ROUTING OPTIMIZATION STRATEGY

USING GENETIC ALGORITHM

By

MADIHA SARFRAZ

(2006-NUST-MS PhD-CSE (E)-07)

Submitted to the Department of Computer Engineering
in fulfillment of the requirements for the degree of

Master of Science
in
Computer Software Engineering

Thesis Advisor
Dr. Muhammad Younus Javed

MS-CSE-6

College of Electrical & Mechanical Engineering
National University of Sciences and Technology
2010

Acknowledgements

I thank Almighty Allah for the successful completion of my thesis. I gratefully acknowledge the encouragement and support of my Advisor Dr. Muhammad Younus Javed. He has made available his support in a number of ways and has been a great mentor always providing me with the much needed encouragement and thoughtful direction.

I am heartily thankful to Dr. Shaleeza Sohail for her guidance, time and encouragement from the initial to the mature level, enabled me to develop an understanding of the subject. I was very fortune to have been able to work with her. Her detailed and constructive comments were vital to the development of this thesis.

I owe my deepest gratitude to Dr. Almas Anjum for his help and precious advice. I am also thankful to my advisory committee for their feedback.

I am indebted to my friends Umber and Javaria for their support, confidence and for sharing their invaluable assistance.

I would also like to convey thanks to the National University of Science and Technology (NUST), College of E & ME and Faculty of Department of Computer Engineering for providing me the financial means and laboratory facilities for my research work.

My parents, my grandmother (late), brother and sister are mentioned last to emphasize the special nature of their tremendous support and patience all through my candidature. And also wishes to express love and gratitude for their understanding & endless love, through the duration of my studies and completion of the project.

.

Dedication

I dedicated this thesis in the honour of my parents, my grandmother (Late), my brother and sister who always supported me and prayed for my success

Abstract

Intelligent analysis and designing of network routing provides an edge in this increasingly fast era. In this work, a variation of Genetic Algorithm (GA) for finding the Optimized shortest path of the network is presented. The algorithm finds the optimal path by using an objective function consisting of the bandwidth and delay metrics of the network and also through bandwidth and utilization metrics. The main distinguishing element of this work is the use of "2-point over 1-point crossover". The population comprises of all chromosomes (feasible and infeasible). Moreover, it is of variable length, so that the algorithm can perform efficiently in all scenarios. Rank-based selection is used for cross-over operation. Therefore, the best chromosomes are crossed over and give the most suitable offsprings. If the resulting offsprings are least fitted, they are discarded. Mutation operation is used for maintaining the population diversity. Various experiments have been performed for the population selection. The experiments indicate that random selection method is the most optimum. Hence, the population is selected randomly once the generation is developed. The results prove that proposed algorithm finds the optimal shortest path more efficiently than existing algorithms.

Table of Contents

List of Figures

List of Tables

CHAPTER 1: INTRODUCTION

Optimal shortest path finding is gaining enormous importance nowadays. It is one of the essential aspects in the field of networks. Besides, it has variety of applications in other areas like transport, graph structure etc. Genetic algorithm is a part of Evolutionary Computing. It's a rapidly mounting field of Artificial Intelligence.

For working out multipurpose engineering and scientific problems [30, 36] is a very challenging task. Multiobjective Engineering problems are cost minimization, increase performance, maximization of reliability [36]. These problems are difficult but are pragmatic [36]. In such optimization problems, there is a conflict in objectives [30, 36], averting concurrent optimization of every defined objective [36] and thus there may be a requirement of complex computations [30]. The range of multiobjective optimization problem is from linear objective function to Pareto-based methods [30]. There isn't any specific algorithm or problem domain defined. Multiobjective Optimization is also called as "multiperformance optimization" or "multicriteria optimization" [31].

Problem Statement

Carry out literature review of existing and previous research techniques in connection with routing optimization. Obtain simulation results and present detailed analysis of the proposed approach. Design & implement an effective Multi-Objective Genetic Algorithm for network routing optimization:

- Handling the network path optimization using bandwidth, utilization and delay constraints
- Population Selection
- Crossing over Technique

1.1 The Importance of Network Path Optimization using Genetic Algorithm

Genetic algorithms is significantly using in various application areas. A vital solution for end-to-end Network Optimization is needed. Genetic algorithms combine the good information hidden in a solution with good information from another solution to produce new solutions with good information inherited from both parents, inevitably leading towards optimality. GA is tailored to handle multiobjective problems through the fitness functions and also it commences methods for encouraging diversity in solution

The optimized Shortest Path Problem (SPP) is based on the analogy of finding shortest path between the source node and destination node in a network. The two nodes are connected through a link that represents the distance between them. The more the distance, the greater will be the time that the packet takes to reach the next node and vice versa.

The data traffic is increasing in current networks. There should be balance in broadcast and quality of service of the network traffic. If network load increases, it will affect the quality of service as it will cause delay in the network.

In this research work, a novel approach of optimized routing is introduced which is based on Genetic Algorithm. It finds out the optimum and the most suitable path which is qualifies Objective Function criteria. Objective Function value is associated with every chromosome and that fitness strength is evaluated by a function defined by user known as Fitness Function [27] or Objective Function.

The Objective Function criteria is that network bandwidth should be available, the delay will be less, utilization of the link should be less and if there are paths which have same delay factor value, than the path having less number of hops will be selected. Hence, we are handling Traffic Engineering aspect in networks. In a network, if the packet is taking more time in transmission from source to destination, than more will the delay. If the network is loaded, than its utilization factor will be. The utilization factor is assessing as per M. Ericsson, M.G.C. Resende and P.M.Pardalos approach [5] based on Genetic Algorithm.

A genetic algorithm is a global technique of optimization [12] which is based on the Natural Selection [11] [1]. GA helps in simplifying many elements in network path optimization. Optimization is ideal for routing and high performance of the Data Network. The routing problem thus can be simplified by optimization [14]. In this research work certain constraints of bandwidth, delay, utilization and hop count are defined, which than evaluates the quality of the genetic string. We thus develop a routing approach which is based on Genetic Algorithm. The chromosome is representing the nodes of that path and path can be of variable length. It is not merely efficient and robust but it congregates promptly.

Real life scenarios are having multiobjective characteristic in nature. As they have multiple objectives and these objectives may have conflict with each other [35]. The objective are measured in a defined unit i.e. they are measurable. Genetic Algorithm is a Population based technique, so it’s most suited for solving optimization problems even those having multiobjective requirement [36]. The Genetic Algorithm’s ability of simultaneously searching various areas at a time thus makes possible finding a solution set which is diverse for various problems. Crossover property of Genetic Algorithm exploits good solution’s arrangement in respect with various objectives and creates non dominated new offsprings. Genetic Algorithm is a heuristic approach for designing of multiobjective and optimization problems [36]. Schaffer [37] proposed first Multiobjective Genetic Algorithm; famous as Vector Evaluated Genetic Algorithm. Many methods for using GAs for coping up with optimization of problem which is simultaneous over a lot of dimensions is being proposed, inclusive Pareto-ranking use [41]. The design components and issues of Multiobjective GA are Fitness Function, diversity, elitism and constraint handling which are described by Abdullah Konak, David W. Coit and Alice E. Smith [36].

In the majority of real world engineering scenarios simultaneously involve optimization through multiobjective where trade-offs consideration is vital. In preceding decade, genetic evolutionary techniques are primary tools for solving problems of multiobjective which are associated with real world.

The remaining part of the Thesis is structured as follows:

1.2 Thesis Outline

This chapter introduces the importance of optimal path finding and its essentials in the field of networks. The rest of the section provides an outline of remaining of the chapters in this thesis.

Chapter 2- Related Work

This chapter surveys the work related to network path optimization using genetic algorithm.

Chapter 3- Overview of Genetic Algorithm

This chapter presents an overview of Genetic Algorithm outline and its processes. It discusses its implementation details based on natural selection. It also elaborates the genetic algorithm operators.

Chapter 4- Multiobjective GA

This chapter presents an overview of multiobjective genetic algorithm. It discusses the basic design issues and components of GA involved.

Chapter 5- Multiobjective Optimization

This chapter presents an overview and general approaches to multiobjective optimization. Also presents its ultimate goals and the conflicting goals.

Chapter 6- Proposed Routing Optimization Strategy using Genetic Algorithm

This chapter is dedicated to the implementation of our proposed algorithm. In this the algorithms are defined and also implemented.

Chapter 7- Experimental Results and Analysis

This chapter presents the experiments and testbed, network structure and population selection scenarios. After that experimental procedure, results obtained and a detailed analysis of the results are presented.

Chapter 8- Future Research Direction

This chapter gives an idea of further enhancement of this research work.

Chapter 9- Conclusion

This chapter concludes the thesis.

CHAPTER 2: RELATED WORK

CHAPTER 3: OVERVIEW OF GENETIC ALGORITHM

Genetic Algorithm (GA) is a search paradigm that emulates an environment with ‘survival of the fittest’ scenario where populations of data can fight and only the fittest survive. The basic concept of Genetic Algorithm for evolution follows the principles foremost laid down by Charles Darwin. Genetic Algorithms (GAs) are heuristic search algorithm on the concept of natural selection. GA pioneer is John Holland [3] [20] [36]. Holland technique is most effective as he not only reflected on the role of mutation but he also makes use of genetic recombination [15]. The genetic recombination (also called as crossover) significantly improve the competence of the algorithm to achieve and actually find out the optimum result. Holland had a dual aim; first to improve the concept of understanding of natural selection process, Secondly to develop artificial structure having functionality analogous to natural system [16].

The GA process starts through a Population, which is ‘a set of possible solutions’. The population is evolved with the assumption that the new population will be better than the old one. The new solutions are selected on the basis of their fitness. The cycle of evolutions continues until some terminal condition is satisfied.

Optimization is the best way of selecting the appropriate solution from given options. In any optimization case there is an Objective Function. To get an “Optimum” does not necessarily mean “maximum”. It means the best value for the Objective. For getting optimization doesn’t actually means to have the Best result. It means to have the best value which is meeting the criteria most accurately. Genetic Algorithm is outstanding for all tasks requiring optimization and is highly useful in any state of affairs.

GA modeling is done after natural selection mechanism. Each parameter of optimization is encoded by gene by appropriate representation of bits [33] or discrete units [36]. The respective genes of all parameters constitute a chromosome which is capable of designing a solution [33] [36]. A chromosome set represents several individual solutions which compose a population [33]. GA works with collection of chromosomes called population. Population initialization is random [36]. The chromosomes which are well fitted in the population are selected for reproduction [33]. In nature, weak and unfit species within their environment are faced with extinction by natural selection. The strong ones have greater opportunity to pass their genes to future generations via reproduction. In the long run, species carrying the correct combination in their genes become dominant in their population. Sometimes, during the slow process of evolution, random changes may occur in genes. If these changes provide additional advantages in the challenge for survival, new species evolve from the old ones. Unsuccessful changes are eliminated by natural selection [36]. Mating is done through crossover by combining genes of different parents to produce offsprings. The offsprings are added into the population and the process starts again, thereby creating an artificial Darwinian Environment [33]. GA is inspiration is from evolutionist theory of origin of species. GA deals with encoding problems instead of problem itself.

The procedure of a generic GA is given as follows:

3.1 Outline of the Genetic Algorithm

I. **[Start]** Generate a population of n chromosomes randomly which are suitable solutions.

II. **[Fitness]** Establish a function to evaluate the fitness of each chromosome in the population. Fitness f(x) of each chromosome x is calculated

III. **[New population]** Create a new population by repeating following steps until the new population is complete. After each repetition we get another generation.

- **[Selection]** Select parents from a population as per their fitness value
- **[Crossover]** Cross over the parents is done and new offspring are formed.
- **[Mutation]** Offsprings are mutated as per their mutation rate
- **[Accepting]** New Offsprings are placed in the new population if they are better than already existing chromosomes.

