MULTICAST ROUTING ALGORITHMS FOR COMPUTER NETWORK

A THESIS

submitted in fulfillment of the

requirements for the award of the degree

of

Doctor of Philosophy

in

Computer Science & Engineering

by

Sushma Jain

(Roll No. 9020361)

COMPUTER SCIENCE & ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA–147004 PUNJAB–INDIA

November, 2012

CANDIDATE'S DECLARATION

I hereby certify that the work which is being presented in the thesis entitled **MULTICAST ROUTING ALGORITHMS FOR COMPUTER NETWORK** in the fulfilment for the award of the Degree of **Doctor of Philosophy** and submitted in the **Computer Science & Engineering Department** of the **THAPAR UNIVERSITY** is an authentic record of my own work carried out under the supervision of **Prof. J.D. Sharma**.

The matter presented in this thesis has not been submitted by me for the award of any other degree of this or any other University.

Sushma Jain

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Dr. J.D. Sharma
Professor,
I.I.T. Roorkee

Date : 04th June 2012

The Ph.D. Viva-Voce examination of Ms. Sushma Jain has been held on 03rd Nov 2012

Supervisor **Member** **Member** 3.11.12 **Head, CSED** **External Examiner** 3/11/12

Dean (RSP)

ABSTRACT

The computer networks are also being used to send information in the form of data streams or packets to a selective, usually large number of users or destinations. These networks are experiencing explosive growth due to the advances in optical fiber and switch technologies. These networks are being used for various distributed multimedia applications such as audio/video-conferencing, remote education, E-commerce, software delivery, games, telephony, virtual whiteboard, etc. The applications such as video/audio conferencing are delay-sensitive whereas the applications such as interactive gaming are bandwidth-intensive. The factors, such as explosive growth of computer networks, users and variety of resource intensive networked applications, require efficient routing algorithms for optimizing the utilization of network resources.

The **multicast** is a technique used to facilitate the information exchange by routing data from one or more sources to a potentially large number of destinations such that overall utilization of resources is minimized in some sense. For efficient multicasting, a tree is constructed, where due to the presence of smart switches; data traverses the link only once.

The resource intensive applications have a wide diversity of Quality of Service (QoS) requirements for data transfer. The commonly used QoS measures are cost, bandwidth, end-to-end delay, delay jitter, packet loss ratio and hop count, etc. The QoS based routing aims at transferring information simultaneously from one or more sources (or senders) to a group of destinations (or receivers), namely the multicast group, by optimizing objective(s) while satisfying a set of QoS constraints. A solution to the QoS multicast routing problem is to build a multicast tree, Steiner tree, which spans from the source node to all destinations. The diverse QoS requirements make the routing problem intractable and NP-complete. The computation complexity increases in searching a tree for certain combinations of QoS requirements.

The QoS multicast routing problem like various real world optimization problems involves different criteria and therefore can be formulated as multiobjective optimization. The conflicting criteria such as minimizing the tree cost and maximizing the residual bandwidth of the tree subjected to end-to-end delay are to be confronted in solving the multiobjective multicast routing and a solution representing a good compromise between several criteria be obtained.

The Hopfield neural network (HNN) and population based search and optimization techniques are considered for the multicast routing because of their ability to provide the solution to practical and complicated optimization problems. Using HNN based approach, the objective function is expressed as quadratic energy function and the associated weights between neurons are computed using the gradient descent of energy function.

The multicast routing problems involve discontinuities and disjointed feasible spaces. These problems can be adequately handled by the population based search and optimization algorithms because these methods obtain the optimal solution by improving the solution with the progress of iterations and search for good solutions in parallel. The capability of these algorithms such as the capability of generating multiple promising solutions in a single run and evolving a population of solutions towards the Pareto front make them especially adequate to deal with Pareto-based multiobjective multicast routing.

The work reported in this thesis is carried out to develop the algorithms for constructing the multicast tree that optimizes the computer network resources on the basis of both single objective optimization and multiobjective optimization by satisfying QoS constraints using Hopfield neural network and population based search and optimization algorithms. The specific objectives are –

- Investigations on multicast routing using HNN for unconstrained optimization, constrained optimization and multiobjective optimization.
- Investigations on QoS constrained multicast routing using population based search and optimization algorithms.
- Investigations on QoS constrained multiobjective multicast routing using population based search and optimization algorithms.

The multicast routing using Hopfield neural network (HNN) is investigated for unconstrained routing, delay-constrained routing, and delay-constrained multiobjective routing. The multicast tree is obtained for various formulations of optimization problems such as cost optimization, residual bandwidth optimization and cost-residual bandwidth optimization without and with delay constraints. The *shortest path tree* (SPT) and *multicast tree* (MT) formulations have been investigated to construct the tree between source and various destinations. The respective optimization problems are mapped into the dynamic Hopfield model to obtain associated energy function. The minimization of energy function is attained with the progress of the iterations and consequently the optimal multicast tree is obtained at the convergence. The effectiveness of the developed algorithms is tested on an undirected weighted network to obtain optimum multicast trees for different sets of source and destinations. The cost-residual bandwidth optimization, a multiobjective optimization formulation, is attempted through weighted formulation of the objectives in energy term. The optimal Pareto front is constructed after running the simulation for different combinations of weights and the compromised solution is obtained using the fuzzy-cardinal ranking.

The investigations on QoS constrained multicast routing are carried out using population based stochastic search and optimization algorithms. Two algorithms namely *tree-structured genetic algorithm* (TSGA) and *tree-structured particle swarm optimization* (TSPSO) are proposed to obtain optimal multicast tree. The novel tree structured representation employing topological features is proposed for construction of multicast tree. The solution is represented as ordered *M*-array structure where each array is representing the random path from source to a destination. The order of M arrays is the order of destination nodes in multicast group. The multicast tree is constructed by visiting each node of *M*-array solution in one-by-one manner. The various operations in the TSGA and TSPSO are performed while preserving the tree-structured representation of the solution. For the purpose, *tree-crossover*, *tree-mutation* and *tree-merge* operations are formulated, which are resulting into loop free multicast tree. The tree-mutation is also used in TSPSO to avoid sticking of the algorithm at local optimum. The effectiveness of TSGA and TSPSO is studied for obtaining optimal multicast tree for delay constraint minimum cost multicast routing, delay constraint optimum residual-bandwidth multicast routing, and delay and delay jitter bound multicast routing for various random networks. The random networks are generated using BRITE network simulator which is working on Waxman

model. Their performance is tested for different sizes of multicast group, different sizes of networks, varying delay and delay-jitter bounds.

To investigate the QoS constrained multiobjective multicast routing, two algorithms namely *tree-structured multiobjective genetic algorithm* (TSMGA) and *tree-structured multiobjective particle swarm optimization* (TSMPSO) are proposed. These algorithms are the population based search and optimization algorithms based on the evolutionary principle**.** The development of TSMGA is inspired from the elitist non-dominated sorting genetic algorithm. The crowded non-dominating sorting is used to preserve elitism. The TSMPSO method is inspired to return set of Pareto optimal solutions. Both TSMGA and TSMPSO methods employ topological assisted tree structured representation of the solution. To avoid premature convergence, the mutation is used stochastically. After obtaining the Pareto-optimal front, the best compromised solution is obtained using fuzzy cardinal priority ranking. The multi-objective multicast routing is attempted for two formulations namely cost-residual bandwidth optimization and cost-residual bandwidth-packet loss probability optimization. In both these multi-objective optimization, the end-to-end delay and delay jitter are represented as constraints. The effectiveness of the developed TSMGA and TSMPSO algorithms is tested and compared for different sizes of multicast group and varying size of various networks including the random networks formed using network topology generator BRITE.

ACKNOWLEDGEMENTS

My foremost and profound gratitude goes to my guide **Dr. J.D. Sharma**, Professor, Electrical Engineering Department, Indian Institute of Technology, Roorkee for his proficient guidance and continuous monitoring, motivation, helping nature and useful criticism throughout the research work leading to this Thesis in the present form. I have a deep sense of admiration for his goodness and inexhaustible enthusiasm.

It is my proud privilege of being the student of Prof. J.D. Sharma. His wide experience, prolific research knowledge, innovative and visionary approach are very much inspiring to me. His continuous encouragement is always a force to recon and to introspect. I have profound regards for him and express life time heart felt thanks to Prof. Sharma.

I am thankful to Dr. Seema Bawa, Professor, Computer Science & Engineering Department and Dr. R.K. Sharma, Professor, School of Mathematics & Computer Applications for being the members of Doctoral Committee and spending their valuable time in reviewing and critically examining the work.

I am thankful to present Chairman of Doctoral Committee Dr. Maninder Singh, Associate Professor and Head, Computer Science & Engineering Department and Dr. Inderveer Chana, Associate Professor & PG Coordinator for the much needed support throughout the work.

My heart felt gratitude is due to Dr. P.K. Bajpai, Distinguished Professor and Dean, Research & Sponsored Projects, Dr. Sushil Mittal, Sr. Professor, School of Chemistry & Biochemistry, Dr. K.K. Raina, Deputy Director and Honorable Director Dr. A. Mukherjee for the encouragement, support and providing the necessary facilities to carry out and complete this work on steady course.

I wish to express my deep sense of gratitude to all the faculty and staff members of Computer Science & Engineering Department and all persons who with their encouraging, caring words, constructive criticism and suggestions or by smile have contributed directly or indirectly in a significant way towards completion of this mammoth project.

My sincere and special thanks are due to my soul mate Dr. Sanjay K. Jain for emotional support and strength at various stages of the work. His useful criticism, involvement with my work, painstaking and critical reading of the text, helped me tremendously in improving upon the thesis.

Special, sincere, heartfelt gratitude goes to my family members, particularly my father-in-law and my children, whose sincere prayers, best wishes, support and encouragement has been a constant source of assurance, guidance, strength and inspiration to me during this research work. I am proud to humbly dedicate this research work to my children *Drishti* and *Akshat*.

(Sushma Jain)

TABLE OF CONTENTS

	Candidates Declaration			
	Abstract			***i***
	Acknowledgements			***v***
	Table of Contents			***vii***
	List of Figures			***xi***
	List of Tables			***xv***
1.	**Introduction**			***1***
	1.1	Overview		***1***
	1.2	Computer Communication Methods		***3***
	1.3	State of Art		***5***
	1.4	Objectives of the Research Work		***7***
	1.5	Organization of Thesis		***8***
	1.6	Concluding Remarks		***10***
2.	**Literature Review**			***11***
	2.1	General		***11***
	2.2	Related work		***12***
	2.3	Author's Contribution		***19***
	2.4	Concluding Remarks		***22***
3.	**Multicast Routing Problems**			***23***
	3.1	General		***23***
	3.2	Concept of Multiobjective Optimization		***24***
		3.2.1	Domination and Pareto Optimality	***26***
		3.2.2	Best Compromised Solution	***27***

	3.3	Network Model		*28*
		3.3.1	Path Metrics	*28*
		3.3.2	Tree Metrics	*29*
	3.4	Multicast Routing Optimization Problems		*31*
		3.4.1	Multicast Routing	*32*
			3.4.1.1 Minimum Cost Tree	*32*
			3.4.1.2 Optimum Residual Bandwidth Tree	*33*
		3.4.2	Multiobjective Multicast Routing	*34*
			3.4.2.1 Optimum Cost-Residual Bandwidth Tree	*34*
			3.4.2.2 Optimum Cost-Residual Bandwidth-Packet Loss Tree	*35*
	3.5	Concluding Remarks		*36*
4.	**Multicast Routing Using Hopfield Neural Network**			*37*
	4.1	General		*37*
	4.2	Hopfield Neural Network Model		*38*
	4.3	Mapping Routing Tree under HNN		*41*
		4.3.1	Forming Shortest Path Tree	*43*
			4.3.1.1 Minimum Cost Shortest Path Tree	*43*
			4.3.1.2 Optimum Residual Bandwidth Shortest Path Tree	*49*
			4.3.1.3 Shortest Path Tree Algorithm	*50*
		4.3.2	Forming Multicast Tree	*52*
			4.3.2.1 Multicast Tree Algorithm	*55*
		4.3.3	Forming Multiobjective Multicast Tree	*55*
			4.3.3.1 Multiobjective Multicast Tree Algorithm	*58*
	4.4	Results and Discussion		*60*
		4.4.1	Shortest Path Tree Formulation	*61*
		4.4.2	Multicast Tree Formulation	*63*
		4.4.3	Multiobjective Multicast Tree Formulation	*68*

	4.5	Concluding Remarks		*72*
5.	**Multicast Routing using Population Based Optimization Algorithms**			*75*
	5.1	General		*75*
	5.2	Tree Structured Encoding Scheme		*76*
		5.2.1	Forming Solution as M-array Structure	*77*
		5.2.2	Forming Multicast Tree	*78*
	5.3	Tree-Structured Genetic Algorithm		*80*
	5.4	Tree-Structured Particle Swarm Optimization		*85*
	5.5	Evaluation Functions for Routing Problems		*88*
		5.5.1	Function for QoS Constrained Minimum Cost Tree	*88*
		5.5.2	Function for QoS Constrained Optimum Residual Bandwidth Tree	*89*
	5.6	Results and Discussion		*90*
		5.6.1	Cost minimization	*91*
		5.6.2	Residual Bandwidth Optimization	*98*
	5,7	Concluding Remarks		*103*
6.	**Multiobjective Multicast Routing using Population Based Optimization Algorithms**			*105*
	6.1	General		*105*
	6.2	Tree-Structured Multiobjective Genetic Algorithm		*107*
		6.2.1	Tree-Structured Genetic Algorithm Steps	*112*
	6.3	Tree-Structured Multiobjective Particle Swarm Optimization		*114*
	6.4	Multiobjective Multicast Routing Problems		*117*
	6.5	Results and Discussion		*119*
		6.5.1	Cost-Residual Bandwidth Optimization	*119*
		6.5.2	Cost-Residual Bandwidth-Packet Loss Probability Optimization	*126*
	6.6	Concluding Remarks		*131*

