003.

[38] George Michelogiannakis, and William J. Dally, "Elastic Buffer Flow Control for On-Chip Networks," IEEE Transactions On Computers, vol. 62, No. 2, February 2013.

[39] Loi, I. ,Kakoee, M.R., Benini, L., "A fully-synthesizable single-cycle interconnection network for Shared-L1 processor clusters," *Design, Automation & Test in Europe Conference & Exhibition, 2011*, Grenoble, 14-18 March 2011,pp. 1-6.

[40] Albert-laszlo Barabasi, Jennifer Frangos, "*Linked: The New Science Of Networks Science Of Networks*,"Perseus Publishing, 2002.

[41] Dong Chen ,Eisley, N.A. , Heidelberger, P. ,Senger, R.M. ,Sugawara, Y. ,Kumar, S. ,Salapura, V. , Satterfield, D.L. , Steinmacher-Burow, B. ,Parker, J.J., "The IBM Blue Gene/Q interconnection network and message unit," *SC '11 Proceedings of 2011 International Conference for High Performance*

Computing, Networking, Storage and Analysis, Seatle, WA, 12-18 Nov. 2011, Article No. 26, pp 1-10.

[42] Zhen Zhang , Greiner A. , Taktak S., “A reconfigurable routing algorithm for a fault-tolerant 2D-Mesh Network-on-Chip,” *Design Automation Conference, 2008, 45th ACM/IEE,* Anaheim, CA, 8-13 June 2008, pp. 441-446.

[43] Assaf Shacham, Keren Bergman, and Luca P. Carloni, “Photonic Networks-on-Chip for Future Generations of Chip Multiprocessors,” *IEEE Transactions On Computers*, vol. 57, no. 9, September 2008.

[44] Luca Scardovi, Murat Arcak, and Eduardo D. Sontag, “Synchronization of Interconnected Systems With Applications to Biochemical Networks: An Input-Output Approach,” *IEEE Transactions On Automatic Control*, vol. 55, no. 6, June 2010.

[45] Sriram R. Vangal, Jason Howard, Gregory Ruhl, Saurabh Dighe, Howard Wilson, James Tschanz, David Finan, Arvind Singh, Tiju Jacob, Shailendra Jain, Vasantha Erraguntla, Clark Roberts, Yatin Hoskote, Nitin Borkar, and Shekhar Borkar, “An 80-Tile Sub-100-W TeraFLOPS Processor in 65-nm CMOS,” *IEEE Journal Of Solid-State Circuits*, vol. 43, no. 1, January 2008.

[46] Mohammad Al-Fares, Alexander Loukissas, Amin Vahdat,” A scalable, commodity data center network architecture,” *Proceedings of the ACM SIGCOMM 2008 conference on Data communication*, Seattle, Washington, USA, August 17–22, 2008, Pages 63-74.

APPENDIX A

PUBLICATIONS

[1] Aradhya Saini, Akash Punhani, Nitin, *"*A Genetic Algorithm based Optimized Interconnection Network,*"* communicated in Third International Symposium on Women in Computing and Informatics (WCI-2015), August 10-13, 2015, SCMS, Aluva, Kochi, Kerala, India.