IV. **[Termination**] If the termination condition is satisfied, stop, and return the best solution in current population

V. **[Loop]** Go to step # 2

The flow chart of GA is below:

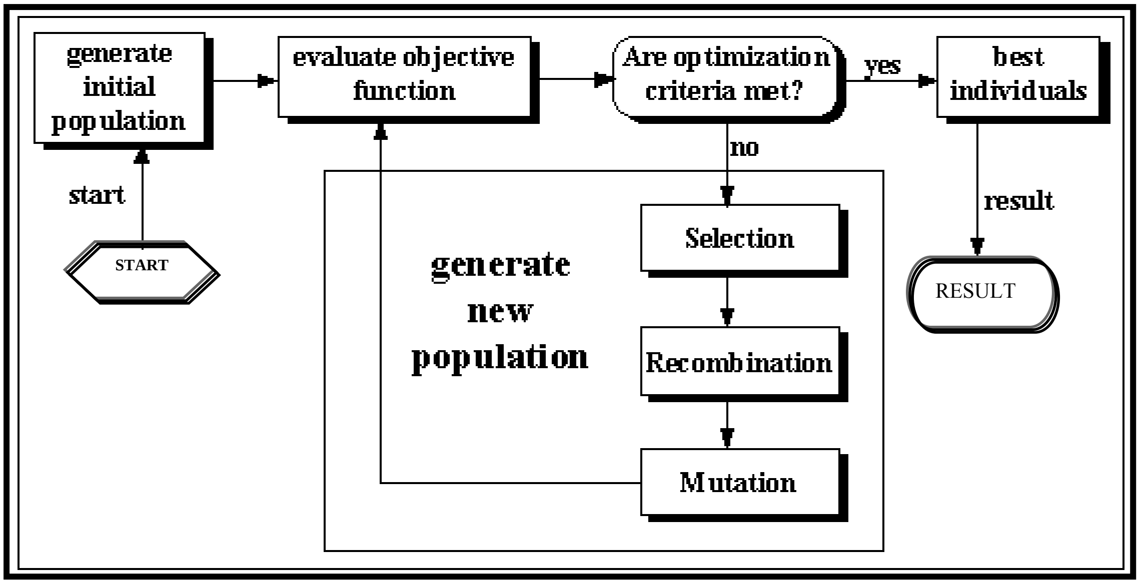

Figure 3.1. Standard Genetic Algorithm (SGA)

3.2 Implementation Details Based on Natural Selection

After the initial population generation, the GA efficiency is dependent on the use of its operators, which makes up GA course of action [11]. The Genetic Algorithm operations are [12] [1] [20] and their effect is described in section [20].

3.2.1 Selection & Reproduction

Selection is the process which equates to survival of the fittest; select the chromosomes from the population according to fitness value. This operates on the basis of Darwinian Theory of survival of the fittest among living organisms. This natural selection is an artificial version. The chromosomes are select from the population as per their Objective Function value. The Objective Function is the ultimate node for selecting chromosome survival. Chromosome with a higher fitness value has a higher probability of contributing their one or more offspring in next generation. There are several methods for selection of the best-fitted chromosome: Roulette Wheel Selection, Boltzman Selection, Tournament

Selection, Rank Selection, Steady State Selection and others. The details are in section 3.4.

3.2.2 Recombination or Crossover

Crossover represents mating between individuals. It is a primary distinguishing factor from other optimization techniques. The chromosomes are selected on the basis of selection operator. The crossing over of the selected parents is done on the basis of certain crossing over mechanism; 1-point crossover or 2-point crossover. A crossover of the parents thus forms new offspring. The details are in section 3.5.

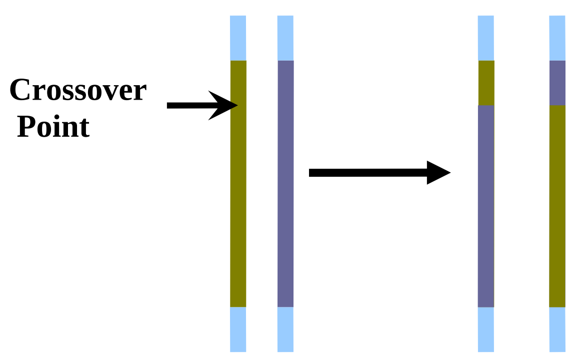

Figure 3.2 (a) Figure 3.2 (b)

Figure 3.2 (a)Crossing over of two parents (b)Offsprings generated after crossover

3.2.3 Mutation

Mutation introduces modifications. Its role is secondary in maneuver of GA as in natural population the mutation rate is also small [12]. After the crossing over of parents, the new offsprings are mutated at each locus of the chromosome with a probability (known as mutation probability). The details are in section 3.6.

3.3 Genetic Programming Operators

The Genetic Algorithm operators are explained as below:

Encoding Schemes

The chromosome should enclose information about the solution it represents.

3.3.1 Binary Encoding

Binary string is an encoding technique. The chromosome could look like:

Chromosome 1	1101100100110110
Chromosome 2	1101011000011110

Figure. 3.3. Binary Encoding

Bit in the string represent the characteristic of the solution or whether some particular characteristic is present or not.

3.3.2 Permutation Encoding

Ordering problems are solved through Permutation encoding. It includes the traveling salesman problem or a task ordering problem. Each chromosome is represented by a string of numbers, which corresponds to number in an order. For example, in the Traveling Salesman Problem each number would represent a place to be visited.

Chromosome 1	1 4 7 9 6 3 5 0 2 8
Chromosome 2	9 3 2 5 8 1 6 0 4 7

Figure. 3.4. Permutation Encoding

3.3.3 Value Encoding

When there are some complicated values (such as real numbers) and the Binary Encoding would not suffice than the Value Encoding can be used. As value encoding is very good for various problems, it is often required to develop some precise crossover and mutation methods for the chromosomes.

Chromosome 1	A B E D B C A E D D
Chromosome 2	N W W N E S S W N N

Figure. 3.5. Value Encoding

In chromosome 1 above, A could represent a particular task, B another, etc. For chromosome 2, N could be north, S south and thus could be the path through a maze.

3.3.4 Tree Encoding

Tree Encoding is used to actually have programs or expressions evolve. In tree encoding each chromosome is a tree of some objects, such as methods or commands in the programming. LISP is often used for this because it can be easily parsed into Tree.

Example of a Problem: Finding a function from given input and output values.

Task: Find a function that will give the best output for all given inputs.

3.4 Reproduction & Methods for Selecting the Best Chromosomes

There are different selection procedures in GA depending on how the fitness values are used. Proportional selection, ranking, and tournament selection are the most popular selection procedures [36]. The various other methods for choosing the best chromosomes: Boltzman selection, steady state selection and others.

3.4.1 Roulette Wheel Selection or Proportional selection

It’s a simple reproduction method in which offspring strings are allocated using a roulette wheel. The slots are sized as per their fitness value. This is a way of choosing individuals from the population, in a way that is proportional to their fitness. Parent’s selection is according to their fitness. The better the fitness value of the chromosome enhances the chance of its selection; however it is not assured that the fittest individual goes to the next generation.

A roulette wheel is showing all chromosomes in the population according to fitness as shown in the figure below. The wheel is "spun" and the marble falls into one of the slots. The wheel is spun to “select each chromosome that is to mate”.

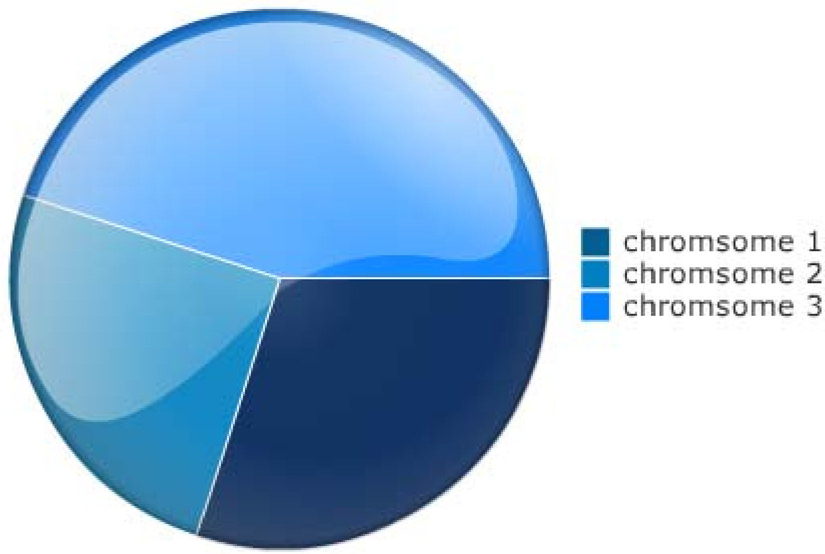

Figure. 3.6. The wheel is "spun" once for each chromosome that is chosen.

3.4.2 Steady State

In a Steady-State Genetic Algorithm one individual of the population is changed at a time. The selection of an individual from the population is according to its fitness. It is copied and the copy mutated. A second individual of the population is selected which is replaced by the mutated individual. In crossover two individual of the population are chosen, a single child created which replaces an individual of the population. Any selection method can be opted for selecting individual for mutation or the parents. The number of replacement strategies are there, which are:

- Replace Worst
- Replace a Randomly Chosen Individual
- Select replacement using the Negative Fitness

The major difference between Steady-State Genetic Algorithm and Generational Genetic Algorithm is that for each P individual of population generated in the Generational GA there are 2P selections. Therefore the selection strength and genetic drift for a Steady-State Genetic Algorithm is twofold to that of Generational Genetic Algorithm. The Steady-State Genetic Algorithm, consequently, appears twofold as fast although it can lose out in the long lasting terms because it does not explore the landscape as well as the generational GA.

3.4.3 Tournament

In Tournament Selection 'n' individuals are selected randomly and the fittest individual is selected. The most familiar type of tournament selection is binary tournament selection, where just two individuals are selected.

3.4.4 Elitism

The best chromosome is copied to the population in the next generation. The rest are chosen in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best found solution in the population. A variation is to eliminate an equal number of the worst solutions, i.e. for each "best chromosome" carried over a "worst chromosome" is deleted.

3.4.5 Rank Selection

The roulette method of selection will have problems when the fitness differs greatly. For Example, if the best chromosome fitness is 90% of the entire roulette wheel then the other chromosomes will have a slim chance of being selected.

Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst will have fitness 1, second worst 2 etc. and the best will have fitness N (number of chromosomes in population).

GA uses two operators to generate new solutions from existing ones: crossover and mutation [36].

3.5 Crossover

It's the most significant operator in GA. In this two chromosomes (called parents) are combined and forms new chromosomes (called offspring). The parent selection is from existing chromosomes in population having greatest fitness power. By applying all this iteratively the good gene chromosome appears frequently in population, thus leading to overall better solution [36].

3.5.1 Single Point Crossover

In this the two chromosomes will swap their bits. Crossover is performed by selecting a random gene along the length of the chromosomes and swapping all the genes after that point.

Randomly choose a crossover point, then for offspring 1

a) Copy everything in parent 1 before the crossover point &

b) Copy everything in parent2 after this point to the new chromosome.

For offspring 2 do the reverse.

Chromosome 1	A B C D E F G H I J
Chromosome 2	0 1 2 3 4 5 6 7 8 9
Offspring 1	A B 2 3 4 5 6 7 8 9
Offspring 2	0 1 C D E F G H I J

Figure. 3.7. Single Point Crossover

The crossover point above is 2.

3.5.2 Two Point Crossover

Same as above except this time two crossover points are randomly chosen.

Chromosome 1	A B C D E F G H I J
Chromosome 2	0 1 2 3 4 5 6 7 8 9
Offspring 1	A B 2 3 4 5 6 7 8 J
Offspring 2	0 1 C D E F G H I 9

Figure. 3.8. Two Point Crossover

Here the crossover points were at 2 and 9.