7.	**Conclusions and Scope for Further Work**		***133***
	7.1	General	***133***
	7.2	Summary of Important Findings	***133***
	7.3	Scope for Further Work	***137***
	List of Publications		***139***
	References		***141***
	Appendix-A		***151***
	Appendix-B		***159***

LIST OF FIGURES

Fig. No.	*Figure Title*	*Page No.*
3.1	*Various objectives and optimal solution*	*24*
3.2	*Illustration of multiobjective optimization procedure*	*25*
3.3	*Illustration of non-dominance and Pareto front for min-min optimization*	*26*
3.4	*Illustration of Pareto fronts for various two-objective optimization problems*	*27*
4.1	*(a) Representation of Hopfield neural network by electrical equivalent circuit* *(b) Circuit representation of a neuron*	*39*
4.2	*An example graph*	*48*
4.3	*The flowchart for computing delay-constrained shortest path tree*	*51*
4.4	*The flowchart for computing delay-constrained multicast tree*	*56*
4.5	*The flowchart for computing best compromise delay-constrained multicast tree*	*59*
4.6	*The resulted trees for cost minimization using shortest path tree formulation*	*62*
4.7	*The resulted trees for residual-bandwidth maximization using shortest path tree formulation*	*62*
4.8	*The resulted trees for cost minimization using multicast tree formulation*	*63*
4.9	*The variation in energy and neural network dynamics for delay-bound cost optimization*	*64*
4.10	*Multicast trees for delay-constrained minimum cost optimization*	*65*
4.11	*The resulted trees for residual-bandwidth maximization using multicast tree formulation*	*66*

4.12	The variation in energy and neural network dynamics for delay-constrained residual bandwidth optimization	67
4.13	Multicast trees for different multicast groups for optimum residual bandwidth	68
4.14	Effect of varying HNN coefficients (μ_1=μ_7) on cost and residual bandwidth	69
4.15	Effect of varying μ_7 on cost and residual bandwidth	70
4.16	Effect of varying μ_1 on cost and residual bandwidth	70
4.17	Pareto-optimal fronts for multiobjective cost-residual bandwidth optimization	71
4.18	Best compromised multicast trees for optimum cost-residual bandwidth	72
4.19	Pareto-optimal front and best compromised tree for cost-residual bandwidth optimization for multicast group {7,10,15,19,23}	72
5.1	Flowchart to generate M-array structure of solution	78
5.2	Flowchart representing the steps to obtain multicast tree	79
5.3	An eight node example network topology	80
5.4	An illustration of solutions and tree matrix and multicast tree	81
5.5	Realizing tree-crossover operation in the proposed scheme	83
5.6	Realizing tree-mutation operation in the proposed scheme	83
5.7	The illustration of TSGA algorithm for multicast routing	84
5.8	Tree-merge operation to obtain new position of the particle	86
5.9	The illustration of TSPSO algorithm for multicast routing	87
5.10	Representation of optimal solution and multicast tree for minimum cost routing	92
5.11	Comparative performance of TSGA and TSPSO for different sizes of multicast groups on 24-node network	94
5.12	Effect of varying multicast group on 100-node random networks for delay-constrained minimum cost for varying delay bound Δ	96

5.13	*Effect of varying multicast group size on 100-node random networks for delay and delay jitter bound minimum cost routing for varying Δ and δ*	*97*
5.14	*The performance of TSGA and TSPSO for handling constrained minimum cost routing*	*97*
5.15	*Minimum cost routing for varying network size*	*98*
5.16	*Representation of multicast tree for residual bandwidth optimization*	*99*
5.17	*Effect of varying multicast group on 100-node random networks for delay-constrained optimum bandwidth for varying delay bound Δ*	*101*
5.18	*Effect of varying multicast group size on 100-node random networks for delay and delay jitter bound optimal residual bandwidth routing for varying Δ and δ*	*102*
5.19	*The performance of TSGA and TSPSO for handling constrained optimum residual bandwidth routing*	*103*
6.1	*Process to create Pareto fronts through non-dominating sorting*	*109*
6.2	*Population and fronts in objective space for min-min type problem*	*109*
6.3	*Crowding distance for a solution*	*110*
6.4	*Schematic representation to maintain elitism and population selection*	*111*
6.5	*Flowchart for tree-structured multiobjective genetic algorithm*	*113*
6.6	*Flowchart for tree-structured multiobjective particle swarm optimization*	*116*
6.7	*A 15-node Bellcore network topology*	*119*
6.8	*Initial Pareto fronts for 15-node Bellcore network*	*120*
6.9	*Pareto optimal front for 15-node Bellcore network*	*121*
6.10	*Representation of optimal compromised solution and multicast tree*	*121*
6.11	*The Pareto-optimal front for multicast group {7,10,15}*	*122*
6.12	*The Pareto-optimal front for multicast group {7,10,15,19,23}*	*122*
6.13	*The optimal multicast trees for 24-node USIP Backbone Network*	*122*
6.14	*Pareto optimal front and initial Pareto "front 1" through TSMGA*	*123*

6.15	*Pareto-optimal front using TSMPSO for 100-node network*	*123*
6.16	*Pareto optimal fronts from TSMGA and TSMPSO for 100-node network*	*124*
6.17	*Effect of multicast group on the convergence time on 100-node network*	*125*
6.18	*The effect of network size on convergence time*	*126*
6.19	*Pareto-optimal solutions in objective space for 24-node network*	*128*
6.20	*Illustration of Pareto-optimal solutions for Fig. 6.19(b) as cost-residual bandwidth characteristics for different packet-loss probability*	*128*
6.21	*Pareto-optimal solutions in objective space for 100-node network*	*130*
6.22	*Illustration of Pareto-optimal solutions for Fig. 6.21 as cost-residual bandwidth characteristics for different packet-loss probability*	*130*
A1	*Representation of 24-node USIP Backbone Network*	*151*

LIST OF TABLES

Table No.	*Table Title*	*Page No.*
4.1	*Neuron output matrix V for the example graph (Fig. 4.2)*	*48*
4.2	*Summary of delay-constrained minimum cost optimization using multicast tree formulation for different multicast groups (source s={0}, delay bound* $\Delta = 20$*)*	*65*
4.3	*Summary of residual bandwidth optimization using multicast tree formulation for different multicast groups*	*67*
4.4	*The optimal cost and residual bandwidths for equal values of* μ_1 *&* μ_7	*69*
4.5	*Cost and residual bandwidths for different values of* μ_7 *and fixed* μ_1*=300*	*69*
4.6	*Cost and residual bandwidths for different valued of* μ_1 *and fixed* μ_7*=300*	*70*
5.1	*Effect of the population size on the delay and delay-jitter bound optimum cost and convergence time for 24-node USIP Backbone network*	*91*
5.2	*Summary of minimum cost multicast tree formulation using TSGA and TSPSO*	*91*
5.3	*Effect of delay bound* Δ *on delay-bound minimum cost routing (source s = {8}, Multicast group = {1, 4, 19, 23})*	*93*
5.4	*Effect of delay jitter bound* δ *on delay and delay-jitter bound minimum cost routing (source s = {8}, Multicast group = {1, 4, 19, 23}, delay bound* Δ*=25)*	*93*
5.5	*Summary of delay and delay-jitter bound minimum cost for different multicast groups (delay bound* $\Delta = 20$*, delay jitter bound* $\delta = 10$*)*	*93*
5.6	*The structure of optimal solution and tree for different multicast groups*	*95*

5.7	Effect of delay bound for varying size of multicast group on 100 node random network	96
5.8	Effect of delay and delay jitter bounds for varying size of multicast group on 100 node random network	97
5.9	TSGA and TSPSO performance for minimum cost routing for different networks (multicast group size in 10% network size)	98
5.10	Summary of optimum residual bandwidth multicast tree formulation using TSGA and TSPSO	99
5.11	Effect of delay bound Δ on delay-bound optimum residual bandwidth routing (source s = {8}, Multicast group = {1, 4, 19, 23})	100
5.12	Effect of delay jitter bound δ on delay and delay-jitter bound optimum residual bandwidth routing (source s={8}, Multicast group={1,4,19,23}, delay bound Δ=25)	100
5.13	Effect of delay bound on delay bound optimum residual bandwidth routing for varying size of multicast group on 100 node random network	101
5.14	Effect of delay and delay jitter bounds for varying size of multicast group on 100 node random network	102
6.1	Cost and available bandwidth for Initial Pareto fronts for 15-node Bellcore network	120
6.2	Summary of Pareto optimal fronts from TSMGA and TSMPSO for 100-node	124
6.3	Summary of convergence time for different sizes of multicast group	125
6.4	Summary of convergence time for different sizes of random networks	126
6.5	The set of Pareto-optimal solutions for 24-node network	127
6.6	The set of Pareto-optimal solutions for 100-node network	129
A1	Link data for 24-Node USIP Backbone Network	152
A2	Link data for 100-Node Random Network	153

INTRODUCTION

1.1 OVERVIEW

A computer network is an interconnection of various computer systems located at different places. The computer providing resources to other computers on a network is known as server whereas the computers accessing the shared network resources are known as workstations or nodes. Computer Networks may be classified as local area network (LAN) or wide area network (WAN). The LAN links computers within a limited geographical area. The client/server model is most popular for the LAN. The WAN links the computers to large geographical area spanning even a country or continent through telephone lines, microwaves and satellite transmission media. The computers on WAN, referred as hosts, are connected to subnet which include transmission facility, switching equipments and routers. With routers, LANs can be connected to WAN. The Internet is a network of networks where millions of computers all over the world are connected and the computer users on the Internet can contact one another. The Internet is a huge resource of accessible information to the peoples and thereby it is now indispensable.

The computer networks are also being used to send information in the form of data streams or packets to a selective, usually large number of users or destinations. The users are increasing continuously and the computers are being used nowadays into more routine areas of life while providing better interface to information. This explosive growth of computer networks and the dependency on computers are attributed to the advances in audio, video, data storage, optical fibre and switch technologies. These technologies help in providing more effective communications and therefore the computer networks are being used for various distributed multimedia applications such as audio/video-conferencing, remote education, E-commerce, software delivery, games, telephony, virtual whiteboard, etc. The applications such as video/audio conferencing are delay-sensitive and require that data streams reach all participants without excessive delay. The applications such as interactive gaming are bandwidth-intensive and require that the link congestion should be averted to avoid delay or failure of information transmission. Therefore, the factors, such as explosive growth of computer networks, users and variety of resource intensive networked applications, require efficient routing algorithms for optimizing the utilization of network resources.

The **multicast** is a technique used to facilitate the information exchange or data transfer by routing data from one or more sources to a potentially large number of destinations such that overall utilization of resources is minimized. The process of routing consists of routing algorithms and routing protocols. The routing protocols are responsible for information exchange and routing dynamics. For efficient multicasting, routing algorithm constructs a tree. Assuming the presence of smart switches, data transfer takes place in parallel to the destinations, and data traverses the link only once. Therefore, the multicast routing can utilize the network resources more efficiently. The multicasting is effective when large information is transmitted to selective or subsets of users. The multicasting can be extended to various applications that involve sending copies of data to multiple destinations.

The multicast applications can be classified as interactive, less interactive and non-interactive. Interactive applications include videoconferencing, computer-supported cooperative work, and virtual whiteboard etc. The applications like remote education require a lesser amount of interaction while applications like mailing lists are non-interactive. The interactive multimedia applications are becoming increasingly important as the networks are capable of handling media traffic to end users.

For real time applications, various requirements that are regarded as Quality of Service parameters must be guaranteed. The QoS is the result of joining two terms quality and service and can be defined as a measurement of how well the network behaves. It also defines the characteristics and properties of specific services. In networking, the quality can be described as the process of delivering data in a reliable or at least someway better than the normal manner. These QoS requirements include end-to-end delay bound, delay-jitter bound, minimum bandwidth, packet loss probability etc.

The notion of QoS in networks means the data delivery service should satisfy certain performance requirements. These requirements define the QoS guarantees that a network should provide. The QoS routing in computer networks is defined as the process of transferring information from a source to a destination (or a group of destinations) through network elements under certain constraints or performance metrics. It means that QoS constrained network routing must be able to utilize the network resources efficiently while providing the requested QoS requirements. The diverse QoS requirements make the routing problem intractable and NP-complete. It involves high computation complexity in forming a tree based on certain combinations of QoS requirements.

The QoS based multicast routing in computer communication networks is an important technique to support data transmission in computer networks. It aims at transferring information simultaneously from one or more sources (or senders) to a group of destinations (or receivers), namely the multicast group, by optimizing objective(s) while satisfying a set of QoS constraints. The QoS multicast routing algorithm is aimed to build a multicast tree, which spans from the source node to all destinations. Like various real world optimization problems involving different criteria, the QoS multicast routing can also be formulated as multiobjective optimization. For example, the multicast route is to be identified by minimizing the cost and maximizing the residual bandwidth subjected to end-to-end delay as constraint. These possibly conflicting criteria are to be confronted in solving the multiobjective optimization problems and an appropriate solution representing a good compromise between several criteria be obtained.

1.2 COMPUTER COMMUNICATION METHODS

There are three methods of communication through a computer network. These are unicast, broadcast and multicast.

Unicast is the term used to describe communication where a piece of information is sent from a single source to a specified destination. It is predominant form of communication on LANs and within the Internet. The LANs and IP networks support the unicast communication. As the communication is from a single host to another single host, there is one device transmitting a message destined for one receiver.