3.5.3 Uniform Crossover

A certain number of genes are randomly selected to be "swapped"

Chromosome 1	A B C D E F G H I J

Chromosome 2	0 1 2 3 4 5 6 7 8 9
Offspring 1	0 B C D E 5 G H I 9
Offspring 2	A 1 2 3 4 F 6 7 8 J

Figure. 3.9. Uniform Crossover

3.5.4 Specific Crossover

A specific crossover for a specific problem For example, A permutation encoding is where one crossover point is selected. The gene is copied from parent 1, the second parent is scanned and if the number is not yet included it is added and then the process repeats.

3.5.5 Arithmetic Crossover

Some arithmetic operation is performed on the two strings to create a new string. Here it was AND.

Chromosome 1	1 0 1 1 1 0 1 1 1 0 1
Chromosome 2	1 1 0 0 1 0 0 0 1 1 1
Offspring 1	1 0 0 0 1 0 0 0 1 0 1

Figure. 3.10. Arithmetic Crossover

3.6 Mutation

Mutation means change (in gene). Its application is at gene level [36]. Mutation is needed as during reproduction and crossover phase there may occasionally any lose of potential genetic information. [13]. It introduces changes randomly into features of chromosomes. Normally mutation rate is quite small and is dependant upon chromosome length. So, the newly produced will not be much different from original chromosomes. Mutation role is critical in GA. It again introduces genetic diversity [36].

Mutation is usually carried out by flipping bit at mutation rate. Mutation depends on the GA Encoding Scheme as well as GA Crossover. The purpose of mutation operation is to maintain the population diversity. Mutation is dependent on the encoding as well as the crossover. After crossover, mutation takes place. Mutation changes the new offspring randomly.

For binary encoding, the bits are switch randomly from 0 to 1 or 1 to 0. Mutation is as follows:

Original offspring 1	1101111000011110
Original offspring 2	1101100100110110
Mutated offspring 1	1100111000011110
Mutated offspring 2	1101101100110110

Figure. 3.11. Mutation (Binary Encoding)

This is the chance that a bit within a chromosome will be flipped is a "Mutation Rate". i.e., 0 becomes 1, 1 becomes 0.

As mutation depends on both encoding and crossing over, so when we are having permutation encoding, the mutation is changing two genes which are as:

Original offspring	9 4 6 2 3 1 0 5 7 8
Mutated offspring	9 6 4 2 3 1 0 5 7 8

Figure. 3.12. Mutation (Permutation Encoding)

3.7 Effects of Genetic Operators

The role of genetic operators is very important. It has effect on the selection of chromosome because selecting alone without mutation will lead to a population having best individuals. If crossover is also done with selection, it will thus lead to causes the algorithm to have sub-optimal results. If mutation is done alone, there will be an arbitrary walk through the whole lot. Use of selection and mutation both in an algorithm, creates an efficient, noise tolerant and high performance algorithm [20].

3.8 Fitness function

Fitness value is associated with every chromosome and that fitness strength is evaluated by a function defined by user known as fitness function [27].

3.9 Who can benefit from GA

Nearly everyone can gain benefits from Genetic Algorithms, once he can encode solutions of a given problem to chromosomes in GA, and compare the relative performance (fitness) of solutions. An effective GA representation and meaningful fitness evaluation are the keys of the success in GA applications [21].

GA is useful, robust and proficient when:

- Having large, poorly understood and complex search space
- Domain sector is scant or expert information is difficult to encode for narrowing the search space.
- Mathematical analysis is not available.
- Failure in search methods which are traditional

The main advantage of GA is that can easily cater arbitrary types of objectives and constraints; all of them can be catered by assigning weight to the Objective function. GA is use for Modeling and as a Problem Solver. GA can be applied in many fields of entertainment, engineering problems, scientific and business including:

Optimization: GA has widely been used in optimization. It handles combinatorial and numerical optimization problems. Circuit designing, traveling salesman (TS) Problem are one of the few examples.

Automatic Programming: It is also used for developing programs for particular tasks and also for designing structures for computation. For Example, network sorting and automata

Robot and Machine Learning: GA has been used for robot and machine learning applications. From this neural networks can also been designed. It has also used for evolving rules for classification, designing and controlling of robots and production of symbolic systems.

Immune system models: Genetic Algorithms are used for modeling of human immune system during one’s lifetime and also at evolutionary stage.

Ecological models: GA has also been used for modeling ecological happenings including ecological flow of resources and symbiosis.

Genetics Models of Population: Used for studying genetics queries. For example, "what are the conditions under which gene for crossing over will be evolutionary feasible".

Relations between learning and evolution: It is also used for studying how learning of individual and evolution of species affect each other.

Models of social systems: Social systems evolutionary aspects can also been studies from GA, co operational evolution, communicational evolution, and ant colony behavior.

3.10 Artificial and Real Life… Interchangeable or Not?

In this advance era of technology, the progress will soon be beyond simulation and models of living things. It will be in the form of actual beings. Its not that it's been formed in the form of molecules, it's actually from computer but its behavior is as real as that of living things [21]. Thus living beings can simply be programmed from computer technology. Tomorrow's virus will be alive, if today's not [21]. This statement is called as **"strong A-life"** opposite to **"weak A-life"** [21]. There are two schools of thoughts regarding this statement.

According to the "strong A-life" school of thought foes, no matter at what stage the technology will reach, the life creation merely from programming through computer cannot be done. The arguments put forward by them are as follows:

- Life created by computer non-material entity.
- It cannot move around as its immotile
- It's don't have dying capacity
- Same individual's life span varies on different machines.

A programmed life is non-material while a living being is material entity; a thing that has mass and takes space, chemically composed and has physical properties too. A material thing satisfies life's definition; in take solid, energy utilization, excrete the unused, have reproduction tendency.

On the other hand the computer generated life does not assure this definition of life. It is immotile. Expansion and contraction are the only actions that can be detected from a programmed life [21]. It doesn't have dying capability. They cannot die actually.

As real beings are composed of molecules and are arranged in a delicate and complex manner. As soon as the living being dies out than arrangement is no more. On the other hand, the artificial being exits when the system on which its operating ceases. But program rarely dies out since machine turns on; the life generated by computer "resurrects" [21].

Individual's life span is dependent on the machine. It varies when the machine is different. When the same program (considered as an artificial being), is being run on different machines, it's supposed to have same have same instances and attributes [21]. Thus, it can be considered as same individual. So in this way the two instances have different live span because it any of the machine stops running, than the real life concept is no more exits [21].

The most well known supporter of **"strong A-life"** schools of thought "Christopher Langton", says the artificial being do not exist the way we know. It's a virtual medium where they live. He further supported his argument that models can be built so real that they can stop to be life's model and thus become life's paradigm themselves. He also claims that any of the definition in its broader spectrum includes all living beings, also includes objects of processes of computer and thus will be considered as "actually alive" [21].

Genetic Algorithms joins hidden good information of one solution with other solution that also has good information for producing good information solution inherited from parent solutions. That thus leads to optimality. The algorithm's ability of exploration and exploitation simultaneously, an emerging theoretical rationalization, triumphant real-world application problems supports and builds up that conclusion of GA as a robust and powerful optimization technique [20].

CHAPTER 4: MULTIOBJECTIVE GENETIC ALGORITHMS

Genetic Algorithms are most suited for solving multiobjective optimization scenarios. Thus a GA handling single objective scenario can be changed and a multiple set of solutions can be run in one go [36]. GA has ability of simultaneously searching different areas of a solution set. This ability thus supports it in finding a diverse solution set for difficult and complex problems. The crossing over operator exploits good solutions structure with respect to the Fitness Function for creating unexplored areas of Pareto Front [36]. Additionally, Multiobjective Genetic Algorithms (MOGA) often not require user for prioritizing, weighing or scaling objectives. Therefore, Genetic Algorithm is considered to be the most well known heuristic technique for designing of multiobjective optimization scenarios. Schaffer [37] proposed first MOGA; known vector evaluated GA (VEGA). Many methods for using GAs for coping up with optimisation of problem which is simultaneous over a lot of dimensions are being proposed, inclusive Pareto ranking use [41].

Genetic Algorithm is recognized as a well known and most popular for multiobjective optimization [38]. In general, multiobjective GA difference from each other is according to their fitness function, diversification techniques or elitism approach [36].

4.1 Design Issues and Components of MultiObjective GA

The design components and issues of Multiobjective GA described by Abdullah Konak, David W. Coit and Alice E. Smith are as follows:

4.2 Fitness Functions

4.2.1 Weighted Sum Approaches

In this approach weight is assigned to every objective function which is normalized. It's a classical approach and is called as Priori Approach as user has to give weights. Problem is solved through objective function against weights given then results into a single solution. If there is a requirement of multiple solutions, then the scenario must be handled multiple times after assigning different weights. Selection of weight vector for every run is the main issue in this method.

Advantage of this approach is that its implementation is simple. As one objective is in assigning fitness and it can be used with lesser modification. Additionally, this method is computationally competent. In this every Pareto-optimal solution can not be examined when true Pareto front is not convex. Thus, multiobjective GA founded on the weighed sum approach has is not good in finding out solutions which are homogeneously disseminated over non convex surface which is tradeoff [43].

4.2.2 Altering Objective Functions

Main benefit of alternating objectives approach is that it is straightforward to implement and is computationally competent like single-objective Genetic Algorithm. Basically approach is clear-cut extension of single objective Genetic Algorithm for solving multiobjective cases. The main downside of objective switching is population convergence to solutions that are better in single objective but is deprived in other objective.

4.2.3 Pareto Ranking Approaches

Pareto Ranking Approach unequivocally utilizes notion of Pareto dominance in evaluation of assigning or fitness selection possibility for solutions. Population ranking is as per dominance rule, and then strength value is assigned to every solution according to their rank.

4.3 Diversity

Diversity in population is an important aspect of multiobjective GA for getting solutions which are distributed uniformly over Pareto Front. Lack of preventive actions leads to form relatively less cluster in population. This is process is known as Genetic Drift. Approaches devised for preventing Genetic Drift are Fitness sharing, Crowding distance and Cell-based density and are described as below:

4.3.1 Fitness Sharing

In this approach the unexplored areas are searched by reducing artificially the strength of solutions in dense areas. For achieving this aim the heavily populated areas are targeted and then penalty approach is applied for penalizing solutions present in such areas. This pioneer of notion was DE Goldberg and J. Richardson [44].

4.3.2 Crowding Distance

In this technique fitness sharing parameters are not used. Its goal is to obtain unvarying distribution of solutions all along well-known Pareto Front. Its major benefit of that crowding distance approach is measure of density of population about a solution is evaluated devoid of requiring parameter defined by user.

4.3.3 Cell-based Density

In Cell-based Density Approach objective space is separated into m-dimensional cells. The solution count is referred as cell density and solution density is equivalent to cell density where solution is present. This information of density is used for achieving diversity; this aspect is similar to Fitness Sharing technique. The chief benefit of Cell-based Density technique is overall density map of fitness function space is acquire as a consequent of density computation.

4.4 Elitism

Elitism means the superlative solution obtained during search and it all the time survives in next generation. Thus every non dominated solution find out by multiobjective GA is known as elite solution. This approach is not that much easy as single objective GA optimization as in this probability of elite solution is more. In beginning the elitism approach is not used in GA. But in recent GA it is being used.

Two strategies used by Multiobjective GA for implementing elitism are [45]:

Keeping in population the elitist solutions

Accumulating in external secondary catalog the elitist solution and then reintroduce them in population

4.4.1 Strategies to Maintain Elitist Solutions in the Population

During random selection there isn't any surety of survival of non-dominated solution to next generation. A simple implementation of elitism technique in multi-objective Genetic Algorithm is copying nondominated solution in the population Pg to Pg+1. Then remaining Pg+1 is filled by choosing from rest of dominated solutions in Pg. This will only work when total count of nondominated offspring and parent solutions is larger than Ng. For addressing this problem many approaches are being proposed. Thus, in this approach size of population is main factor in GA.

4.4.2 Elitism with External Populations

When using external catalog for storing of elitist solutions, many issues occur. First issue is that which solution will be stored in elitist catalog C. Most multiobjective GA store non-dominated solutions identified so far during the search [11], and E is updated each time a new solution is created by removing elitist solutions dominated by a new solution or adding the new solution if it is not dominated by any existing elitist solution. The operation for solution of this issue is expensive computationally. Numerous data structures [46, 47] are being proposed for efficient updates, storage and searches for catalog C.

Other issue is size of catalog C. As there might exist a very huge figure of Pareto optimal solutions for the problem. The elitist catalog can cultivate exceedingly large. Thus pruning methods are being proposed for controlling size of C.

4.5 Constraint Handling

Real world's optimization problems which include constraints must be resolved. A tremendous survey [48] on techniques for handling constraints in evolutionary methods is presented by Coello.

A single objective Genetic Algorithm may use any of the four following constraint handling methods:

- Discarding solutions which are infeasible. This is also called as "Death Penalty"
- By use of penalty function strength of infeasible solution can be lessened
- Craft operators for always producing practical solutions
- Transformation also called as Repair of infeasible solutions to feasible

Constraints handling is not satisfactorily explored in multiobjective Genetic Algorithm [50]. For example, all key multiobjective Genetic Algorithms presuppose problem exclusive of constraints. Although strategies (a), (c), and (d) for handling of constraint are applicable directly to multiobjective scenario. Major drawback of this process is that it is complex computationally and supplementary parameters like reference set size and size of niche. However amendments are possible.

4.6 Parallel and Hybrid MultiObjective GA

All multiobjective Genetic Algorithm comparative research and studies agrees that preserving diversity and elitism mechanism improves efficiency and performance. However, it requires usually extensive effort in computation and memory. Additionally, computing fitness function may take substantial time in real world scenarios. Thus researchers are interested in reducing time of execution and multiobjective Genetic Algorithm's requirement of resources by use of advance data structure.

Hybridization of Genetic Algorithm having local search algorithm is applied frequently in single objective Genetic Algorithm. The approach is referred as "Memetic Algorithm" [49].

Important issues in multiobjective Genetic Algorithm's hybridization by local searching algorithms are:

- Selection of solution for applying local search
- Identification of solution in neighborhood as new optimum or best result in the presence of local non dominated solutions.

In the majority of real world engineering scenarios simultaneously involve optimization through multiobjective where trade-offs consideration is vital. In preceding decade, genetic evolutionary techniques are primary tools for solving problems of multiobjective which are associated with real world.

This chapter presented multiobjective GA and focused on their encountered issues and components during implementation of multiobjective Genetic Algorithm.

CHAPTER 5: MULTIOBJECTIVE OPTIMIZATION

5.1 Overview

For working out multipurpose engineering and scientific problems [30, 36] is a very challenging task. Multiobjective Engineering problems are cost minimization, increase performance, maximization of reliability [36]. These problems are difficult but are pragmatic [36]. In such optimization problems, there is a conflict in objectives [30, 36], averting concurrent optimization of every defined objective [36] and thus there may be a requirement of complex computations [30]. The range of multiobjective optimization problem is from linear objective function to Pareto-based methods [30]. There isn't any specific algorithm or problem domain defined. Multiobjective Optimization is also called as "Multiperformance Optimization" or "Multicriteria Optimization" [31].

The area of Multiobjective Optimization Problems is extremely complex and mathematically difficult, with many under-researched areas and outstanding problems [40].

Therefore, Genetic Algorithm is a very well known meta-heuristic which is most suited for these kinds of problems. GA is tailored to handle multiobjective problems through the fitness functions and also it commences methods for encouraging diversity in solution [36].

MOP consists of objectives, decision variables and constraints in which few or all objective functions are non linear or linear. The evaluation function maps the decision variables to vectors. The mapping may or may not be on some area of objective function set which is dependant upon constraints and functions composing specific MOP. MOPs are exemplified by discrete measure of objectives (the performance) that may be (in) reliant and/or disproportionate. Multiple objectives as being optimized mostly always conflict, places partial instead of total ordering in solution space. MOP may require specialized techniques for optimization due to such characteristics (constraints and conflicting, multiple objectives) [30].

5.2 General Approaches to MultiObjective Optimization

The two general approached for multiobjective optimization are as follows [36]:

5.2.1 Combining every objective function into a single objective function or to move all but a single objective to the constraints

First approach is to combine all objective function into a single objective function or to move all but a single objective to the constraints.

- The earlier case, the single objective determination is achievable with methods as mentioned in [36] but main objective is the selection of utility or weights functions for defining decision making preferences. Practically, precise and accurate selection of weights is a difficult task, even for the one who is known with the problem domain. Thus, compounding the shortcoming is that there is a need of scaling among small perturbations and objectives in weights as it at times directs to relatively different solutions.
- In second case, problem is moving the objective in constraints function set. Thus a constraint value is being made for individual pervious objectives. This is mostly random.

In the both of above cases the optimization techniques returns a single solution rather than a multiple solutions. This is the reason for which GA is preferred for considering in multiple domains.

5.2.2 Pareto optimal set of solution

In the Second approach whole Pareto optimal set of solution or a subset is determined. A Pareto Optimal Set is solution set that not dominated with respect to one another. There is always a sacrifice for obtaining gain the other objective, when move single Pareto solution to other. Pareto optimal set of solution is mostly preferred for single results as in real life domain problems they seems practical. The end result of a decision-maker always is a trade-off. These sets may be of variable sizes, but the set size increases with increase objective [36].

5.3 Multi-Objective Optimization Formulation

A multiobjective decision making problem is explained as mentioned [36]. Given is an m-dimensional variable vector a in the solution space POS:

a={a1,…,am}

Find a vector a* which minimize a set of OBF objective functions

b(a*)={b1(a*),…,bOBF(a*)}

The solution space named POS in general bound on the decision variables and is limited by a sequence of constraints, such as

gj(a*)=bj for j = 1, …, m,

In real life optimization problems, there is a conflict in objectives under consideration [30, 36]. Thus optimization of a in respect to an individual objective frequently results in unacceptable state in respect to further objectives. As a result, an ideal multipleobjective solution which concurrently optimizes every objective function is nearly impractical. Investigating a solution set is a practical solution to a multipleobjective problem. Each thus satisfies objectives at an adequate level devoid of being subjugated by another solution.

If all objective functions are for minimization, a feasible solution a is said to dominate another feasible solution

y (a > y)

if and only if, bi(a) ≤ bi(y) for i=1, …, OBF and bj(a) < bj(y) for least one objective function j

If a solution is not dominated by another solution in a solution set space, then that solution is known as “Pareto optimal”. Improvement of solution of Pareto optimal can not be done in respect to another objective without deterioration of atleast another objective.

The set of entire possible 3 non-dominated solutions in POS is considered as “Pareto optimal set”. A in given Pareto optimal set, the resultant objective function values in objective space of solution is called the “Pareto front”. For most of problem, the Pareto optimal count of solutions is massive.

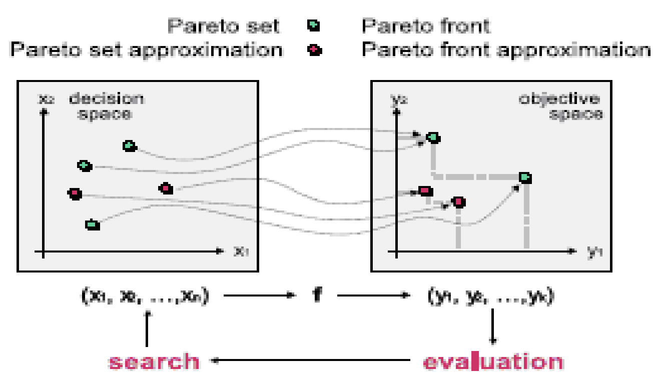

Figure. 5.1. Illustration of a General Multiobjective Optimization Problem

5.4 Pareto optimality Concepts

Pareto optimality concepts and its related notions and terminology are used frequently by decision makers or researchers who often use in literature. For better understanding, Pareto optimality, Pareto dominance, Pareto front and the Pareto optimal set are defined here. Examples are found in Van Veldhuizen (1999) [30]. Key concepts of Pareto are defined mathematically as mentioned:

5.4.1 Pareto optimal set of solutions

Pareto optimal set of solutions is also called as admissible, efficient or non-inferior solutions. Their subsequent vectors are known as "nondominated" [30]. The solutions have apparently no clear relation, a part from their membership in Pareto Optimal Set. The set of entire solution is formed having subsequent vectors which are not dominated in respect with all other vectors which are in comparison. Thus Pareto Optimal Solutions are "Classified" as based on computed functional values.

5.4.2 Pareto Front

When these computed values are plotted in Objective Solution space, then the nondominated vectors are communally called as Pareto Front. The Pareto optimal set is basically subset of entire feasible or potential solutions in Ω [30]. In calculated or estimated Pareto Front in which every vector is not dominated in respect with entire objective vectors which are generated by evaluating entire feasible solutions in the form of Ω. The decision maker frequently chooses solutions through selection of adequate objective performance which is represented by Pareto front.

Selecting MOP solution which optimizes mere objective may ignore solutions which from on the whole a perspective is "better." The Pareto optimal set includes such "better" solutions. Identification of Pareto optimal solutions is a Key for Decision Makers for selecting an optimum solution which satisfies the objective function at best. The preciseness of the DM's point of view is dependent on true Pareto optimal set and also on set which is presented as Pareto optimal.

The multiobjective optimization mostly offers limited points which may or may not truly be a Pareto Optimal points. Thus a real life can be modeled upon machine (computer). There will be a loss of fidelity between reality based model and discrete, finite model. Compound multiobjective optimization doesn't provide them analysis determination of real Pareto Front.

5.5 Ultimate Goal of a Multi-Objective Optimization Algorithm

Identifying solutions in Pareto Optimal Set is the eventual goal of multiobjective optimization. Though, Identifying all Pareto Optimal Set for lots of multiobjective scenario is almost impracticable. The reason behind this is its size. Additionally solution optimality's evidence is computationally impractical such as, combinatorial optimization scenarios. Thus practical method for multiobjective optimization is investigating solution set which includes Pareto Optimal Set to the degree that is feasible or best known.

5.6 Multiobjective Optimization Approach Conflicting Goals

Keeping all of these apprehensions in view, multiobjective optimization technique must have to accomplish below mentioned conflicting goals [36]:

Best Pareto front have to be close to True Pareto Front. Preferably, Pareto set should have to be Pareto Optimal Set's subset. This goal is served best by intensifying search on a specific Pareto front Region.

Pareto set having best solutions must be distributed uniformly. This thus provides a actual picture of trade off to the decision maker. The demand of this goal is search effort which should be distributed uniformly over Pareto front.

Additionally, entire Pareto front's spectrum should be imprisoned by best Pareto Front. Therefore, there is a requirement of investigation at intensive level of Objective Functions. In this goal main aim is the extension of Pareto Front on both sides, thus exploring further more solutions.

CHAPTER 6: PROPOSED ROUTING OPTIMIZATION STRATEGY USING GENETIC ALGORITHM

The contribution of this research work is in the field of Network Optimization and Artificial Intelligence's sub-field of Genetic Algorithm. An effort has been made to join both of these fields and hence contributing in the research field. As Optimization is the best way of selecting the appropriate solution from given options and Genetic Algorithm is outstanding for all tasks requiring optimization and is highly useful in any state of affairs. A genetic algorithm is a global technique of optimization [12] which is based on the Natural Selection [11] [1]. The optimized Shortest Path Problem (SPP) is based on the analogy of finding shortest path between the source node and destination node in a network. Optimization is ideal for routing and efficiency and performance of the Data Network. The routing problem thus can be simplified by optimization [14].

Genetic Algorithms combine the good information hidden in a solution with good information from another solution to produce new solutions with good information inherited from both parents, inevitably leading towards optimality.

The network traffic is increasing rapidly and there should be balance in broadcast and quality of service of the network traffic. Thus the optimization is needed badly for network control of QoS and load balancing. If network load increases, it will affect the quality of service as it will cause delay in the network.