In unicast, a separate copy of the data is send from the source to each client that requests it for example software download. Typical unicast applications are http, smtp, ftp and telnet etc. In unicast, there is a likelihood that paths to different destinations may share common links and the same data would be traversing a link multiple times. Therefore, the data transfer to the destinations is serialized which could lead to delays, even if the resources involved for each destination are independent and can be used simultaneously.

Broadcast is the term used to describe communication where a single piece of information is sent from one source to all clients or destinations in the network. In this case there is just one sender, but the information is sent to all connected receivers. The radio and television are widely used examples of broadcasting. The broadcast transmission is supported on most LANs and network layer protocols. The broadcast could reach all hosts on the subnet, all subnets, or all hosts on all subnets. The modern routers are capable to block IP broadcast traffic and restrict it to the local subnet.

Consider video server sending out networked TV channels. The simultaneous delivery of high quality video to each of a large number of clients will exhaust the capability of even a high bandwidth network. Thus, the broadcast communication is resulting into a stability issue for applications requiring high bandwidth. Broadcast also needlessly slows the performance of client machines as each client must process the broadcast data. The unicast also wastes bandwidth by sending multiple copies of data. To significantly ease scaling to larger groups of clients, multicast networking can be employed.

The **multicast** involves concurrently sending the same information to a group of destinations such that exactly one copy of the packet traverses a link. The multicast is a type of communication in which there may be more than one senders and the information sent is meant for a set of receivers. The multicast group computers must be a part of multicast IP based network. Multicast routers communicate among themselves using routing protocols and deliver the multicast datagram from the sender to the receivers. The router on receiving the datagram looks up its routing table and forwards it to the appropriate outgoing interface. When a host decides to join a particular multicast group, it sends the request to the local multicast router. The local multicast router makes an entry for this group and propagates the information to other multicast routers to establish the multicast routes.

For efficient multicasting, a tree is constructed. Assuming the presence of smart switches, data transfer takes place in parallel to the destinations and a link carries data over it only once. Therefore, the tree avoids the deficiencies resulted due to unicast and broadcast. When the multicast group is a subset of all possible recipients, multicasting is advantageous because it conserves network bandwidth and thereby facilitates more efficient use of the network infrastructural resources. The data transmission can be restricted to only the paying subscribers for services such as video on demand and the sensitive information is disseminated to only a select group of recipients.

1.3 STATE OF ART

The multicast is the ability of the communication network to accept a single message from an application and deliver its copies for multiple recipients. An efficient implementation of multicast permits the better usage of resources and harness the power of communication networks such as LAN, WAN and Internet.

Deering and Cheriton (1990) introduced the concept of multicast in late '80s. The concept of multicast was tested by Internet Engineering Task Force (IETF) in 1992 for audio-cast (Casner and Deering, 1992). Thereafter a lot of research has taken place in the area of multicast routing algorithms, routing protocols and real time applications. The routing process consists of routing algorithms (static) and routing protocols (dynamic) (Waxman, 1988). The routing protocols are responsible for information exchange and routing dynamics.

The communication network consisting of a set of nodes and a set of links can be modelled as undirected or directed graphs. In undirected graphs, the existence of a link is important and not the direction. The process to select unicast route has been treated as shortest path problem and minimum weight path is selected between a pair of nodes wishes to communicate (Deo and Pang, 1984). In multicast communication, the minimum weight tree spanning all the nodes in the multicast group is constructed. The polynomial-time algorithms are available for shortest path (Stalling, 1998) and spanning tree (Gallager *et al.*, 1983). The multicast communication can be source-specific or group shared and accordingly the multicast tree is source-based tree or group-shared tree. The source-specific tree is also regarded as Steiner tree. The Steiner tree was originally conceived as a problem of geometry and derives its name from Jacob Steiner (Gilbert and Pollak, 1968). The minimum Steiner tree problem aims to minimize the total cost of the multicast tree and is NP-complete (Hwang and Richards, 1992).

The QoS in the communication system is closely related to the performance of the routing system. The QoS can be defined as the collective effect of service performance which determines the degree of satisfaction of service user (Masip-Bruin *et al.*, 2006). The concept of QoS was introduced in late '80s with the introduction of ATM. The concept of QoS has been introduced in the internet by IETF contributions namely Intserv, Diffserv, RSVP and MPLS. The QoS routing algorithms solve the multi-constrained optimal path/tree problem. With the consideration of QoS, the problem to construct path/tree becomes NP-complete and intractable (Garey and Johnson, 1979). The QoS Steiner tree problems are mainly solved by heuristics (Kou *et al.*, 1981; Kompella *et al.,* 1993; Parsa and Zhu, 1998). These heuristics include delay constrained tree selection (Kompella *et al.*, 1993), bandwidth constrained tree selection (Hong *et al.*, 2003) etc.

The effectiveness of population based search and optimization iterative methods in solving the combinatorial optimization problems has attracted the researchers for their applicability for QoS routing (Ahn and Ramakrishna, 2002; Ravikumar and Bajpai, 1998; Randaccio and Atzori, 2007; Mohemmed *et al.*, 2008). These method search many individuals in parallel and improve the solution with the progress of iterations. Among these methods, the genetic algorithm developed by Holland (1975) and the particle swarm optimization (PSO) developed by Kennedy and Eberhart (2001) are very popular.

Hopfield and Tank (1985) first used the Hopfield neural network to solve travelling salesman optimization problem. The HNN works on the minimization of quadratic energy function and compute the associated weights between neurons using the gradient descent of energy function. The Hopfield neural network has been used for constructing shortest path and multicast tree (Ali and Kamoun, 1993; Pornavalai *et al.*, 1995).

The optimization of resources and guaranteeing the QoS requirements are two conflicting interests. Guaranteeing high-level or rigid QoS will require large resources. Even the confliction exists when different objectives are optimized simultaneously. With this confliction, an optimal solution that is better then all other solutions can not be obtained. Pareto, in 1906, introduced the concept of non-inferior solution, however, its application to various engineering problems was started around 1970 (Stadler, 1979). There are various evolutionary techniques to solve multiobjective optimization problems (Deb *et al.,* 2002; Coello *et al.,* 2004). These methods have been used for various engineering problems such as flow control, job shop scheduling (Li *et al.*, 2008; Ishibuchi and Murata, 1998). There exists few papers on multiobjective multicast routing (Roy and Das, 2004; Pinto *et al.*, 2005).

The dynamic multicasting, where a multicast group node may join or leave the multicast session, has been attempted through multicast tree rearrangement (Kun *et al.*, 2006). The task of efficiently forwarding/replicating the data is handled by routing protocols. There exists several protocols such as distance vector multicast routing protocol (DVMRP) (Waitzman *et al.*, 1988), Multicast extension of open shortest path first protocol (MOSPF) (Moy, 1994), Protocol independent multicast dense mode (PIM-DM) (Deering *et al.*, 1996) to support source based tree structure. There are several protocols such as Core based tree (CBT) (Ballardie *et al.*, 1993), PIM-sparse mode (PIM-SM) (Deering *et al.*, 1996) to support core based tree.

1.4 OBJECTIVES OF THE RESEARCH WORK

From the reported literature review, it is observed that the work on multicast routing algorithms is diversified. The QoS multicast routing itself is a complex combinatorial optimization problem, the interdependency and confliction among multiple QoS parameters makes the problem more difficult when QoS multiobjective multicast routing is attempted. The Hopfield neural network (HNN) and population based search and optimization algorithms have the potential to solve complicated problems.

After the literature review, the scope is identified to investigate the usage of HNN for multiobjective routing; develop an encoding scheme for forming multicast tree that can be used with the population based search and optimization algorithms and investigate their effectiveness for single and multiobjective multicast routing.

The research work entitled **"MULTICAST ROUTING ALGORITHMS FOR COMPUTER NETWORK"** is carried out with the following objectives:

- Investigations on multicast routing using HNN for unconstrained optimization, constrained optimization and multiobjective optimization.
- Investigations on QoS multicast routing using population based search and optimization algorithms.

- Investigations on QoS constrained multicast multiobjective routing using population based search and optimization algorithms.

1.5 ORGANIZATION OF THESIS

The specific contributions by the author to achieve the above mentioned objectives are summarized into seven chapters. The brief description of these chapters is outlined as –

The ***Chapter–I*** introduces the computer communication and details on the state-of-art on the area of the research work. This chapter also enlists the objectives of the present study and outlines the organization of thesis.

The ***Chapter-II*** provides the brief literature review on various aspects of QoS multicast routing and routing algorithms. The author's contributions to identified research objectives are also described in this chapter.

The ***Chapter-III*** introduces the concept of non-dominance, Pareto-front and best compromised solution. This chapter provides detailed description of the studied QoS routing problems. The network model and various path/tree metrics are discussed. The mathematical formulations of QoS multicast routing optimizations namely minimum cost multicast routing, optimum residual bandwidth multicast routing are presented under delay and delay-jitter bounds. Multiobjective multicast routing problem formulations for cost-residual bandwidth optimization and cost-residual bandwidth-packet loss probability optimization are also presented.

The ***Chapter-IV*** presents a method based on Hopfield neural network (HNN) to investigate unconstrained multicast routing, delay constrained multicast routing and delay constrained multiobjective multicast routing problems. The dynamic model of HNN is briefly reviewed and routing problems are mapped in the HNN model. The performance has been studied for minimum cost and optimum residual bandwidth, delay-constrained minimum cost, delay-constrained residual bandwidth, and optimum cost-residual bandwidth, delay-constrained optimum cost-residual bandwidth multicast routing. The strategy is presented to obtain the compromised optimal tree for optimum cost-residual bandwidth and delay-constrained optimum cost-residual bandwidth multicast routing.

The formulation has been attempted to obtain shortest path tree (SPT) and multicast tree (MT). The SPT is obtained by recursively obtaining shortest paths from source to various destinations and combining them to ensure that a link is participating only once in the multicast tree. The multiobjective optimization is represented as weighted sum objective. The Pareto-optimal front is obtaining by the multiple runs of the formulation for different weights. The compromised optimal multicast tree is obtained by fuzzy-cardinal priority ranking. The performance has been investigated for different sizes of multicast group and for different size networks.

The ***Chapter-V*** presents tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) method for single-objective QoS constrained multicast routing. The novel tree structured encoding scheme that is suited for TSGA and TSPSO is proposed for multicast routing. The performance has been studied for delay-bound, delay and delay-jitter bound multicast routing for cost minimization and residual bandwidth maximization. The performance of the developed algorithms TSGA and TSPSO is compared.

A novel representation scheme that employs topological features is formulated to represent the solutions and various operations in TSGA and TSPSO are performed while preserving the tree-structured representation of the solution. The constraints are added as penalty in the objective/fitness functions. The optimal multicast trees are obtained for delay and delay-jitter bound optimum cost and optimum residual bandwidth on various random networks. The performance of TSGA and TSPSO is compared on the basis of convergence time for varying sizes of multicast group and varying sizes of networks.

The ***Chapter-VI*** presents the tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSMPSO) method for QoS constrained multiobjective multicast routing. The TSMGA and TSMPSO are Pareto-based search and multiobjective optimization algorithms. After obtaining the Pareto-optimal front, the best compromised solution is obtained using fuzzy cardinal priority ranking. The multicast tree is obtained for cost-residual bandwidth and cost-residual bandwidth-packet loss probability optimization under delay and delay-jitter constraints. The performance of the developed algorithms TSMGA and TSMPSO are compared.

The proposed algorithms TSMGA and TSMPSO are developed by implementing the concept of non-dominance for both objectives and constraints to investigate multiobjective multicast routing and to obtain Pareto-optimal front. The tree structured encoding scheme presented in *Chapter-V* is used in TSMGA and TSMPSO. The best compromised optimal tree is obtained using fuzzy-cardinal priority ranking for cost-bandwidth optimization, cost-bandwidth-packet loss probability optimization under delay and delay-jitter bounds for various networks. The performance of TSMGA and TSMPSO is compared on the basis of convergence time for varying sizes of multicast group and varying sizes of networks.

The ***Chapter-VII*** highlights the specific contributions and the main conclusions and also states the scope for further work in this area.

1.6 CONCLUDING REMARKS

In this chapter the communication methods in computer network are reviewed and the state of art on routing in computer network is presented. The research objectives and the organization of the thesis have been summarized.

LITERATURE REVIEW

2.1 GENERAL

The network routing is needed to send information in the form of data packets and streams from a source to the user or group of users through a path or set of paths. During this transfer, it is desired that the network resources are optimally utilized. The conventional techniques of data transfer are unicast and broadcast. However, when the information is sent to only a subset of users, the multicast is advantageous.

Ever since the first implementation of multicast by IETF in 1992, on the Internet over multicast backbone (Casner and Deering, 1992), the researchers have been studying various aspects of multicast routings. The routing algorithms and routing protocols are important aspects of network routing. The routing protocols describe the procedure to implement theoretical routing algorithms to practical network. The routing protocols are usually in the network layer of the open source interconnection stack. Some of the early protocols used by the internet for delivery of data are based on the calculations of shortest paths like open shortest path first (OSPF) (Moy, 1994). The continuously growing size of network, number of users and applications have thrown many challenges like routing mechanism, flow scheduling, resource reservation etc. There are some typical multicast multimedia applications such as video-on-demand, groupware, virtual conference that require strict guarantee on QoS parameters by the network (Gonsalves and Tobagi, 1989; Chen and Nahrstedt, 1998; Xiao and Ni, 1999; Paul and Raghavan, 2002).

The routing algorithms are important in the network routing to obtain the routes for the flow of the information from a source to a set of receivers. The routes have been obtained mainly by heuristics. Recently, the researchers have used various intelligent methods such as population based search and optimization methods and neural networks for obtaining the path/tree. The problem to obtain optimum multicast tree is complex because there exists a conflict in optimizing the resources and satisfying the QoS requirements.