In this research work, a novel approach of optimized routing is introduced which is based on Genetic Algorithm. It finds out the optimum and the most suitable path which is qualifies Objective Function criteria. In a network, if the packet is taking more time in transmission from source to destination, than more will the delay. The Objective Function criteria is that network bandwidth should be available, the delay will be less and if there are paths which have same delay factor value, than the path having less number of hops will be selected. Hence, we are handling Traffic Engineering aspect in networks.

GA helps in simplifying many elements in network path optimization. In this research work certain constraints of bandwidth, utilization and delay are defined, which than evaluates the quality of the genetic string. We thus develop a routing approach which is based on Genetic Algorithm. The chromosome is representing the nodes of that path and path can be of variable length. It is not merely efficient and robust but it congregates promptly.

This research work is proposing algorithms for the routing through a network using GA optimization:

Algorithm I:

Network Path Optimization using GA approach by using utilization and bandwidth constraints based on the utilization factor proposed by M. Ericsson, M.G.C. Resende and P.M.Pardalos approach [5].

Algorithm II:

Routing Optimization Strategy using Genetic Algorithm utilizing bandwidth and delay metrics

These algorithms are described as follows:

6.1 ALGORITHM I

6.1.1 Genetic Representation

We represent the individuals of the population through chromosomes. Each chromosome consists of a string of numbers. Each number represents a node and the string represents a path in the network.

1	3	5	4	6

Figure. 6.1. Genetic Representation of String

6.1.2 Objective Function

The objective function of the GA for optimization of network path works as follows. It first checks the bandwidth availability and then finds out network utilization factor through:

$$utili_i = \frac{DataSize}{bwAvailable_i} \quad (1)$$

We use the M. Ericsson, M.G.C. Resende and P.M.Pardalos approach of Genetic Algorithm for the weight setting problem in OSPF routines for setting limits [5]. Then, they are assigned to utilization factor between two respective nodes:

$$Limit_i = condition(utili_i) \quad (2)$$

Afterwards, these limits are summed up as:

$$cost = \sum_{i=1}^{n} Limit_i \quad (3)$$

Thus this constitutes our Fitness Function:

$$Fitness\_Function = cost \qquad (4)$$

We measure the delay factor of the path as:

$$delay = \sum_{i=1}^{n} \frac{bwAvailable_i}{DataSize} \qquad (5)$$

In this way, we measure the delay and utilization factor. Our optimum path selection is mainly based on the Utilization factor against which we measure the delay factor involved in the path.

6.1.3 Constraints

Most constraint satisfaction measurements are on the basis of Resource Profiles. These profiles define availability or consumption of resource as a time function [29].

Constraints defined in this algorithm are:

- Bandwidth Availability
- Utilization

6.2 Proposed GA for Path Optimization

Our proposed Genetic Algorithm for network path optimization consists of following steps:

6.2.1 Initialization of Population

The chromosomes or the path is randomly generated. So the probability of invalid paths (or chromosomes) also exists. Thus the population consists of both valid and invalid paths. We have done this for validating the efficiency of the algorithm in all scenarios.

1	3	4	5	6

Figure. 6.2. Structure of a chromosome

The chromosome 1-3-4-5-6 is showing the path from the source node 1 to destination node 6. The total number of hops in this case is 4.

The chromosome structure thus satisfies the requirement for the determination of Best possible shortest path of a network. The chromosome length is kept variable. The more the length of the chromosome, the hops the data have to travel through the network are more.

In initialization phase, the first and the last gene of the chromosome that represent the source and the destination node of the network are only added into the population. We discard the rest of the generated chromosomes.

6.2.2 Parents Selection

After random generation, the fitness value of each chromosome is calculated. On the basis of this value, the Best Fitted parents are selected for cross over. The parent selection is not random.

We use Rank Selection for selection of parents. In the Rank selection, we first rank the population. Afterwards, every chromosome receives fitness from this ranking. The chromosome with maximum Fitness value is selected as the parent. Random generation may deviate from the target chromosome. After experimentation, we have selected Best fitted chromosome instead of both worst fitted chromosomes or one best and one worst fitted chromosome.

6.2.3 Reproduction & Crossing Over

The Best Fitted parents cross over and produce offsprings. The technique proposed and implemented in this research work is as follows:

6.2.4 Proposed Crossing Over Technique: 2-point over 1-point crossover

We have proposed a *"2-point over 1-point crossover"* for our path optimization algorithm. This cross over works is shown as follows.

- First we apply 2-point crossover over the selected chromosomes.
- After this, 1-point crossover is applied to the crossover part i.e. the crossover point chosen is 2.

Chromosome A	1	2	3	5	6	7	8	9
Chromosome B	1	4	3	5	7	6	8	9
Offspring A	1	4	3	5	7	6	8	9
Offspring B	1	2	3	5	6	7	8	9

(a)

Chromosome A	2	3	5	6	7	8
Chromosome B	4	3	5	7	6	8
Offspring A	**2**	**3**	5	7	6	8
Offspring B	4	3	**5**	**6**	**7**	**8**

(b)

Offspring A	1	**2**	**3**	5	7	6	8	9
Offspring B	1	4	3	**5**	**6**	**7**	**8**	9

(c)

Figure. 6.3. Two Point over One Point Crossover (a) Two Point Crossover (b) One Point Crossover (c) Final Offsprings

6.2.5 Mutation

The resulting offsprings are mutated. Mutation has been done in various ways according to the requirement based on the gene of the chromosomes:

Step 1: If there is any repeating node, then trace its location. Find out missing node in the network, than replace the traced repeated node with the missing node.

Step 2: If there is no missing and repeating node, randomly choose two genes of chromosome and swap them.

Step 3: If the chromosome size is minimum (minimum case will be between two nodes), then there will be no mutation in it, because it is only representing the source and destination node and none of them could be flipped.

Step 4: If size is one more than the minimum (i.e, 3 nodes) than only the midpoint value is mutated with the randomly generated missing node.

Offspring A	1	2	3	5	7	6	8	9
Offspring B	1	4	3	5	6	7	8	9

(a)

Mutated Offspring A	1	2	3	8	7	6	5	9
Mutated Offspring B	1	4	7	5	6	3	8	9

(b)

Figure.6.4. Mutation Operation for no repeating nodes case
(a) Before Mutation (b) After Mutation

6.2.6 Evaluating fitness value of the chromosomes

The fitness values of the mutated offsprings are calculated. If there is any chromosome in the population having fitness value worse than the produced offsprings, then that chromosome is discarded and the “Fitted” offspring is added into the population. There will be no effect on the size of the population; it remains constant.

The Primary condition is the availability of bandwidth. If the bandwidth is not available, that chromosome will not been considered. The fitness value of each path is calculated on the basis of the Objective Function.

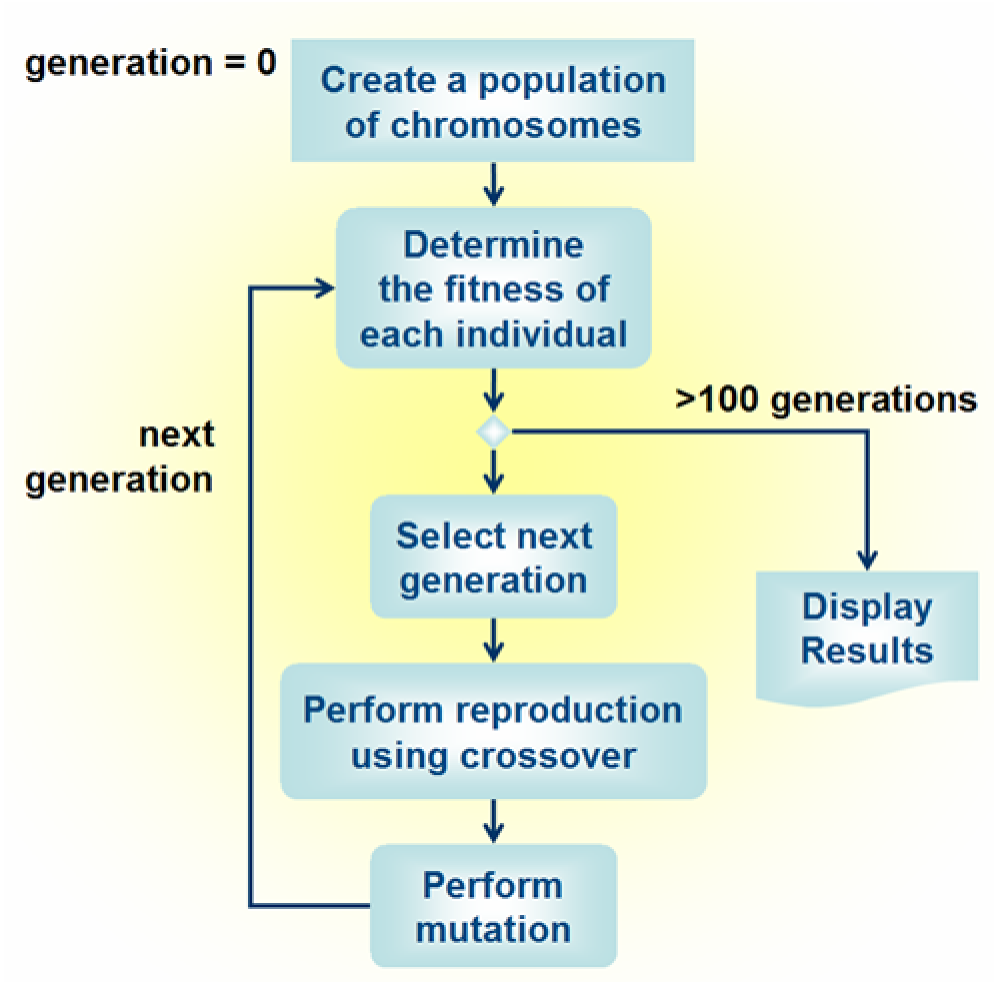

Figure.6.5. Flowchart of the Genetic Algorithm for Path Optimization

6.3 ALGORITHM II

6.3.1 Representation of String

The individual chromosome in the population is represented by a string of numbers. Each number is representing the nodes while the string is representing network path as shown in Figure 6.2.

The chromosome 1-3-5-4-6 as shown in Figure 6.2 thus shows the nodes in the path, starting node 1 and destination node 6. The number of hops for this case is 4. The length of chromosome is variable; not fixed. So, the more number of nodes in the chromosome more will be the number of hops the data travel. The chromosome which have source and destination node same as defined, they than constitutes the population lot. From that population lot the random selection is being done. The rest of the chromosomes are not considered.

6.3.2 Fitness Function

The Fitness Function first checks the availability of the bandwidth of network link and then finds out network delay factor of link. After that the average of delay of the path is taken by dividing the delay with total number of links in the path. All these paths are valid path as their bandwidth availability is checked before finding out their delay factor.

$$AveragePacketDelay = \frac{delay}{no.\ of\ links}$$

Fitness is calculating through Delay function.

$$delay = \sum_{i=1}^{n} \frac{bwAvailable_i}{DataSize}$$

Our optimum path selection is mainly based on the Bandwidth, Delay Factor and hop count. The more the bandwidth, the less the delay and the hop count the more optimum the path will be. When there are two or more paths with same fitness value than the path with less hop count will be selected.

6.3.3 Constraints

Most constraint satisfaction measurements are on the basis of Resource Profiles. These profiles define availability or consumption of resource as a time function [29].

Constraints defined in this algorithm are:

- Bandwidth Availability
- Delay

6.4 Proposed GA for Path Optimization

Proposed Genetic Algorithm for network path optimization is as following:

6.4.1 Population Initialization

The population is generated randomly. So, the valid and invalid paths probability is there. Whole population generated has all possible paths in it. The start and destination nodes of chromosomes are fixed. The population selected randomly in this case is 33% of the whole lot and is generated 30 times.

6.4.2 Parents Selection

The parents are selected on the basis of their best fitness function value. The parent's selection is thus Ranked Based Selection. They are not selected randomly. The Best Fitted parents are than cross over.

In Rank based selection, the chromosomes in the population is ranked on the basis of their Fitness Function value. The chromosome having high Fitness value is selected as parent. Both best fitted chromosomes are used for crossing over as they give offsprings which have tendency of surviving in the natural selection.

6.4.3 Recombination (or Crossing Over)

After crossing over of Best Fitted parents, the offsprings are produced. The main contribution in this research work is of proposing a new crossing over method called as "2-point over 1-point crossover". This method has been implemented in our path optimization algorithm. In this method 2-point crossover over is applied over the selected chromosomes. It segregates the source and destination nodes. Than 1-point crossover is applied to the rest of the chromosome. The crossover point in 1-point crossover is 2.

6.4.4 Mutation

Mutation is implemented on the resulting offsprings. But it has been done as per the scenarios. Scenarios are:

Case 1: In case of repeating node in the chromosome, the location of the repeating node is traced out. After that missing node is find out, which is than placed as the traced location.

Case 2: In case if there no missing or repeating node than randomly pick two nodes from that chromosome and swap them.

Case 3: If length of chromosome is minimum i.e, 2 (as it has only source and destination nodes), then there will not be any mutation in it.

Case 4: If length of chromosome is one more than the minimum i.e, 3, than in that case only middle value is flipped with the missing node.

6.4.5 Evaluating fitness value of the Mutated Offsprings

The mutated offspring's fitness value is than calculated. The chromosomes having worst fitness value than the offsprings are than replaced by the mutated offsprings. The worst chromosomes are than discarded. The size of population remains constant.

The fitness function first checks the bandwidth availability. If the required bandwidth is not available than that path is not considered for the Optimum Path. Otherwise, the Delay factor and the number of hops of the path are calculated. The survival of each path is calculated on the basis of the Objective Function.

CHAPTER 7: EXPERIMENTAL RESULTS AND ANALYSIS

This section describes strategy for experimental evaluation of algorithm proposed. Firstly, the experiments and the testbed used to carry out these experiments are described. Then the results of the experiments are presented and discussed in detail. The experiment shows the proposed algorithm is very efficient in finding the optimized path in a network with those constraints.

Experiments and Testbed

Now we describe the details of the experiments and the testbed.

7.1 Network Structure

We evaluate the performance of our proposed algorithm on the following network.

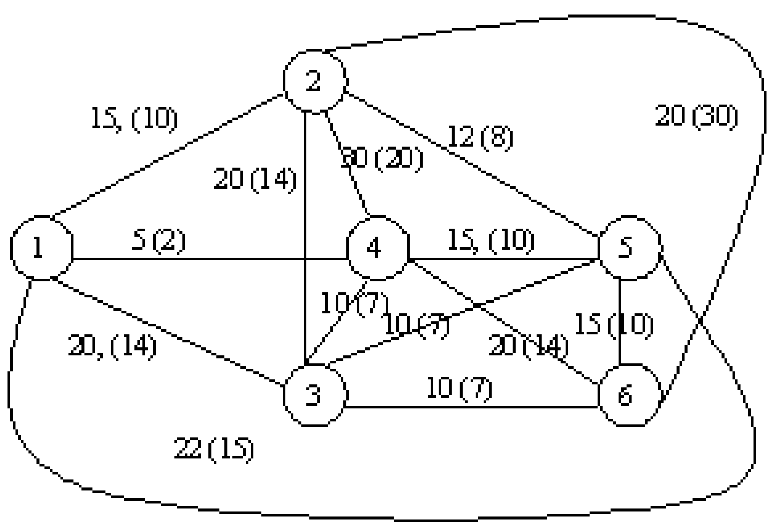

Figure.7.1. Network Structure

The network consists of six nodes/routers. The fundamental components of this network are node and connecting link. The two nodes are connected through a link which represents the distance between them. The more the distance, more the time packet will take to reach the next node and vice versa. The existence of link between two nodes does not necessarily imply that the packet can pass through that path; the data can only get through the path if the path meets the bandwidth requirement.

The path optimization algorithm has been coded in C++ language. In this structure, different scenarios have been catered for; for example, for checking bandwidth availability, the packet size of 10MB has not been allowed to pass through the link connecting nodes 1 and 4 because required bandwidth is unavailable.

7.2 Population Selection Scenarios

The experiments have also been carried out for population selection and have been divided into following categories:

- Blocks
- Half
- Random selection
- Full

All these experiments, repeat has been done five times and then the average result is reported. Therefore, the results are statistically significant and not random occurrences.

7.2.1 Blocks

In this experiment, the initial population is divided into chunks of alternative 5 chromosomes and then adds it into a population and all the generations are traversed.

Another experiment also consists of 5 chunks but first 5 chromosomes are not included, such that all possible options can be included in it. This procedure is repeated on 10 alternative chunks.

In experiments of blocks of 5 and 10 there is a variation in the results as shown in as shown in Figure 7.2 & Figure 7.3. Both graphs showing the utilization and delay results. So, this has not been selected for population selection.

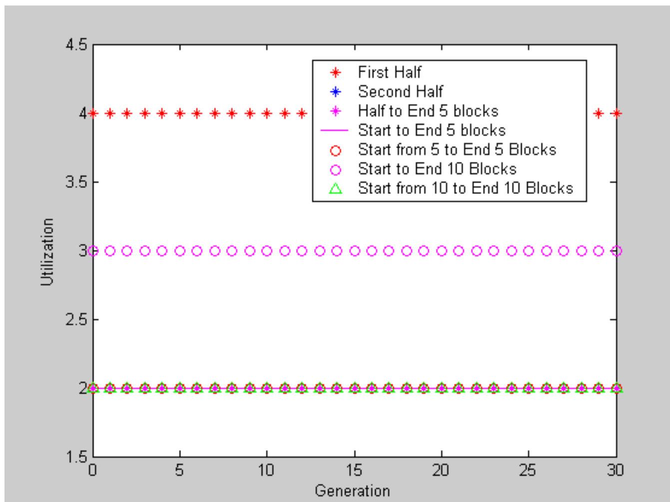

Figure 7.2. Utilization shown for half and blocks of population in 30 generations

7.2.2 Half

In this setup, the initial population has been divided into two parts and the computation has been performed on both divisions independently.

In experiments of half of the population there s a variation in the results as shown in Figure 7.2 & Figure 7.3. Both graphs showing the utilization and delay results. So, this has not been selected for population selection.

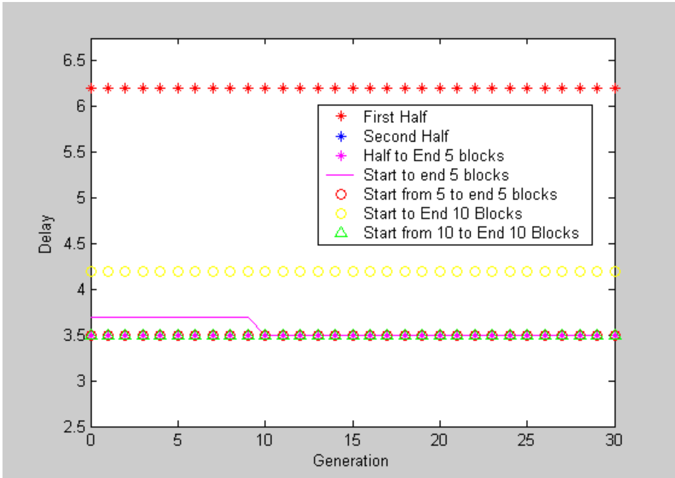

Figure 7.3. Delay shown for half and blocks of population in 30 generations

7.2.3 Random

In this scenario, the population has been selected randomly from the whole lot. The number of chromosomes i.e., population size is chosen as per requirement. The population has been divided into 30, 50 and 100 chromosomes. All of them have been iterated 30, 50 and 100 times each as shown in figure 7.4.

The utilization graph (Figure 7.4) shows that the result is best with 100 population size and 30 generations.

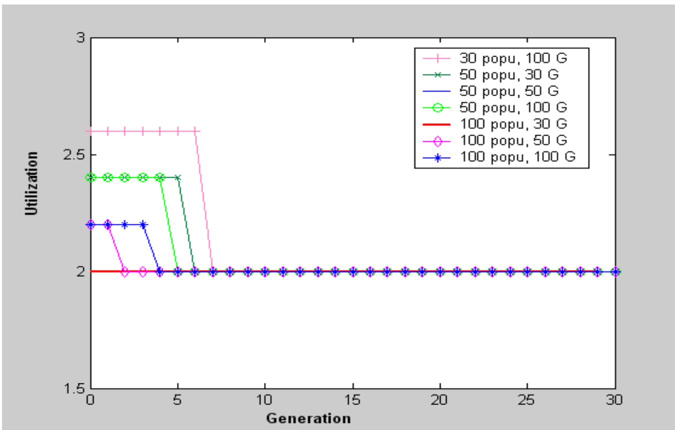

Figure 7.4. Bandwidth Utilization in 30 Generations

After taking averages, the population size 30 which are generated 30 and 50 times is discarded, as it's not giving the required results in one of the randomly selected population as it has a probability of not producing required results. The rest of the results shows that the population size 100 generated 30 times is giving optimum result.

The relevant result as per delay is shown in Figure 7.5.

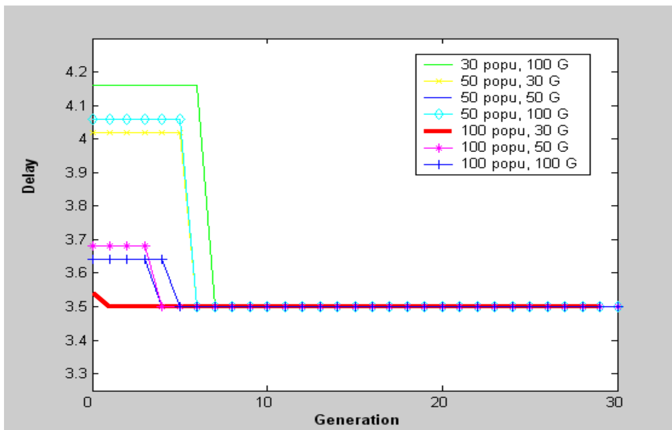

Figure 7.5. Delay in 30 Generations

7.2.4 Full

Here, the experiment has been performed on the whole population. The whole population is once generated and the results are obtained.

Optimum Required Delay: 3.5

Optimum Required Utilization: 2.0

7.3 Experimental Procedure

The experiments are done over a network consisting of 6 nodes called/routers. The proposed two algorithms are implemented as follows:

- In the first algorithm bandwidth and utilization of the whole population is calculated. The third parameter ‘Time’ is also involved. Time is an indirect measure of the distance traveled by the packet. The notations shown here are like, Bandwidth (Time). Various types of bandwidth and time constraints are included in it, so that we can also get the better result in a complicated network.
- In the second algorithm main factors Bandwidth Availability and Delay involved in the network traffic. In this the third parameter Time is also included.

7.4 RESULTS

The algorithm is then applied over the network for finding the optimum path based on the required criteria. The population is selected randomly. The slots of 30, 50, and 100 of population are generated 30, 50, 100 and 300 times. The random experiments are repeated 5 times. Then the average of the results is calculated and presented as the final result.

The population selection experiments are done on the first algorithm. In the second algorithm the obtained results are applied over the network with their respective constraints.

ROUTING OPTIMIZATION STRATEGY USING GENETIC ALGORITHM UTILIZING BANDWIDTH AND UTILIZATION METRICS

Now the results of experiments are presented and discuss in detail. The results show that proposed path optimization algorithm is robust in finding optimal path in a network while meeting the constraints of bandwidth utilization and delay.

The utilization graph (Figure 7.6) shows that the result is best with 100 population size and 30 generations. The optimum result is obtained from first till last iteration from our algorithm in the above mentioned population and generation. 'X' iterations generated are repeated 5 times and than average is obtained to ensure the efficiency of the algorithm.