2.2 RELATED WORK

The work reported on various aspects of routing mechanisms that includes routing algorithms, routing protocols, applications etc. is much diversified. Here, the brief literature review is presented on multicast routing algorithms with the objective of complementing the work by various researchers.

The reviews on various aspects of multicast routing have appeared time to time. The issues associated with QoS multicast routing mechanism have been summarized (Wang and Hou, 2000; Striegel and Manimaran, 2002; Masip-Bruin *et al.,* 2006). The work by Oliviera and Pardalos (2004) is the summary of various optimization problems for routing.

The shortest path is a path that minimizes the sum of the weights on the individual links along a path from the source node to a destination node. Using the shortest path algorithms, shortest path tree (SPT) can be obtained. The SPT minimizes the sum of weights on the individual links along each individual path from source node to each multicast group member. The SPT can be least cost tree or least delay tree when the link weight represent cost or delay respectively. If the weights on links are unity, resulting tree is the least hop tree.

The shortest path problem is a fundamental routing problem to graph theory and graph based applications. The shortest path problem has applicability in many engineering areas (Deo and Pang, 1984). These areas include routing in computer, communication and transportation systems, flow control and scheduling, robotic path planning etc. Its various nomenclatures are minimum weight problem, quickest path problem etc. The well-known algorithms to solve shortest path problem are Bellman-Ford algorithm and Dijkstra algorithm to name a few (Stalling, 1998). These algorithms are exact and run in polynomial time. Garey and Johnson (1979) suggested that the delay-constrained shortest path problem is NP-hard and can not be solved in polynomial time. There is a distributed version of the Bellman-Ford algorithm (Awerbuch *et al.,* 1991). Deering and Cheriton (1990) used the reverse path forwarding, which forms the shortest paths from the receivers to source, to present reverse shortest path tree.

There are algorithms to formulate delay-constrained shortest path tree (Sriram *et al.,* 1998; Alrabiah and Znati, 2001). Sriram *et al.* (1998) presented distributed delay constrained least cost tree by using preferred neighboring link approach. Alrabiah and Znati (2001) obtained the delay bounded tree by first forming low cost tree and then accounting for delay constraint.

A minimum spanning tree spans all the members and minimizes the total weight of the tree. This has the application in the design of electric circuit, telephone network, road infrastructure, cable TV network, computer network etc. Similar to shortest path problem, the minimum spanning tree algorithm is polynomial time. The algorithm becomes NP-hard when delay constrained is applied (Bertsekas and Gallager, 1992). Gallager *et al.* (1983) reported the algorithm to construct distributed minimum spanning tree. Salama *et al.* (1997) compared various heuristics for delay constrained minimum spanning tree.

A multicast group is a set of nodes in a network that need to share the same piece of information. A multicast group can be static or dynamic (Waxman, 1988). The static group remains unchanged; however, members can join or leave a dynamic multicast group. The problem of routing information in static groups is frequently modeled as a type of Steiner tree problem. The Steiner tree problem which derives its name from Jacob Steiner (Gilbert and Pollak, 1968), was originally conceived as a problem of geometry. The minimum Steiner tree problem aims to minimize the total cost of the multicast tree and is NP-complete (Hwang and Richards, 1992). The minimum Steiner tree problem reduces to the minimum spanning tree problem when the multicast group includes all nodes in the network. The heuristics (Kou *et al.,* 1981; Takahashi and Matsuyama, 1980) find trees whose cost is within twice of the optimal Steiner tree cost.

The heuristic by Takahashi and Matsuyama (1980) forms a source specific tree that contains the source node while the multicast group members are added one by one to the existing tree via the least cost path to a node already in the tree. The heuristic by Smith (1983) forms trees with each multicast group member and then unites the trees that are closest in terms of cost by adding the appropriate links until it ends up with a single tree.

The heuristics (Jiang, 1993; Bauer and Verma, 1996) obtain distributed minimum Steiner tree. Tanaka and Huang (1993) compared the performance of various static unconstrained minimum Steiner tree algorithms. Ramanathan (1996) proposed a heuristic for asymmetric networks, which permits trading off low tree cost for fast execution time. Barathkumar and Jaffe (1983) presented the heuristics to optimize the cost and delay of the routing tree.

The Steiner tree problem has been extended to include QoS requirements. These problems are also NP-complete and heuristics are used to solve them. The delay-constrained minimum Steiner tree algorithms are mostly centralized and source-specific (Kompella *et al.,* 1993; Parsa and Zhu, 1998; Feng and Yum, 1999). The heuristic (Kompella *et al.,* 1993) works under the assumption that the link delays and the delay constraint are integers while the link costs may take any positive real value. The delay constrained closure graph is first computed and then the least cost path between two nodes is computed. In the bounded shortest multicast routing algorithm proposed by Parsa and Zhu (1998), the link cost is defined as a function of link utilization. A least delay tree rooted at source and spanning all group members is first computed. Thereafter, the tree super edges are iteratively replaced by the cheaper super edges not in the tree. The directed network has been assumed in constructing the Steiner tree (Feng and Yum, 1999). The algorithm presented by Divakaran *et al.* (2005) minimizes the bandwidth under delay jitter for multimedia data transfer for low bandwidth network. The algorithm presented by Kumar *et al.* (2010) analyses the end-to-end delay for mobile ad-hoc networks.

Salama *et al.* (1997) proposed a semi-constrained shortest path broadcast tree algorithm for obtaining Steiner tree by optimizing cost subjected to delay and delay jitter constraints. Noronha and Tobagi (1994) proposed an algorithm based on integer programming to construct the Steiner tree. Rouskas and Baldine (1996) constructed the multicast trees subject to both delay and delay jitter constraints. The bandwidth is used as constraint by Hong *et al.* (2003) in QoS routing.

The heuristics (Jia, 1998; Wang *et al.,* 1999; Sriram *et al.,* 1998) are used for obtaining the constrained distributed Steiner tree. Jia (1998) developed the delay bound algorithm under the consideration that the least cost tree between two nodes is always the shortest path between them. The algorithm presented by Wang *et al.* (1999) constructs multicast tree for multimedia group communication The algorithm proposed by Sriram *et al.* (1998) uses unicast routing concept for generating least cost delay constrained multicast tree.

In the distributed heuristic (Kompella *et al.,* 1993b) every node is assumed to maintain a distance vector of minimum delay to every other node in the network. The tree starts with a source node and receivers are added one by one to augment the tree. A distributed multicast routing algorithm (Huang *et al.,* 2003) constructs the widest available bandwidth tree under the constraint of end-to-end delay and delay jitters.

Dynamic multicast routing algorithms (Kadirire, 1994; Chakraborty *et al.,* 2003) avoid rerouting an entire multicast tree whenever a node joins or leaves a multicast session. In dynamic multicast routing algorithms, when a node leaves a multicast session, the path connecting that node is simply pruned from the tree if it is not used to connect any other multicast group members. The situation is more difficult when a node joins an existing multicast session. Waxman (1988) presented multicast routing algorithms for both static and dynamic multicast groups. The geographic spread dynamic multicast algorithm (Kadirire, 1994) maximizes the geographic spread in constructing the multicast tree. The heuristic presented by Chakraborty *et al.* (2003) uses modified Ballman-Ford algorithm for dynamic multicast routing under multiple constraints.

The neural networks are parallel, distributed information processing structure consisting of many processing elements connected via weighted connections and are considered to exploit the computational power of the human brain. Hopfield and Tank (1985) demonstrated the solution of traveling salesman problem, an optimization problem, through Hopfield neural network. The objective function was expressed as quadratic energy function and the associated weights between neurons are computed using the gradient descent of energy function. The HNN has been used to compute the shortest paths (Rauch and Winarske, 1988; Ali and Kamoun, 1993; Araύjo *et al.,* 2001) for computer and communication networks. Rauch and Winarske (1998) computed the shortest path for communication system. The algorithm to compute the shortest path by Ali and Kamoun (1993) is adaptive because the weight matrix contains the convergence information. The researchers (Pornavalai *et al.,* 1995; Ghanwani, 1998; Feng and Douligers, 2001; Venkataram *et al.,* 2002; Hemmiger, 2002) used neural network to solve QoS constrained multicast routing. Pornavali *et al.* (1995) applied the HNN to obtain delay constrained Stainer Tree. The energy function of shortest path is modified to include the delay constraint. The cost term in the energy function is modified by incorporating penalty to existing links to obtain delay constrained multicast tree. The inequality delay constraints are taken care by linear-programming neuron (Kennedy and Chua, 1987). Feng and Douligers (2001) computed the minimum delay path using HNN for communication system. Gelenbe *et al.* (1997) used random neural network to compute the Steiner tree. Venkataram *et al.* (2002) highlighted that the cost term is appearing in the bias terms and not in connection matrix. The connection matrix is independent to the changes in the network flow that may happen in real time and thus can be realized in hardware. Vijaykumar and Venkataram (2003) proposed a method to construct the multicast route using neural network for mobile network. Kohonen's self-organizing map has been used for clustering of nodes. Thereafter, the HNN is used to construct the multicast tree that passes through the nodes belonging to the centre cluster. Shen and Wang (2008) summarized that the parallel structure of HNN helps realize the optimal solution and it also has the capability for hardware realization. Wang *et al.* (2009) used noisy chaotic neural network, a modification to the HNN model, for delay constrained multicast routing. Liang *et al.* (2011) attempted multi-path routing using HNN. The HNN was applied for multiprocessing by Chen (2011). The complexity of HNN computation was reduced by having conventional two dimensional representations instead of three dimensional. There is a paucity of literature on the applicability of HNN for multiobjective optimization. The objectives are combined by the weighted sum like approach to be optimized by HNN (Balakrishanan *et al.,* 2003, Ahn and Ramakrishnan, 2004; Mehdi and Ali, 2009).

The genetic algorithm (GA) was first developed by Holland as an experiment to use computer program emulating the concept of evolution (Holland J.H., 1975). The GA is a population based evolutionary algorithm that mimics the evolutionary process of selection, variation and genetics to constitute search and optimization procedures. The GA operators namely, reproduction, crossover, and mutation combine genetics with the Darwin's theory of ''Survival of Fittest". The GA represents an intelligent exploitation of an adaptive random search within a defined search space to solve a problem. The GA searches for multiple good solutions in parallel by taking advantage of gene similarities available in the family of possible solutions to the related problem (Goldberg, 1989). The evolutionary programming (EP) by Fogel is similar to GA, however, it does not employ crossover (Fogel *et al.,* 1966).

The particle swarm optimization (PSO) is a heuristic search technique developed by Kennedy and Eberhart for optimization (Kennedy and Eberhart, 2001). The technique was inspired by flocking of birds and animal social behaviours. The particles operate like a swarm exploring the search space for possible optimal solutions. The behaviour of the particles is influenced by their tendency to learn to adjust the flying speed and direction from their own success and from the success of peers. There are studies to compare the performances of GA and PSO (Eberhart and Shi, 1998; Boeringer and Werner, 2004; Elbeltagi and Hegazy, 2005; Pham *et al.,* 2011). The PSO uses leaders to guide the search instead of using an explicit selection as was in GA. Although, there is no notion of offspring generation in PSO, it can be considered an evolutionary algorithm (Eberhart and Shi, 1998). The most attractive feature of PSO is that it requires less computational memory and small implementation code. As the position and velocity of particles are updated in a continuous manner, PSO is suited for continuous optimization problems.

The GA and PSO are successfully applied to various NP complete and NP hard combinatorial optimization problems (Goldberg, 1989; Lin *et al.,* 2003; Liu *et al.,* 2007; Gong *et al.,* 2012). The evolutionary algorithms basically solve the unconstrained optimization problem, whereas most of the problems are constrained optimization type. There are methods to handle constraints (Venkatraman and Yen, 2005; Mallipeddi and Suganthan, 2010) in the evolutionary optimization methods. Mostly the constraints have been added as penalty in the objective function and the constrained problem is solved as unconstrained optimization problem.

The evolutionary algorithms are used for shortest path computation (Ahn and Ramakrishna, 2002; Nesmachnow *et al.,* 2007). An equation is derived by Ahn and Ramakrishna (2002) to determine size of the population for the shortest path problem to be solved by genetic algorithm. There are various papers to report QoS multicast routing for static networks (Ravikumar and Bajpai, 1998; Zhang and Leung, 1999; Zhengying *et al.,* 2001; Haghighat *et al.,* 2004; Randaccio and Atzori, 2007; Yen *et al.,* 2008). The approach proposed by Ravikumar and Bajpai (1998) uses GA to obtain delay bound least cost multicast tree. The method presented by Randaccio and Atzori (2007) employs GA based solution to the group multicast problem by generating a set of possible trees for each session in isolation. Yen *et al.* (2008) considered the energy consumption efficiency in forming the route and the nodes are selected with the minimum energy consumption. The GA is used to form bandwidth-delay constrained least-cost multicast tree (Zhengying *et al.,* 2001; Haghighat *et al.,* 2004).

The PSO has been used for solving various routing problems (Mohemmed *et al.,* 2008; Wang *et al.,* 2010; Mala and Rajagopalan, 2011; Sun *et al.,* 2011). The algorithm for computing the shortest path by PSO (Mohemmed *et al.,* 2008) uses priority-based encoding for particle representation and a heuristic operator for reducing the possibility of loop-formation. Wang *et al.* (2010) obtained the QoS constrained least cost multicast tree using PSO. The algorithm uses tree re-shaping by tree merging and circle elimination for improving the performance. Sun *et al.* (2011) converted the QoS multicast routing problem into an integer programming problem and solved it using quantum behaved PSO.

The search and optimization methods namely simulated annealing (Zhang *et al.,* 2005; Kun *et al.,* 2006), Tabu Search (Youssef *et al.,* 2002; Wang *et al.,* 2004; Shen *et al.,* 2005), Ant colony (Huang *et al.,* 2007; Tsenga *et al.,* 2008) have been applied for multicast routing. Youssef *et al.* (2001) compared the general aspects of evolutionary algorithms, simulated annealing (SA) and Tabu search (TS). Ribeiro *et al.* (2007) presented a review on metaheuristic optimization methods for computer communications. A dynamic multicast routing algorithm based on SA (Kun *et al.,* 2006) finds minimum cost multicast tree by satisfying delay and delay jitter bounds. The QoS multicast routing has been presented using GA, SA and TS for multimedia group communication (Wanga *et al.,* 2006). The ant colony based algorithm (Tsenga *et al.,* 2008) constructs broadcast spanning tree under delay and degree constraints. The hybrid GA (Vijayalakshmia and Radhakrishnanb, 2008; Zahrani *et al.,* 2008; Zhang *et al.,* 2009) is used for multicast routing. In the hybrid GA (Vijayalakshmia and Radhakrishnanb, 2008), the constraints are handled using artificial immune based method. The method combining GA and SA (Zhang *et al.,* 2009) is used for least-cost QoS multicast routing.

Pareto, in 1906, has introduced the concept of non-inferior solution in context of economics, however, application of this concept to various engineering problems was started around 1970 (Stadler, 1979). There are many real-world problems that involve simultaneous optimization of several objective functions. Generally, these objectives are often competing and conflicting. With such conflicting objectives, an optimal solution that is better than all other solution can not be obtained. Therefore, a true multiobjective optimization provides a set of optimal solutions where no one can be considered to be better than any other with respect to all the objective functions. These optimal solutions are known as Pareto-optimal solutions (Deb, 2001). Various multiobjective evolutionary algorithms use the concept of Pareto domination and can obtain multiple and well-spread Pareto-optimal solutions (Zitzler and Thiele 1999; Deb, 2001). A review on various evolutionary multiobjective algorithms is presented (Zhou *et al.,* 2011). There is a mechanism to reduce the run time complexity of evolutionary algorithms (Jensen, 2003). The non-dominated sorting genetic algorithm (Deb *et al.,* 2002) combines the ranking approach and diversity mechanism into the fitness assignment. It can rapidly assign the solutions into several fronts and maintains the diversity. The various aspects of multi-objective PSO have been summarized (Coello *et al.,* 2004; Leong and Yen, 2008; Tsai *et al.,* 2010). The algorithm (Coello *et al.,* 2004) uses mutation to maintain the diversity of swarm and external repository to store particles to be used for guiding. The concept of dynamic population size was used in multiobjective PSO (Leong and Yen 2008). The flow shop scheduling, a combinatorial optimization problem, has been attempted by multiobjective GA (Ishibuchi and Murata 1998) and multiobjective PSO (Li *et al.,* 2008).

The problem of handling constraints for evolutionary multiobjective optimization has drawn the attention of many (Fonseca and Fleming, 1998; Cai and Wang, 2006; Woldesenbet *et al.,* 2009). At the convergence of evolutionary multiobjective optimization, the set of Pareto-optimal solutions is resulted. The best compromised solution from the optimal Pareto solutions was obtained using fuzzy approach (Fuller and Carlsson, 1996; Farina and Amato 2004).

The multi-objective genetic algorithm based approach (Roy and Das, 2004) is used to optimize end-to-end delay, total bandwidth consumption and residual bandwidth utilization. Araújo and Garrozi (2010) combined multicast routing objectives using weighting factors and used single objective optimization. Ant colony approach was used to simultaneously optimize cost, end-to-end delay and average delay (Pinto *et al.,* 2005). The multiobjective simulated annealing (Xu and Qu, 2011) is presented for multicast routing.

2.3 AUTHORS CONTRIBUTION

After the comprehensive review of the literature on the topic of research interest, the following areas are identified for further investigation in this research work –

- The multicast tree using Hopfield neural network (HNN) has been constructed. However, multicast tree has been constructed for delay bound cost minimization only. It is identified to investigate the usage of HNN for construction of multicast tree for optimum bandwidth and for delay-constrained optimum bandwidth as well.

- There is a paucity of literature on the application of HNN for multiobjective optimization. It is vital to investigate the multiobjective multicast routing using HNN without and with end-to-end delay constraint.

- The population based search algorithms although have been used for QoS constrained multicast routing, their performance depends on encoding scheme. It is considered vital to develop a representation scheme exploiting the topological features of the tree and investigate its effectiveness for various QoS constrained multicast routing problems.

- There is a paucity of literature on the multiobjective multicast routing using population based search and optimization methods. It is essential to investigate the effectiveness of the proposed tree representation scheme for the QoS multiobjective multicast routing for different set of objectives and obtain a best compromised tree.

The investigations have been carried out in the identified research areas to achieve the research objectives. The specific contributions to accomplish these research objectives are presented herewith –

I. Multicast Routing Using Hopfield Neural Network

- The delay-constrained routing problems are mapped in HNN model to obtain shortest path tree (SPT) and multicast tree (MT) for cost optimization and bandwidth optimization.

- The SPT is obtained by recursively obtaining delay-constrained shortest paths from source to various destinations and combining them to ensure that a link is appearing only once in the shortest path tree.

- The performance has been studied for optimum cost, delay-constrained optimum cost, optimum residual bandwidth and delay-constrained optimum residual bandwidth.

- The effectiveness of the developed dynamic models of HNN has been studied for different multicast groups.

II. Multiobjective Multicast Routing Using Hopfield Neural Network

- The delay-constrained multiobjective multicast routing problem is mapped into HNN model by weighted sum approach.
- The Pareto-optimal front is constructed by executing the formulation for varying preference to respective objectives.
- The compromised optimal multicast tree is obtained using fuzzy-cardinal priority ranking.
- The compromised optimal multicast trees are obtained for cost-residual bandwidth optimization and delay-constrained cost-residual bandwidth optimization for different multicast groups.

III. Multicast Routing Using Evolutionary Algorithms

- Two algorithms namely tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) are proposed to obtain optimal multicast tree.
- A novel representation scheme that employs topological features is formulated. The solution is represented as ordered M-array structure. Where, each array represents a random path from a destination node to source node. The order of the arrays represent the order of destination node in the multicast group. The multicast tree is constructed by visiting each node of encoded M-arary solution in one-by-one manner. The various operations in TSGA and TSPSO are performed while preserving this tree-structured representation.
- The constraints are added as penalty in the fitness functions and the optimal multicast trees are obtained for delay and delay-jitter bound optimum cost and optimum residual bandwidth on various random networks.

- The performance of TSGA and TSPSO is compared on the basis of convergence time for different multicast groups and the networks of different sizes.

IV. Multiobjective Multicast Routing Using Evolutionary Algorithms

- Two algorithms namely tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSPSO) are proposed for QoS constrained multiobjective multicast routing. The novel structured representation of the solution, as discussed above, is used.

- The proposed algorithms are developed by implementing the concept of non-dominance for both objectives and constrains in obtaining the Pareto-optimal front.

- The compromised optimal tree is obtained using fuzzy-cardinal priority ranking for cost-residual bandwidth optimization, cost-residual bandwidth-packet loss probability optimization under delay and delay-jitter bounds for various networks.

- The performance of TSMGA and TSMPSO is compared on the basis of convergence time for different multicast groups and the networks of different sizes.

2.4 CONCLUDING REMARKS

In this chapter the brief review of the literature on various aspects of routing has been presented. The summary of the author's contributions to the identified research objectives is also presented in this Chapter.

MULTICAST ROUTING PROBLEMS

3.1 GENERAL

The optimization refers to the process of finding an optimal solution and it is relevant to every sphere. A well formulated optimization problem consists of three components namely an objective function, variables and constraints. The optimization process provide optimal solution (values of unknown variables) while optimizing (minimize or maximize) the objective function and satisfying the constraints. During the process, the infinite search space resulted by the possible combinations of values of the variables is explored to yield the optimum value of the objective function.

Many of the real world problems are multiobjective by nature. In the multiobjective optimization (MOO) problem, the objectives representing the goals are conflicting such as cost, comfort, reliability etc. and therefore it is difficult to decide the most suitable selection. Such problems exist in design, scheduling, performance evaluation etc. and have been studied by economists, mathematicians, engineers and scientists. It is also referred by multicriteria decision making or vector optimization.

Stadler (1979) and Dauer and Stadler (1986) presented historical survey on the multiobjective optimization. The MOO is known since the economic equilibrium theory by Adam Smith in 1776. The first mathematical treatment of the MOO problem was given by Edgeworth in 1881. Pareto, in 1906, assumed that every individual seeks to maximize his utility in case nothing stands in his way. This has established the concept now being referred as Pareto optimum. The phrase Pareto optimum was first used by Little in 1950. In 1951, the notion of efficient point was first introduced by Koopmans, which is now being known as concept of non-dominance. Around the same time in 1951, Kuhn introduced first formal mathematical treatment of MOO.

The typical problems that are associated with networks or graphs are shortest path formulation and Steiner tree formulation. The shortest path problem is regarded as polynomial time in terms of complexity theory. Such problems can be solved by deterministic algorithms in polynomial time. With the increase in complexity, even by incorporating constraints, such as delay-constrained shortest path, the problem becomes nondeterministic polynomial (NP). The NP-complete problems, such as QoS multicast routing problems, are the hardest problems (Garey and Johnson, 1979).

This chapter introduces the single objective and multiobjective optimization where the concepts of non-dominance, Pareto-front and compromised solution are summarized. The network model and various QoS constraints are discussed. The mathematical formulations of studied multicast routing optimization problems are also summarized.

3.2 CONCEPT OF MULTIOBJECTIVE OPTIMIZATION

A single objective optimization or simply optimization is a process to obtain the decision vector **x** by seeking the optimum (minimum or maximum) value of a well defined objective. The mathematical formulation is given as –

$$\textit{find } \mathbf{x} = [x_1, x_2, x_3 ....x_n]^T$$

$$\begin{array}{ll} \textit{optimize} & f(\mathbf{x}) \\ \textit{subjected to} & \mathbf{g}(\mathbf{x}) \le 0 \\ & \mathbf{h}(\mathbf{x}) = 0 \\ & \mathbf{x} \in \mathbf{\Omega} \end{array} \tag{3.1}$$

where, **x** is a decision vector, **Ω** is decision space, **g**(**x**) is a set of inequality constraints and **h**(**x**) is a set of equality constraints.

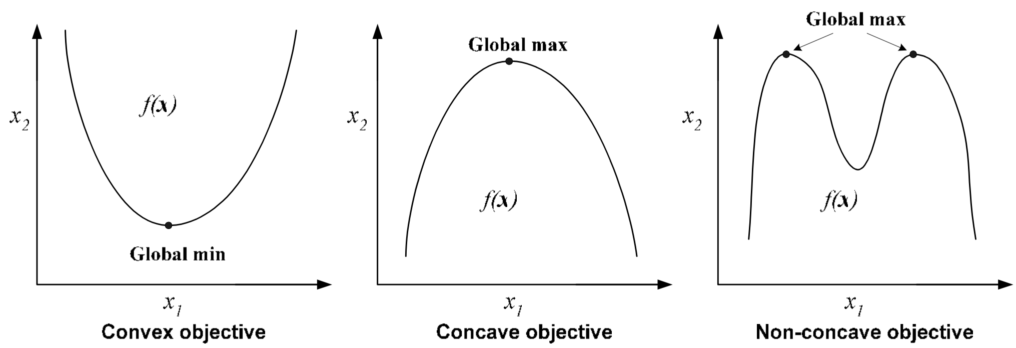


Fig. 3.1 Various objectives and optimal solution

As shown in Fig. 3.1, if the problem is convex for a minimizing or concave for a maximizing objective function, there exists only one optimal solution. If the problem is non-convex or non-concave, there exists more than one optimal solution having the same objective function value.

Although, the single objective optimization provides a powerful tool to explore the decision space, many real world problems are multiobjective by nature. A multiobjective optimization problem can be formulated as -

$$\begin{aligned} &\textit{optimize} \quad \mathbf{f}(\mathbf{x}) = [f_1(\mathbf{x}), f_2(\mathbf{x}), f_3(\mathbf{x}), ...f_Z(\mathbf{x})]^T \\ &\textit{subjected to} \quad \mathbf{g}(\mathbf{x}) \leq 0 \\ &\qquad\qquad\qquad \mathbf{h}(\mathbf{x}) = 0 \\ &\qquad\qquad\qquad \mathbf{x} \in \mathbf{\Omega} \end{aligned} \tag{3.2}$$

where, $\mathbf{f}(\mathbf{x})$ is a set of Z objective functions, $\mathbf{x}$ is decision vector, $\mathbf{\Omega}$ is decision space, $\mathbf{g}(\mathbf{x})$ is a set of inequality constraints and $\mathbf{h}(\mathbf{x})$ is a set of equality constraints.

The objectives in eq. (3.2) often conflict with each other. Improvement of one objective may lead to deterioration of another objective. Thus, a single solution which can optimize all objectives simultaneously does not exist. In multi-objective optimization, the effort is made in finding the set of trade-off optimal solution by considering all objectives to be equally important. These trade-off solutions are called Pareto-optimal solution. The decision to make a single choice of optimal solution, referred as compromise-optimal solution, can be made after using higher level qualitative consideration. Therefore, the procedure involves two steps as shown in Fig. 3.2 and summarized as follows:

Step 1 : Run the multiobjective optimization to find multiple trade-off optimal solutions.

Step 2 : Choose compromise-optimal solution using higher level information.