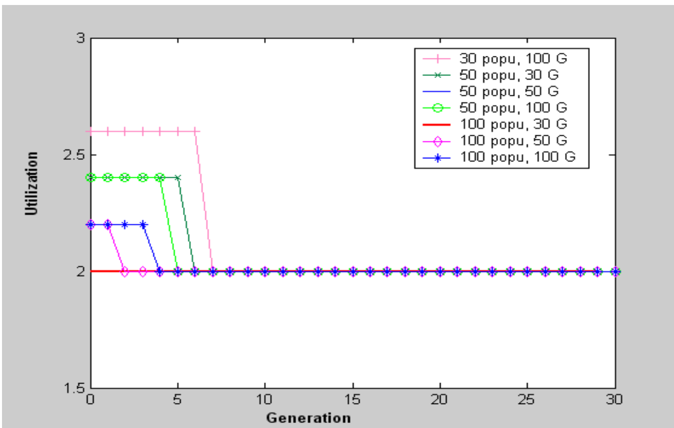

Figure 7.6. Bandwidth Utilization in 30 Generations

After taking averages, the population size 30 which are generated 30 and 50 times is discarded, as it's not giving the required results in one of the randomly selected population as it has a probability of not producing required results. The rest of the results shows that the population size 100 generated 30 times is giving optimum result.

The relevant result as per delay is shown in Figure 7.7.

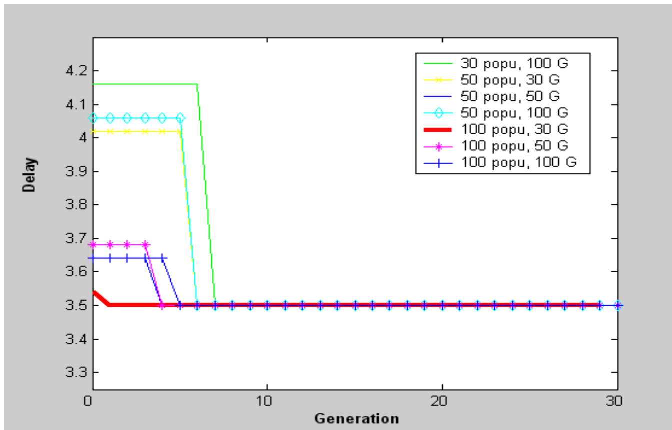

Figure 7.7. Delay in 30 Generations

This shows that in all options (as shown in table 1) the optimum path is obtained before 10 generation. From our algorithm the optimum result is obtained till 8th generation in all random cases as shown in table1. The best result is get in 100 population generated 30 times, in which the optimum path is get after 1st generation.

The whole population is once generated and the results are obtained.

Optimum Required Delay: 3.5

Optimum Required Utilization: 2.0

These results are compared with the results obtained as shown in table 7.1. This table is showing results obtained after running the algorithm 5 times over the network having different population set; randomly generated. In this case the result is near to the desired path. While in the rest though the generations are less but the optimum result is not achieved as early as in this case.

Hence, taking population 66% and generate it 30 times (for being on safer side) will attained the required result, though the optimum path is attained at the 2nd generation. The less the utilization, the more suitable the path will be.

In the rest of the experiments of half and blocks of 5 and 10, (as shown in figure 7.8 & figure.7.9) there is a variation in the results as shown in graph. Both graphs showing the utilization and delay results. So, this has not been selected for population selection.

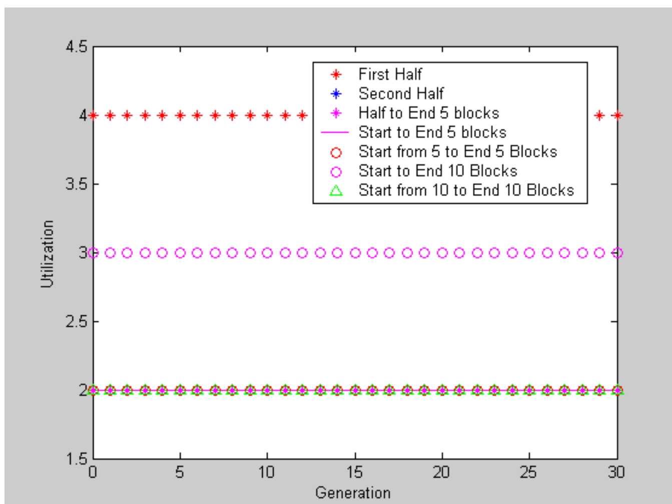

Figure 7.8. Utilization shown for half and blocks of population in 30 generations

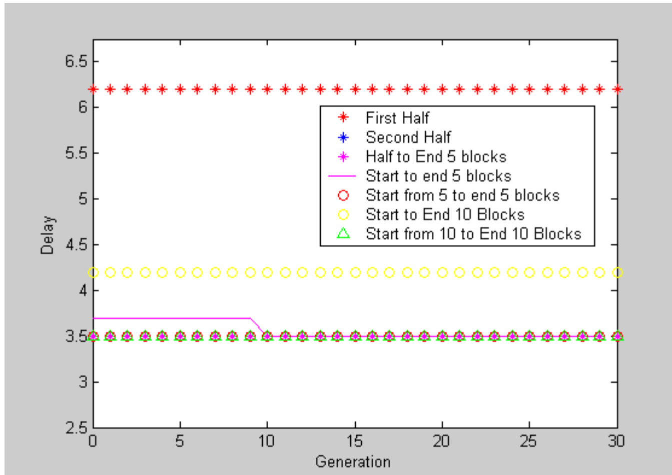

Figure 7.9. Delay shown for half and blocks of population in 30 generations

Population	Generation	Delay Start	Delay End	Gen_Delay # *	Utilization Start	Utilization End	Gen_Utili # *
30	100	4.16	3.5±0	7.8	2.6	2.0±0	7.8
50	30	4.02	3.5±0	6.8	2.4	2.0±0	6.8
50	50	3.64	3.5±0	5.4	2.2	2.0±0	5.4
50	100	4.06	3.5±0	7.4	2.4	2.0±0	6.4
100	**30**	**3.54**	**3.5±0**	**1.2**	**2**	**2.0±0**	**1**
100	50	3.68	3.5±0	4.4	2.2	2.0±0	3.4
100	100	3.64	3.5±0	4.8	2.2	2.0±0	4.8

Table 7.1: Showing Delay and Utilization Results obtained

Gen_Delay # ***** Generation # at which Optimum Delay is obtained

Gen_Utili # ***** Generation # at which Optimum Utilization is obtained

ROUTING OPTIMIZATION STRATEGY USING GENETIC ALGORITHM UTILIZING BANDWIDTH AND DELAY METRICS

The randomly selected population result is compared with the whole population generation and also with the results of the [7]. The results thus obtained are better than both of them. In random selection the results are obtained twice and then the average result is shown here. Therefore, the results are statistically significant. The criterion is:

Population Size	33%
Generations	30

Table 7.2: Criterion of Iteration of Population

First we check the bandwidth availability and then calculate the delay. Then the algorithm is applied over the network for finding the optimum path based on the required criteria.

The whole population is first pass through the algorithm as shown in the graph below. The population is generated 50 times. The average Delay minimum is 1.166667 and is till 50 iterations. So, there will be a delay in getting the optimum result.

Another experiment is done by randomly selecting the population and the algorithm is applied over it. The population size is 33% and the generation is produced 30 times. The process is repeated 4 times and than the average is taken. Average result of random selection is shown in the graph below (asterisk sign). Thus there is an improvement in this graph.

The network structure is showing few possible path as it is not possible to show all paths. After the genetic algorithm optimization strategy is being applied on the network, the optimized the optimized path (red colored path) is shown as:

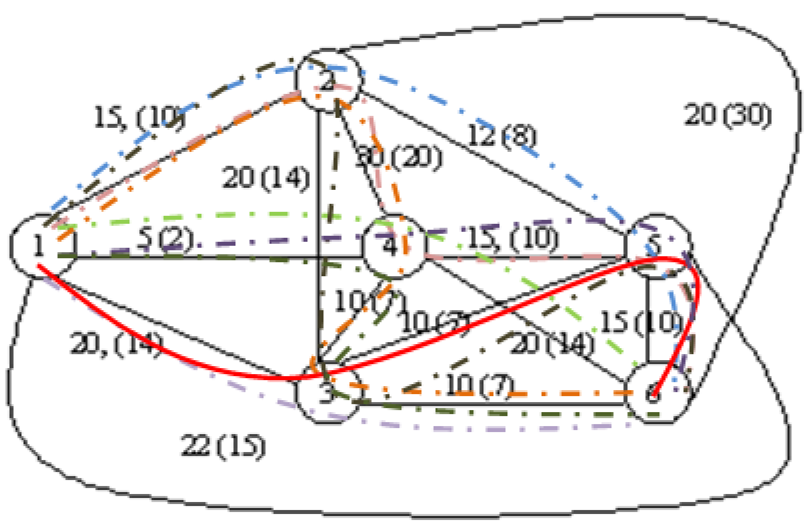

Figure 7.10. Optimized Path

The graph is near to solution in 9th generation while the full population is constant till 50th generation.

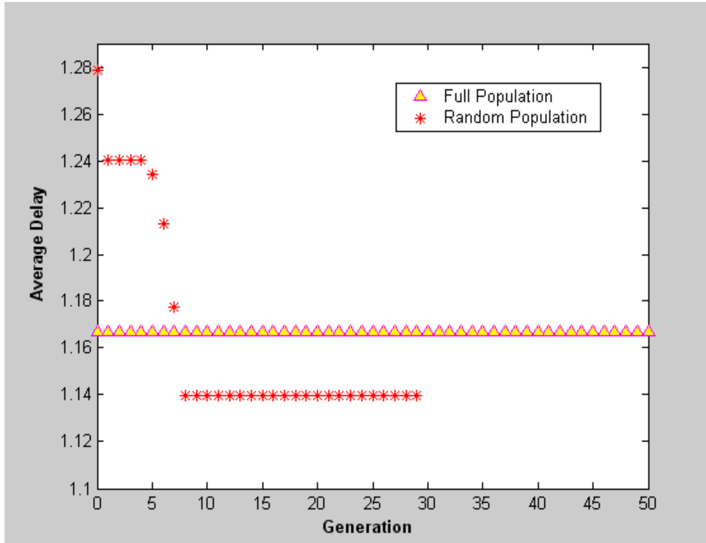

Figure 7.11. Showing Average Delay in 50 Generations (both of Full Population and Random Selection)

The number of hops in the random population selection is also decreases. At start the hops count of the paths are more as per the network scenario, but in final result the number of hops is lesser. Thus it is also handling the traffic engineering problem of the network. This is thus sharing and balancing the network load as per bandwidth and delay criteria.

In the full population case the number of hops will remain same and there is no change in the hop count till the 50 generation. It will show some change in later stage.

After this the random population is taken which is 33% of the whole lot and is iterated 30 times. The same procedure is repeated and than the average is taken. The graph is hence showing their average result.

As mentioned in the work of [7] the result improves after 20th generation while in this the optimum result is achieved in early generations. So, there is an improvement in the result shown by our experiments. The number of nodes is also more than those mentioned on work of [7].