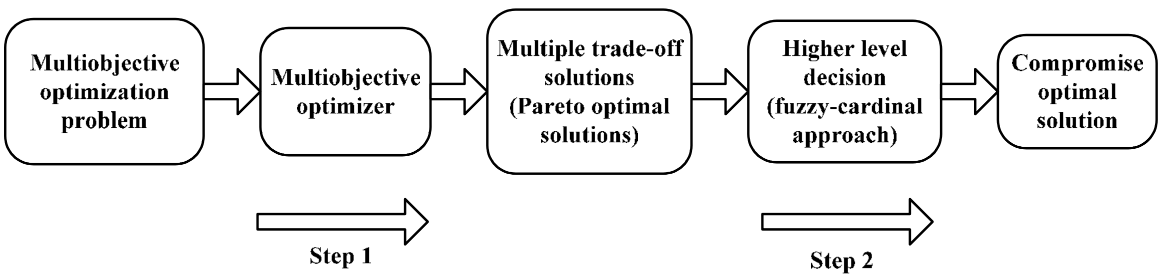


Fig. 3.2 Illustration of multiobjective optimization procedure

3.2.1 Domination and Pareto Optimality

A solution is said to be *Pareto optimal* if it is not dominated by any other solution in the solution space. A Pareto optimal solution cannot be improved with respect to any objective without worsening at least one other objective. The set of all feasible non-dominated solutions in the decision space is referred to as the *Pareto optimal set,* and for a given Pareto optimal set, the corresponding objective function values in the objective space is called the *Pareto optimal front*.

The multiobjective optimization uses a concept of **domination**. A solution $\mathbf{x}^u$ is said to dominate another solution $\mathbf{x}^v$ ($\mathbf{x}^u \lhd \mathbf{x}^v$) if the following conditions are true:

1. The solution $\mathbf{x}^u$ is no worse than $\mathbf{x}^v$ in all objectives.
2. The solution $\mathbf{x}^u$ is strictly better than $\mathbf{x}^v$ in at least one objective.

The notion $\mathbf{x}^u \lhd \mathbf{x}^v$ suggests that $\mathbf{x}^u$ dominates $\mathbf{x}^v$ or $\mathbf{x}^u$ is non-dominated by $\mathbf{x}^v$ or simply $\mathbf{x}^u$ is better than $\mathbf{x}^v$. This notion of dominance is true regardless of the nature of objectives i.e. minimization or maximization type. Among a set of feasible solutions in the search space **P**, the non-dominated set of solutions **P′** are the solutions that are not dominated by any member of the set **P**. Then the set **P′** is called Pareto-optimal set. This is explained with the help of an example optimization problem comprising two objectives both of minimization type, hereby referred as min-min type problem, and the Pareto front is shown in Fig. 3.3. The non-dominated solutions satisfying the above mentioned two conditions are joined by continuous thick line representing the Pareto-optimal front. The Pareto-optimal front for min-max, max-max and max-min type of optimizations is shown in Fig. 3.4 (a), Fig. 3.4(b), Fig. 3.4(c) respectively. The Pareto-optimal front is always coinciding with the boundary in the feasible solution space.

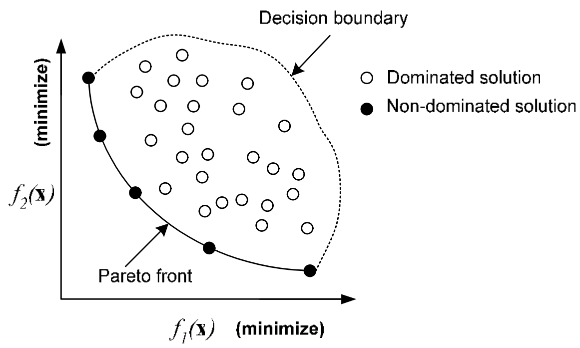


Fig. 3.3 Illustration of non-dominance and Pareto front for min-min optimization

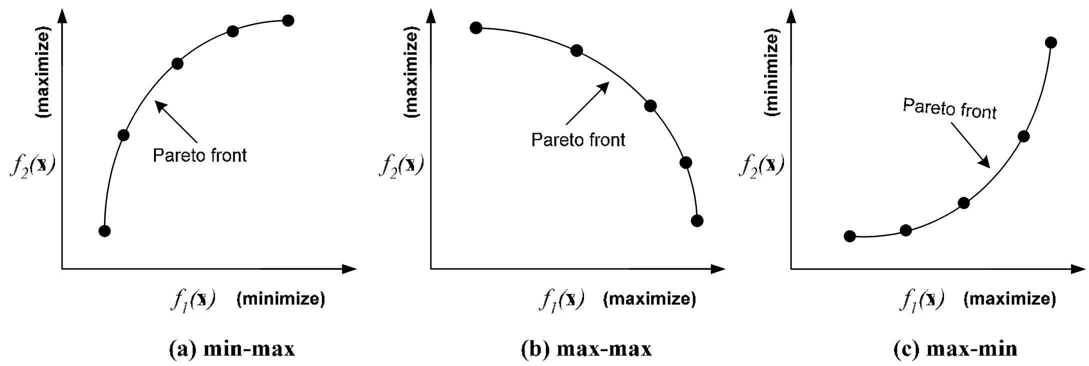


Fig. 3.4 Illustration of Pareto fronts for various two-objective optimization problems

3.2.2 Best Compromised Solution

After convergence of the multiobjective optimization, a set of Pareto optimal solutions is achieved. These are the nondominated solutions and are contained in the Pareto-optimal front. Each of these solutions is better to other solutions in at least one objective or the improvement in one objective can be only after sacrificing other objectives. For practical purposes, one solution among Pareto optimal solutions satisfying different goals to some extend is selected and is known as best compromised solution. The best compromised solution is selected using fuzzy-cardinal priority ranking. It is implemented through the following steps:

Step 1 Calculate the normalize membership function β_k, which provides the fuzzy cardinal priority ranking of the non-dominated, for each solution in the Pareto-optimal front. Let, there are *K* solutions in the Pareto-optimal front then β_k is calculated as –

$$\beta_k = \frac{\sum_{z=1}^{Z} u_z^k}{\sum_{z=1}^{Z}\sum_{k=1}^{K} u_z^k} \qquad \text{for } k=1,2...K \qquad (3.3)$$

Where,

$$u_z^k = \frac{f_z^{\max} - f_z^k}{f_z^{\max} - f_z^{\min}} \qquad (3.4)$$

Where, the u_z^k represents the membership value of *k*th solution and *z*th objective.

Step 2 Select the solution for which the value of β_k is maximum. The corresponding solution is regarded as best compromised optimal solution.

3.3 NETWORK MODEL

The network model is well reported in the literature. The network is simply represented as weighted undirected symmetric graph $G(\overline{V}, \overline{E})$, where $\overline{V}$ denotes the set of nodes and $\overline{E}$ the set of links. The existence of a link *e=(u,v)* from node *u* to node *v* implies the existence of a link *e'=(v,u)* for any $u, v \in \overline{V}$. Let M be a subset of $\overline{V}$ i.e. $M \subseteq \overline{V}$ forms the multicast destination group with each node of *M* is a group member. The node $s \in \overline{V}$ is a multicast source for multicast group *M*. The packets originating from source node have to be delivered to multicast group nodes. A multicast tree *T(s,M)* is a sub-graph of $G(\overline{V}, \overline{E})$ that spans all nodes in *M*, while it may include non-group member nodes as well in the tree.

Each link $e=(u,v) \in \overline{E}$ has associated specifications such as cost *C(e),* delay *D(e*), packet-loss-probability *pl(e),* link capacity $\phi(e)$ and bandwidth utilization $\lambda(e)$ as real positive value. The link cost *C(e)* may be the monitory cost incurred by the use of the network link or communication link. The delay *D(e),* expressed in *ms,* represents the measure of time needed to transmit information through link. The delay includes transmission, queuing and propagation delays. The packet-loss-probability *pl(e)* is the probability of failure of data delivery. This failure of data delivery is resulted during congestion or low bandwidth situation. The link capacity $\phi(e)$, expressed in *Mbps,* represents the maximum data that can be transferred by a link. It is assumed uniform for all links. The bandwidth utilization $\lambda(e)$, expressed in *Mbps,* represents the current traffic demand on the link. The available bandwidth $\alpha(e)$, expressed in *Mbps,* is expressed as the difference of link capacity and the utilized bandwidth. High available bandwidth is needed to avoid the packet loss.

The routing optimization has been attempted through the mathematical formulation of various measures. The performance matrices or measures are classified as path metric and tree metric.

3.3.1 Path metrics

During the multicast routing there exists a path from source to each destination in the multicast group. The path metric represents the value of a performance metric derived from the links measures forming a path from source *s* to a destination node d_i in the multicast group i.e. $P(s, d_i)$. These measures include path cost, end-to-end delay, packet-loss probability and bandwidth availability. These metrics are expressed as :

Path cost : It is additive in nature and is the sum of the cost associated with the links forming a path. It is expressed as -

$$C(P(s,d_i)) = \sum_{e \in P(s,d_i)} C(e) \qquad \text{for } d_i \in M \tag{3.5}$$

Path delay : It is additive in nature and is the sum of the delay $D(e)$ associated with the links forming a path. It is also referred as end-to-end delay and is expressed as –

$$D(P(s,d_i)) = \sum_{e \in P(s,d_i)} D(e) \qquad \text{for } d_i \in M \tag{3.6}$$

Path packet loss probability : The probability of data transmission is multiplicative and is indicative of the reliability of the path. The reliability or the probability of data transmission along a path from source to destination node is equal to the product of probability of data transmission of a link. With this the packet-loss-probability is calculated as -

$$PL(P(s,d_i)) = 1 - \prod_{e \in P(s,d_i)} (1 - pl(e)) \tag{3.7}$$

3.3.2 Tree metrics

During the multicast routing, the tree $T(s,M)$ is constructed, which connects source *s* to all the nodes *M* in the multicast group. Thus, there exists a path from source to each destination. The tree metric represents the value of performance metric derived from the path metrics or the link specifications expressed above. The tree metrics include, tree cost, tree delay, delay-jitter or inter-destination delay, residual bandwidth and packet-loss-probability of the tree (Roy and Das, 2004; Wang and Hou, 2000; Yen et al., 2008). These metrics are defined as –

Tree cost : It is additive in nature and is the sum of cost associated with all the links in the multicast tree. The multicast tree has been formed by minimizing the tree cost by various researchers.

$$C(T(s,M)) = \sum_{e \in T(s,M)} C(e) \tag{3.8}$$

Tree delay : The tree delay is the sum of the transmission delay associated with all the links in the multicast tree (Yen et al., 2008). It is additive and can be taken as objective in obtaining least delay tree.

$$D(T(s,M)) = \sum_{e \in T(s,M)} D(e) \tag{3.9}$$

Residual bandwidth : The residual bandwidth is a measure of the future use of the multicast tree. The effort should be to maximize the residual bandwidth. The residual bandwidth of the tree *RB(T(s,M))* is defined as the ratio of total available bandwidth to total capacity (Roy and Das, 2004). It is expressed as -

$$RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)} = \frac{\sum_{e \in T(s,M)} (\phi(e) - \lambda(e))}{\sum_{e \in T(s,M)} \phi(e)} \tag{3.10}$$

Packet-loss-probability : The packet loss probability of tree $PL(T(s,M))$ is the maximum value among the packet loss probabilities of paths $PL(P(s,d_i))$ from source to various destination nodes in multicast group. It is expressed as -

$$PL(T(s,M)) = \max_{d_i \in M}(PL(P(s,d_i))) \tag{3.11}$$

where, the path packet loss probability $PL(P(s,d_i))$ is calculated by eq. (3.7).

Delay bound : The end-to-end delay or the path delay from source to each destination in the multicast group should not exceed the specified delay bound Δ. With reference to the path delay $D(P(s,d_i)$ expressed by eq. (3.6), the delay bounds can be expressed as -

$$D(P(s,d_i)) \le \Delta \qquad \text{for } d_i \in M \tag{3.12}$$

The eq. (3.12) is a set of equations depending on number of destination modes in the multicast group. The delay bound as a tree measure $DB(T(s,M))$ is represented as an equation in the optimization problem as-

$$DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \le \Delta \quad (3.13)$$

Delay jitter bound : The delay jitter is also referred as inter-destination delay. It is the difference of the path delays from source to two different destinations. The delay jitter is indicative of time difference experienced by two receivers (or destinations) in receiving a data. It is very important for multimedia applications such as virtual classrooms, video-on-demand etc. The delay jitter is expressed as-

$$DJ(T(s,\{d_i,d_j\})) = \left|D(P(s,d_i)) - D(P(s,d_j))\right| \quad \text{for } d_i,d_j \in M \quad (3.14)$$

where, the path delay $D(P(s,d_i))$ is expressed by eq. (3.6).

The **delay-jitter** bounds should not exceed the specified jitter bound δ. It can therefore be expressed as -

$$DJ(T(s,\{d_i,d_j\})) = \left|D(P(s,d_i)) - D(P(s,d_j))\right| \le \delta \quad \text{for } d_i,d_j \in M \quad (3.15)$$

The eq. (3.15) is a set of equations depending on number of destination modes in the multicast group. The delay-jitter bound as a tree measure $DJ(T(s,M))$ is represented as an equation in the optimization problem as-

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i))\right) \le \delta \quad (3.16)$$

3.4 MULTICAST ROUTING OPTIMIZATION PROBLEMS

In the present day multicast routing, there is increasing thrust on Quality of Service (QoS). The QoS is a set of service requirements to be met by the network while transporting data from source to destinations with an associated service guarantee. Various QoS parameters can be end-to-end delay, bandwidth, cost, delay jitter and packet loss probability etc. The QoS routing is important when transferring the information or data from source to potentially large number of destinations, while optimizing the usage of network resource and meeting the QoS requirements of end users. The QoS routing also suggests to gracefully degrade network performance when things like congestion happen.

The network model and path/tree metrics are summarize in section 3.3. To meet the research objectives, the multicast tree has been constructed by optimizing an objective and optimizing various objectives simultaneously. This gives rise to the terminology of multicast routing and multiobjective multicast routing. The nature of problem will change with the inclusion of constraint representing QoS measure. These problems have been attempted as unconstrained multicast routing, delay-constrained multicast routing and delay and delay-jitter bound multicast routings. With the consideration of objectives to be optimized, the various routing problems are summarized in the following two categories -

- Multicast Routing
- Multiobjective Multicast Routing

3.4.1 Multicast Routing

With the multicast routing, the formulation of multicast tree is considered by optimizing only single objective function. The multicast tree is formulated with the following consideration –

- Minimum cost tree
- Optimum residual bandwidth tree

3.4.1.1 Minimum Cost Tree

The total tree cost $C(T(s,M))$ expressed as eq. (3.8) is minimized to obtain the minimum cost Steiner tree. The formulation therefore is-

$$\text{Minimize } f(\mathbf{x}) = C(T(s,M)) = \sum_{e \in T(s,M)} C(e) \qquad (3.17))$$

The Steiner tree that minimizes the total tree cost and satisfied the delay bound constraint for all the destinations in the multicast group is obtained using eq. (3.8) and eq. (3.13) with the following structure of the optimization problem :

$$\text{Minimize } f(\mathbf{x}) = C(T(s,M))$$

$$\text{Subjected to } \quad DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \le \Delta \qquad (3.18)$$

The Steiner tree that minimizes the tree cost and simultaneously satisfies the delay and delay-jitter bounds for all the destinations in the multicast group is obtained eq. (3.8), eq. (3.13) and eq. (3.18) with the following structure of the optimization problem :

Minimize $f(\mathbf{x}) = C(T(s,M))$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta$

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i)) \right) \leq \delta \qquad (3.19)$$

3.4.1.2 Optimum Residual Bandwidth Tree

The Steiner tree that maximizes the total residual bandwidth $RB(T(s,M))$, as expressed by eq. (3.10) of the multicast tree is obtained with the consideration of the objective as -.

$$\text{Maximize } f(\mathbf{x}) = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)} \qquad (3.20)$$

The Steiner tree that maximizes the residual bandwidth and satisfies the delay bound constraint for all the destinations in the multicast group is obtained using eq. (3.10) and eq. (3.13) with the following structure of the optimization problem :

Maximize $f(\mathbf{x}) = RB(T(s,M))$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta$ (3.21)

The Steiner tree that maximizes the residual bandwidth and simultaneously satisfies the delay and delay-jitter bounds for all the destinations in the multicast group is obtained using eq. (3.10), eq. (3.13) and eq. (3.16) with the following structure of the optimization problem :

Maximize $f(\mathbf{x}) = RB(T(s,M))$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta$

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i)) \right) \leq \delta \qquad (3.22)$$

3.4.2 Multiobjective Multicast Routing

The multiobjective multicast routing has been attempted by simultaneously optimizing various constraints. The following optimization problems have been attempted during multiobjective multicast routing :

- Optimum Cost-Residual Bandwidth Tree
- Optimum Cost-Residual Bandwidth-Packet Loss Probability Tree

3.4.2.1 Optimum Cost-Residual Bandwidth Tree

The twin conflicting objectives cost and residual bandwidth are considered for multiobjective optimization. The tree cost *C(T(s,M))* is minimized and the residual bandwidth *RB(T(s,M))* is maximized for tree formulation. The objective function vector $f(\mathbf{x})$ is defined as –

$$\text{Optimize } f(\mathbf{x}) = [\text{minimize } f_1(\mathbf{x}),\ \text{maximize } f_2(\mathbf{x})]^T \tag{3.23}$$

Where, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are expressed as eq. (3.8) and eq. (3.10) respectively.

$$f_1(\mathbf{x}) = C(T(s,M)) = \sum_{e \in T(s,M)} C(e)$$

$$f_2(\mathbf{x}) = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)} \tag{3.24}$$

The delay-constrained optimum cost-residual bandwidth tree is obtained while optimizing the objective function vector expressed by eq. (3.23) subjected to the delay bounds for all the destinations simultaneously. The structure of the problem becomes –

$$\text{Optimize } f(\mathbf{x}) = [\text{minimize } f_1(\mathbf{x}),\ \text{maximize } f_2(\mathbf{x})]^T$$

$$\text{Subjected to} \quad DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \le \Delta \tag{3.25}$$

The delay and delay jitter bound optimum cost-residual bandwidth tree is obtained while optimizing the objective function vector defined by eq. (3.23) and satisfying delay and delay-jitter bounds for all the destinations simultaneously. The structure of the problem becomes –

Optimize $f(\mathbf{x})$ =[minimize $f_1(\mathbf{x})$, maximize $f_2(\mathbf{x})]^T$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta$

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i))\right) \leq \delta \qquad (3.26)$$

3.4.2.2 Optimum Cost-Residual Bandwidth-Packet Loss Probability Tree

The conflicting objectives cost, residual bandwidth and packet loss probability associated with tree are considered for multiobjective optimization. The cost *C(T(s,M))* is minimized, residual bandwidth *RB(T(s,M))* is maximized and the packet loss probability *PL(T(s,M))* is minimized for the tree formulation. The objective function vector $f(\mathbf{x})$ is defined as :

$$\text{Optimize } f(\mathbf{x}) = [\text{minimize } f_1(\mathbf{x}),\ \text{maximize } f_2(\mathbf{x}),\ \text{minimize } f_3(\mathbf{x})]^T \qquad (3.27)$$

Where, $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ and $f_3(\mathbf{x})$ are expressed as eq. (3.8), eq. (3.10) and eq. (3.11) respectively.

$$f_1(\mathbf{x}) = C(T(s,M)) = \sum_{e \in T(s,M)} C(e)$$

$$f_2(\mathbf{x}) = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)}$$

$$f_3(\mathbf{x}) = PL(T(s,M)) = \max_{d_i \in M}(PL(P(s,d_i))) \qquad (3.28)$$

The delay-constrained optimum cost-residual bandwidth-packet loss probability tree is obtained while optimizing the objective function vector expressed by eq. (3.27) subjected to the delay bounds for all the destinations simultaneously. The structure of the problem becomes –

Optimize $f(\mathbf{x})$ = [minimize $f_1(\mathbf{x})$, maximize $f_2(\mathbf{x})$, minimize $f_3(\mathbf{x})]^T$

$$\text{Subjected to } DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta \qquad (3.29)$$

The delay and delay jitter bound optimum cost-residual bandwidth--packet loss probability tree is obtained while optimizing the objective function vector defined by eq. (3.28) and satisfying delay and delay-jitter bounds for all the destinations simultaneously. The structure of the problem becomes –

Optimize $f(\mathbf{x})$ = [minimize $f_1(\mathbf{x})$, maximize $f_2(\mathbf{x})$, minimize $f_3(\mathbf{x})]^T$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \leq \Delta$

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i))\right) \leq \delta \qquad (3.30)$$

3.5 CONCLUDING REMARKS

In this chapter, the concepts of multiobjective optimizations are reviewed through non-dominance, Pareto-optimal front and best compromised solution. The network model and various metrics for path and tree are discussed. The mathematical formulations of QoS multicast routing problems namely cost minimization, residual bandwidth maximization and QoS multiobjective multicast routing problems namely cost-residual bandwidth optimization and cost-residual bandwidth-packet loss probability optimization are discussed. The delay and delay jitter are considered QoS constraints. These formulations are used for various optimization studies that are carried out in various Chapters.

MULTICAST ROUTING USING HOPFIELD NEURAL NETWORK

4.1 GENERAL

The Hopfield neural network (HNN) is a parallel, distributed information processing structure consisting of many processing elements connected via weighted connections. The feedback and massive parallel structure of HNN help to obtain the near optimal solution. The HNN has the capability for hardware realization (Shen and Wang, 2008). The use of HNN to solve constrained optimization problem was initiated by Hopfield and Tank (1985). The traveling salesman problem was solved using HNN. The objective function was expressed as quadratic energy function and the associated weights between neurons are computed using the gradient descent of energy function. The energy function decreases monotonically with the progress of iterations and leads to stable operation.

Ali and Kamoun (1993) proposed the formulation of energy function associated with HNN for shortest path computation. In this the convergence information is stored in the matrix. Pornavali *et al.,* (1995) applied the HNN to obtain delay constrained Stainer tree. The energy function of shortest path is modified to suit the delay constrained shortest path and then changed this energy function to obtain delay constrained multicast tree. The inequality delay constraints are taken care by converting them into equality constraint by linear-programming neuron. The LP neuron (Kennedy and Chua, 1987) after receiving the delay and delay constraint of the path, penalizes the neurons in the corresponding matrix for constraint violation. Feng and Douligers (2001) computed the minimum delay path using HNN for communication system. Wang *et al.,* (2009) used noisy chaotic neural network, a modification to the HNN model, for delay constrained multicast routing. Vijaykumar and Venkataram (2003) proposed a method to construct the multicast route using neural network for mobile network. Liang *et al.,* (2011) attempted multi-path routing using HNN. For the multiobjective optimization using HNN, the objectives are combined by the weighted sum like approach (Balakrishanan *et al.,* 2003, Ahn and Ramakrishnan, 2004; Mehdi and Ali, 2009).

This chapter presents the use of Hopfield neural network (HNN) to investigate the single and multiobjective multicast routing. The dynamic model of HNN is briefly reviewed and routing problems are mapped in the HNN model. Single objective routing has been attempted using shortest path tree (SPT) and multicast tree (MT) formulations. The performance has been studied for minimum cost and optimum residual bandwidth. The multiobjective optimum cost-residual bandwidth is attempted through MT approach by forming weighted sum objective function. The compromise optimal multicast tree is obtained by fuzzy-cardinal ranking.

4.2 HOPFIELD NEURAL NETWORK MODEL

Hopfield has represented the neural network as an analog circuit suited for solving combinatorial optimization problems. Each neuron is represented as the electrical arrangement of nonlinear operational amplifier, resistance, capacitance and current generator. The electrical circuit equivalent of neural network and a neuron are shown in Fig. 4.1 (a) and Fig. 4.1(b) respectively. The neural network has multiple feedback loops. The output of a neuron is fed back to each other neuron. However, it has no self feedback.

The *i*th neuron produces output voltage V_i that is related to input voltage U_i by continuous differentiable activation function $g_i(U_i)$. This activation function $g_i(U_i)$ is expressed as –

$$V_i = g_i(U_i) = \frac{1}{1 + e^{-\theta_i U_i}} \tag{4.1}$$

where, θ_i is the gain of amplifier for *i*th neuron. The output V_i is bounded between the upper and lower saturation levels which are 1 and 0 respectively.

From the neuron model, as shown in Fig. 4.1(b), the circuit equation is obtained by Kirchoff's current law and is written as –

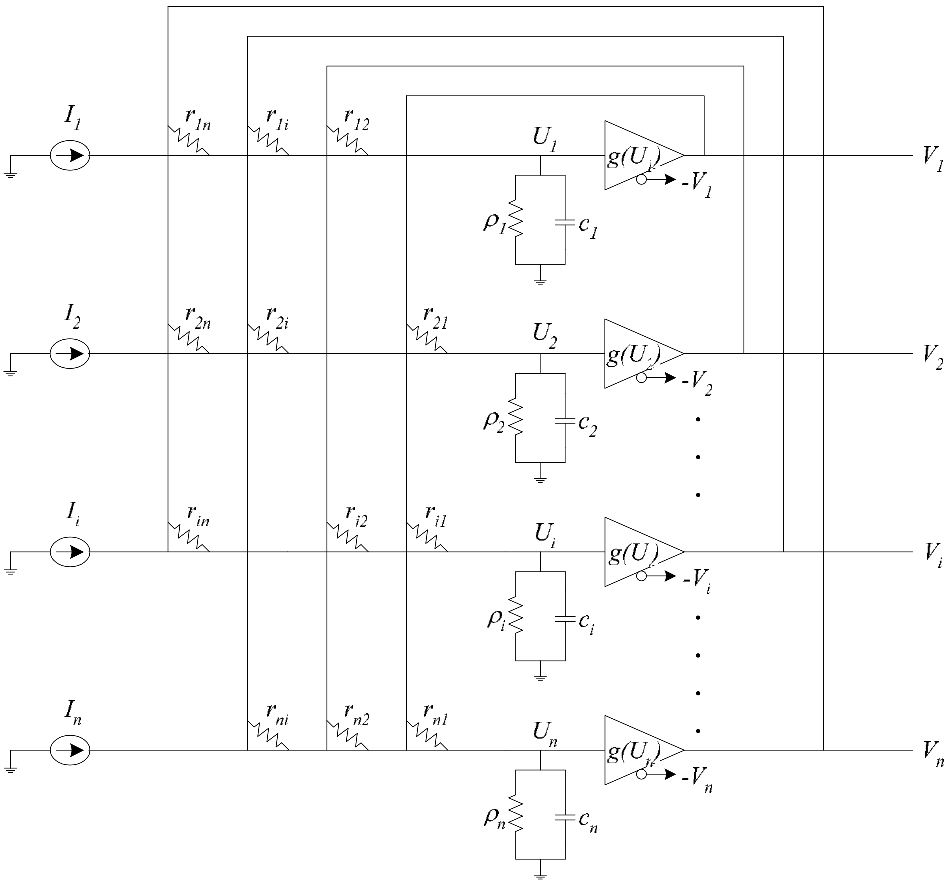


Fig. 4.1(a) Representation of Hopfield Neural Network by Electrical Equivalent Circuit

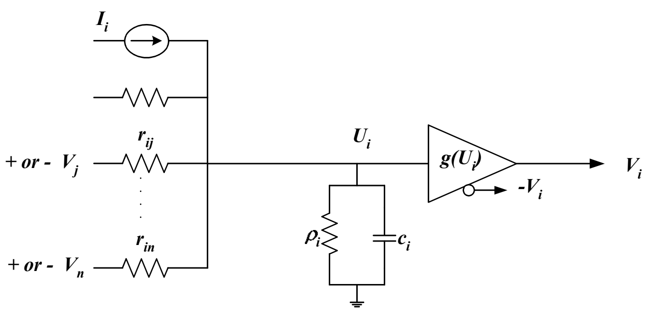


Fig. 4.1(b) Circuit Representation of a Neuron

$$
\begin{aligned}
c_i \frac{dU_i}{dt} &= \sum_{j=1}^{N} \frac{1}{r_{ij}} (V_j - U_i) - \frac{U_i}{\rho_i} + I_i \\
&= \sum_{j=1}^{N} \frac{V_j}{r_{ij}} - \left(\sum_{j=1}^{N} \frac{1}{r_{ij}} + \frac{1}{\rho_i} \right) U_i + I_i
\end{aligned}
\tag{4.2}
$$

The eq. (4.2) is re-written as-

$$c_i \frac{dU_i}{dt} = \sum_{j=1}^{N} T_{ij} V_j - \frac{U_i}{r_i} + I_i \tag{4.3}$$

Where,

$$\frac{1}{r_i} = \sum_j \frac{1}{r_{ij}} + \frac{1}{\rho_i} \quad \text{and} \quad T_{ij} = \frac{1}{r_{ij}} \tag{4.3a}$$

In the above equation, the $T_{ij} = 1/r_{ij}$ represents the interconnecting conductance (weights) between the ith and jth neuron. The r_i and c_i are the leakage resistance and leakage capacitance of the amplifier. Each neuron also receives an external bias input current I_i representing user defined input to neural network.