The distribution of Shortest Average Delays in 50 generations is shown below in graph:

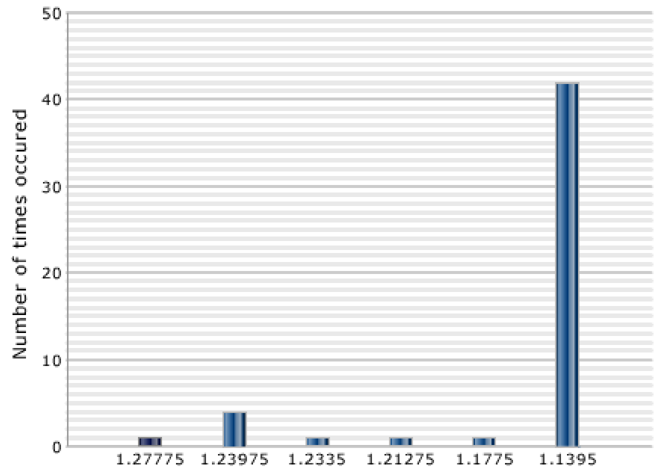

Figure 7.12. Distribution of Shortest Average Delays in 50 generations

Population 1	Population 2	Population 3	Population 4	Average	Count
1.277778	1.277778	1.291667	1.266667	1.2784725	1
1.277778	1.125	1.291667	1.266667	1.240278	4
1.277778	1.125	1.291667	1.266667	1.240278	
1.277778	1.125	1.291667	1.266667	1.240278	
1.277778	1.125	1.291667	1.266667	1.240278	
1.277778	1.125	1.266667	1.266667	1.234028	1
1.277778	1.125	1.266667	1.183333	1.2131945	1
1.277778	1.125	1.125	1.183333	1.17777775	1
1.125	1.125	1.125	1.183333	1.13958325	42

Table 7.3: Showing results taken of randomly generated population and their average

CHAPTER 8: FUTURE RESEARCH DIRECTIONS

This routing algorithm we are using M/M/1 queuing model [28] by using Genetic Algorithm Approach together for Network Path Optimization. This approach is used for handling network constraints of Bandwidth, Utilization and Delay for Path Optimization. If there is a clash in such constraints as well than select that path which has less hops count. Most constraint satisfaction measurements are on the basis of Resource Profiles. These profiles define availability or consumption of resource as a time function [29]. Constraints defined in this algorithm are Bandwidth Availability, Delay, Utilization and Hop count. The fitness function is as follows by using M/M/1 queuing model [28]:

$$T = \frac{1/\mu C}{1 - \rho}$$

T-- mean delay in second
1/μ -- mean packet size in bits
C -- capacity in bps
ρ-- utilization factor

Thus in this algorithm, all the constraints mentioned in the other two algorithms are combined. The path thus obtained will be better.

CHAPTER 9: CONCLUSION

The Intelligent analysis and network routing design is gaining importance and thus gives an edge in this fast era. In this research work, a variation of Genetic Algorithm (GA) is presented for finding the Optimized shortest path of the network. The algorithm finds the optimal path by using an objective function that is based on the bandwidth, delay and utilization constraints over the network. The main contribution of this work is the "2-point over 1-point crossover". Different mutation operation has been performed for different scenarios. We have tested our algorithm on different scenarios and detailed results have been shown and discussed. The results prove our assertion that our proposed algorithm provides better results with more efficiency as compared to previously proposed algorithms.

References

[1] Anton Riedl, “A Versatile Genetic Algorithm for Network Planning”, Institute of Communication Networks Munich University of Technology.

[2] Mitsuo Gen, Lin Lin , “A New Approach for Shortest Path Routing Problem by Random Key-based GA”, *Genetic And Evolutionary Computation Conference Proceedings of the 8th annual conference on Genetic and evolutionary computation*, 2006, pages 1411-1412.

[3] Darrell Whitley, “A genetic algorithm tutorial”, *Statistics and Computing*, 1992, volume 4, number 2, pages 65-85.

[4] Ramon Fabregat, Yezid Donoso, Benjamin Baran, Fernando Solano, Jose L. Marzo, “Multi-objective optimization scheme for multicast flows: a survey, a model and a MOEA solution”, *Proceedings of the 3rd international IFIP/ACM Latin American conference on Networking,* 2006, pages 73-86.

[5] M. Ericsson, M. Resende, P. Pardalos, “A Genetic Algorithm for the weight setting problem in OSPF routing”, *Journal of Combinatorial Optimization*, 2002, volume 6, pages 299-333.

[6] Norio. Shimamoto, Atsushi Hiramatsu, Kimiyoshi Yamasaki, “A dynamic Routing Control based on a Genetic Algorithm”, *IEEE International Conference on Neural Networks,* 1993, pages 1123-1128.

[7] Hong Pan, Irving Y. Wang , “The Bandwidth Allocation of ATM through GA”, *Global Telecommunications Conference, 1991. GLOBECOM '91*, 1991, pages 125-129.

[8] Basela S. Hasan, Mohammad A. Khamees, Ashraf S. Hasan Mahmoud, “A Heuristic Genetic Algorithm for the Single Source Shortest Path Problem”, *IEEE/ACS International Conference on Computer Systems and Applications*, 2007, pages 187-194.

[9] Anton Riedl, “A Hybrid Genetic Algorithm for Routing Optimization in IP Networks Utilizing Bandwidth and Delay Metrics”, Institute of Communication Networks, Munich University of Technology, Munich, Germany.

[10] Yinzhen Li, Ruichun He, Yaohuang Guo, “Faster Genetic Algorithm for Network Paths”, *Proceedings of The Sixth International Symposium on Operations Research & Its Applications (ISORA’06)* Xinjiang, China, August 8-12, 2006. Pages 380-389.

[11] A.W.W.NG and B.J.C. Perera, “Importance of Genetic Alrothim Operators in River Water Quality Model Parameter Optimisation”, school of the Built Environment, Victoris University of Technology, Melbourne.

[12] David E. Goldberg, “*Genetic Algorithms in Search, Optimization & Machine Learning*”, Addison-Wesley Longman Publishing Co., Inc. Boston, MA, USA, 1989.

[13] David M. Tate, Alice E. Smith, “Expected Allele Coverage and the Role of Mutation in Genetic Algorithms”, Proceedings of the 5th International Conference on Genetic Algorithms Pages: 31 - 37 Year of Publication: 1993, ISBN:1-55860-299-2

[14] Zhao-Xia Wang, Zeng-Qiang Chen and Zhu-Zhi Yuan, “QoS routing optimization strategy using genetic algorithm in optical fiber communication networks”, Journal of Computer Science and Technology, Volume 19 , Year of Publication: 2004, ISSN:1000-9000, Pages: 213 - 217

[15] Emmeche C., “Garden in the Machine. The Emerging Science of Artificial Life”, Princeton University Press, 1994, pp. 114.

[16] Goldberg D., “Genetic Algorithms”, Addison Wesley, 1988.

[17] G. Lauks, “Multi-objective optimization of Bandwidth Broker with Evolutionary Algorithm”, *Electronics and Electrical Engineering -Kaunas: Technologija*, 2005, volume 3, pages 59.

[18] Jose Maria A. Pangilinan, Gerrit K. Janssens, “Evolutionary Algorithms for the Multi-objective Shortest Path Problem”, *International Journal of Computer & Information Science & Engineering*, Volume 1, Number 1.

[19] D. G. Gomes, N. Agoulmine, J. N. de Souza., “IP Bandwidth Allocation Management using Agents and Neural Network Approach”, *2002 IEEE Workshop on IP Operations and Management*, 2002, pages 8-12.

[20] “GeneticAlgorithms”,http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html

[21] “GeneticAlgorithms”, http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html

[22] Jennifer Rexford, “Route Optimization in IP Networks”, Department of Computer Science, Princeton University, Princeton, NJ 08544, USA

[23] Dimitri P. Bertsekas, “Linear Network Optimization”, 1991 Massachusetts Institute of Technology.

[24] Bernard Fortz, Mikkel Thorup, “Internet Traffic Engineering by Optimizing OSPF Weights”

[25] Josep R. Medina and Vıctor Yepes, “Optimization of touristic distribution networks using genetic algorithms”, January-June 2003, 95-112

[26] Karla Hoffman, “Traveling Salesman Problem”, George Mason University, Manfred Padberg, New York University

[27] Lau Tung Leng, “Guided Genetic Algorithm”, PhD Thesis, Department of Computer Science, University of Essex Wivenhoe Park, Colchester, Essex CO4 3SQ, United Kingdom

[28] Yantai Shu, Fei Xue, Zhigang Jin and Oliver Yang, “ The Impact of Self-similar Traffic on Network Delay”, Dept. of Computer Science, Tianjin University, Tianjin 300072, P.R. China, School of Information Technology and Engineering, University of Ottawa,Ottawa, Ontario Canada KIN 6N5, 0-7803-43 14-X/98/$10.0081998 IEEE

[29] Matthew Bartschi Wall, “A Genetic Algorithm for Resource-Constrained Scheduling”, Massachusetts Institute of Technology.

[30] D. Van Veldhuizen, G. Lamont, “Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art”, c 2000 by the Massachusetts Institute of Technology, Evolutionary Computation Volume 8, Number 2

[31] Carlos A. Coello Coello, “A Short Tutorial on Evolutionary Multiobjective Optimization”

[32] Deb K., Multi-objective Objective Optimization using Evolutionary algorithms, Wiley and Sons Ltd, (2001).

[33] Johan Andersson, "Applications of a Multi-Objective Genetic Algorithm to Engineering Design Problems", Department of Mechanical Engineering, Linköping University 581 83 Linköping, Sweden

[34]Andersson J. and Wallace D., "Pareto optimization using the Struggle Genetic Crowding Algorithm, Engineering Optimization, vol. 34, No. 6, pp. 623-643, (2002).

[35] Carlos A_ Coello Coello, "Handling Preferences in Evolutionary in Multiobjective Optimization: A Survey", Laboratorio Nacional de Informatica Avanzada

[36] Abdullah Konak, David W. Coit, Alice E. Smith, "Multi-objective optimization using genetic algorithms: A tutorial"

[37] J.D. Schaffer, "Multiple Objective optimization with vector evaluated genetic algorithms", *International Conference on Genetic Algorithm and their applications,* 1985.

[38] http://www.bmva.ac.uk/bmvc/1997/papers/071/node3.html

[39] Vandana Venkat, Sheldon H. Jacobson and James A. Stori, "A Post-Optimality Analysis Algorithm for Multi-Objective Optimization", *Journal Computational Optimization and Applications, Volume 28, 2004.*

[40] http://www.calresco.org/lucas/pmo.htm

[41] Srinivas N, Deb K., "Multiobjective optimization using nondominated sorting in genetic algorithms", J Evol Comput 1994;2(3):221–48.

[42] Zitzler E, Deb K, Thiele L. Comparison of multiobjective evolutionary algorithms: empirical results. Evol Comput 2000;8(2):173–95.

[43] Morse JN, "Reducing the size of the nondominated set: pruning by clustering", *Comput Oper Res 1980;7(1–2):55–66.*

[44] Goldberg DE, Richardson J, "Genetic algorithms with sharing for multimodal function optimization", *Genetic algorithms and their applications, Pproceedings of the Second International Conference on Genetic Algorithms, 28–31 July, 1987. Cambridge, MA, USA: Lawrence Erlbaum Associates; 1987.*

[45] Jensen MT, Reducing the run-time complexity of multiobjective EAs: The NSGA-II and other algorithms. IEEE Trans Evolutionary Computation 2003;7(5):503–15.

[46] Fieldsend JE, Everson RM, Singh S. Using unconstrained elite archives for multiobjective optimization. IEEE Trans Evol Comput 2003;7(3):305–23.

[47] Mostaghim S, Teich J, Tyagi A. Comparison of data structures for storing Pareto-sets in MOEAs. In: Proceedings of the 2002 world congress on computational intelligence—WCCI'02, 12–17 May, 2002. Honolulu, HI, USA: IEEE; 2002.

[48] Coello CAC. A survey of constraint handling techniques used with evolutionary algorithms. Veracruz, Mexico: Laboratorio Nacional de Informtica Avanzada; 1999.

[49] Knowles JD, Corne DW. M-PAES: a memetic algorithm for multiobjective optimization. In: Proceedings of the 2000 congress on evolutionary computation, 16–19 July, 2000. La Jolla, CA, USA: IEEE; 2000.

[50] Jimenez F, Gomez-Skarmeta AF, Sanchez G, Deb K, "An evolutionary algorithm for constrained multi-objective optimization", in Proceedings of the 2002 world congress on computational intelligence—WCCI'02, 12–17 May, 2002. Honolulu, HI, USA: IEEE; 2002.