The Lyapunov energy function associated with the HNN is expressed as-

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij} V_i V_j + \sum_{i=1}^{N} \frac{1}{r_i}\int_0^{V_i} g_i^{-1}(V)\,dV - \sum_{i=1}^{N} I_i V_i \tag{4.4}$$

For symmetric network ($T_{ij}=T_{ji}$), the time derivative of energy function E results into –

$$\begin{aligned} \frac{dE}{dt} &= -\sum_{i=1}^{N} \frac{dV_i}{dt}\left(\sum_{j=1}^{N} T_{ij} V_j - \frac{U_i}{r_i} + I_i\right) \\ &= -\sum_{i=1}^{N} \frac{dV_i}{dt} c_i \frac{dU_i}{dt} \\ &= -\sum_{i=1}^{N} c_i \left(\frac{dV_i}{dt}\right)^2 \frac{dU_i}{dV_i} \end{aligned} \tag{4.5}$$

As the activation function is monotonically increasing, it suggests -

$$\frac{dU_i}{dV_i} > 0 \tag{4.6}$$

With the substitution of eq. (4.6) into eq. (4.5), the eq. (4.5) results as -

$$\frac{dE}{dt} \le 0 \tag{4.7}$$

The eq. (4.7) suggests that energy E always decreases (when $dE/dt < 0$) or remain unchanged ($dE/dt = 0$). The unchanged energy, which is resulted when $dV_i/dt = 0$, suggests that stable state has been reached. Therefore, the HNN always converges to stable state.

It is difficult to add second term $\sum_{i=1}^{N} \frac{1}{r_i} \int_0^{V_i} g_i^{-1}(V)dV$ pertaining to the real problems to energy function eq. (4.4) and can be eliminated (Hopfield and Hank, 1985). After neglecting the second term from eq. (4.4), the energy function that is to be used for optimization becomes-

$$E = -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} T_{ij}V_iV_j - \sum_{i=1}^{N} I_iV_i \qquad (4.8)$$

To simplify the dynamic model, all the neurons are assumed to be characterized by same parameters r_i and c_i. After normalizing the bias current I_i and interconnection weights T_{ij}, with respect to capacitance c_i, the eq. (4.3) describing the dynamics of Hopfield neuron can be written as –

$$\frac{dU_i}{dt} = \sum_{j=1}^{N} T_{ij}V_j - \frac{U_i}{\tau} + I_i \qquad (4.9)$$

Where, time constant $\tau = r_i c_i$.

Combining eq. (4.8) and eq. (4.9), the time derivative dU_i/dt can be expressed as-

$$\frac{dU_i}{dt} = -\frac{U_i}{\tau} - \frac{\partial E}{\partial V_i} \qquad (4.10)$$

The minimum of the energy function occurs at 2^N corners in N-dimensional hypercube defined by $V \in \{0,1\}$.

4.3 MAPPING ROUTING TREE UNDER HNN

The network model and various tree and path measures are briefed in *Chapter 3*. As discussed, the network is represented as undirected graph $G(\overline{V}, \overline{E})$, where $\overline{E}$ is the set of edges or links and $\overline{V}$ is the set of nodes. There is a cost *C(e)*, delay *D(e)* and available bandwidth $\alpha(e)$ associated with each link $e(i,j)$. These cost, delay and bandwidth therefore are also represented as double subscript nomenclature C_{ij}, D_{ij} and α_{ij} respectively. The tree for various objectives under consideration has been constructed using an ($N \times N$) path inclusion criterion matrix $\boldsymbol{V}$, where N is the number of nodes. The each element of the matrix, which is described by double indices (i,j), is representing a neuron. These indices i and j represent the node numbers and denote the row and column number respectively. The diagonal elements of this matrix $\boldsymbol{V}$ are zero. The neuron at the location (i,j) is characterized by its output V_{ij} defined as follows –

$$V_{ij} = \begin{cases} 1 & \textit{if link from node i to node j is in tree} \\ 0 & \textit{otherwise} \end{cases} \tag{4.11}$$

The link connection indicator φ_{ij} is also used and is defined as -

$$\varphi_{ij} = \begin{cases} 1 & \textit{if link from node i to node j does not exist} \\ 0 & \textit{otherwise} \end{cases} \tag{4.12}$$

The HNN is used to find the tree rooted at the source s and spanning to all the members of multicast group $\bar{M} = \{d_1, d_2, ...d_m ...d_M\}$. With this nomenclature, the size of the multicast group M indicates the number of destinations. Thus, there exists a path from source s to each destination node d_m in the multicast group. The tree can be different with the consideration of different objectives or QoS parameters as constraint. For delay-constrained tree construction, the end-to-end delay from source to each destination should not be greater than the specified delay constraint Δ. The end-to-end delay constraint is expressed as -

$$D(P(s,d_m)) = \sum_{e \in P(s,d_m)} D(e) = \sum_{i,j \in P(s,d_m)} D_{ij} \leq \Delta \qquad \textit{for } d_m \in \bar{M} \tag{4.13}$$

The path $P(s,d_m)$ is an ordered sequence of nodes connecting source node s to destination node d_m as ($s \rightarrow i \rightarrow j \rightarrow \; ..... \rightarrow l \rightarrow d_m$).

The HNN is applied to construct the following unconstrained and delay constrained trees –

- Shortest path tree
- Multicast tree
- Multiobjective multicast tree

4.3.1 Forming Shortest Path Tree

The delay-constrained shortest path tree (SPT) is aimed to construct a tree by combining the delay constraint shortest paths from source to each destination in the multicast group. For this purpose, the delay constrained shortest paths from source to each destination using HNN is obtained recursively and they are combined by ensuring that a link is appearing only once in the tree.

The shortest path problem is aimed to find the path $P(s,d_m)$ that has the minimum total cost $C(P(s,d_m))$ or maximize the residual bandwidth $RB(P(s,d_m))$. When the cost is to be minimized, the resulted path is the minimum-cost path and the resulted tree is minimum-cost shortest path tree. The residual bandwidth is a measure expressing the bandwidth availability for future use and is directly related to bandwidth utilization. When the bandwidth utilization is minimized, the resulted tree is regarded as minimum-bandwidth utilization tree or maximum-residual bandwidth tree. The formulation to obtain minimum-cost SPT and maximum-residual bandwidth SPT are discussed in 4.3.1.1 and 4.3.1.2 respectively.

4.3.1.1 Minimum-Cost Shortest Path Tree

The mathematical formulation to obtain delay-constrained minimum-cost path $P(s,d_m)$ is expressed as -

Minimize path $P(s,d_m)$ cost

$$C(P(s,d_m)) = \sum_{e \in P(s,d_m)} C(e) = \sum_{i,j \in P(s,d_m)} C_{ij} = C_{si} + C_{ij} + ... + C_{ld_m} \tag{4.14}$$

Subjected to

$$D(P(s,d_m)) = \sum_{e \in P(s,d_m)} D(e) = \sum_{i,j \in P(s,d_m)} D_{ij} \le \Delta \tag{4.15}$$

In the delay-constrained shortest path tree, the cost from source to each destination is minimum. The methodology to obtain it is explained through the following steps –

- Mapping shortest path
- Mapping delay bound in shortest path

- Dynamics of the Hopfield neural network
- Combining shortest paths to various destinations

Mapping Shortest Path

To meet the objective of minimum cost path, the energy function must favor states that correspond to valid paths between the specified source-destination pairs. Among these, it must favor the one which has the minimum cost. Therefore, for the *m*th destination in the multicast group d_m, the energy function E^m should be formulated to minimize the cost of the path $P(s, d_m)$ and considering the following constraints –

- Prevent the selection of non-existing links in the path.
- At the stable state, the output of the neurons should be either 1 or 0.
- There is only one route from source to destination in matrix V^m.

Therefore, the energy function E^m satisfying such requirements for the path $P(s, d_m)$ from source *s* to *m*th destination node d_m is given by –

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_5^m \tag{4.16}$$

Such that,

$$E_1^m = \frac{\mu_1}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} C_{xi} V_{xi}^m \tag{4.16a}$$

$$E_2^m = \frac{\mu_2}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} \varphi_{xi} V_{xi}^m \tag{4.16b}$$

$$E_3^m = \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{ix}^m \right\}^2 \tag{4.16c}$$

$$E_4^m = \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x \neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \tag{4.16d}$$

$$E_5^m = \frac{\mu_5}{2} (1 - V_{d_m s}^m) \tag{4.16e}$$

The energy term E_1^m is the cost term. The constant μ_1 is the weight to force the minimum cost of the path by accounting cost of existing links only. The energy term E_2^m will penalize the neurons representing the nonexisting links of the network. Therefore, the constant μ_2 is the weight to prevent the nonexistent links being included in the chosen path. However, the virtual link (d_m,s) will not be penalized. The energy term E_3^m is to ensure the construction of continuous path. The constant μ_3 is used to ensure that if a node has been entered in, it will also be exited i.e. number of incoming links to a node is equal to number of outgoing links for every node in the path. The energy term E_4^m will force the neurons output to be either 0 or 1. Therefore, the weight constant μ_4 is to force the state of neural network to converge to one of the corner of the hypercube defined by $V_{xi} \in \{0, 1\}$. The energy term E_5^m and the associated constant μ_5 enforce the construction of path from destination d_m to source s.

Mapping Delay Bound in Shortest Path

The end-to-end delay constraint is defined in eq. (4.15). A new energy term E_6^m, where end-to-end delay constraint is referred as penalty, is added to the energy E^m. This energy term E_6^m is therefore expressed as –

$$E_6^m = \frac{\mu_6}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m,s)}}^{N} D_{xi}^m V_{xi}^m \leq \Delta \tag{4.17}$$

Where, the constant μ_6 is the weight to enforce that the delay of the constructed path is less or equal to specified delay constraint.

This inequality constraint can be taken care by linear programming (LP) neuron (Kennedy and Chua, 1987). When the delay of the path is greater than delay constraint, the LP neuron penalizes all the neurons in the corresponding matrix. This neuron contributes only when constraint is violated. The transfer function h(z) of the LP type neuron is given as -

$$h(z) = \begin{cases} 0 & if\ z \leq 0 \\ z & otherwise \end{cases} \tag{4.18}$$

Where,

$$z = \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} D_{xi}^m V_{xi}^m - \Delta \tag{4.18a}$$

The delay constraint term using LP-type neurons can be defined as

$$E_6^m = H(z) = \int h(z) d(z) \tag{4.19}$$

The total energy function E^m for the path $P(s, d_m)$ from source to *m*th destination d_m including delay bound is therefore defined as -

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_5^m + E_{6,LP}^m \tag{4.20}$$

Using eq. (4.16) and eq. (4.19), the eq. (4.20) is expressed as -

$$\begin{aligned} E^m = {} & \frac{\mu_1}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} C_{xi} V_{xi}^m + \frac{\mu_2}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} \varphi_{xi} V_{xi}^m \\ & + \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{ix}^m \right\}^2 + \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x \neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \\ & + \frac{\mu_5}{2} (1 - V_{d_m s}^m) + \frac{\mu_6}{2} H(z) \end{aligned} \tag{4.21}$$

Dynamics of the Hopfield Neural Network

The dynamic equations of Hopfield neural network are expressed as eq. (4.1), eq. (4.9) and eq. (4.10). These equations are rewritten herewith to account the double subscript representation as –

$$V_{xi}^m = g_{xi}^m (U_{xi}^m) = \frac{1}{1 + e^{-\theta_{xi}^m U_{xi}^m}} \tag{4.22}$$

$$\frac{dU_{xi}^m}{dt} = \sum_{y=1}^{n} \sum_{\substack{j=1 \\ j \neq y}}^{n} T_{xi,yj} V_{yj}^m - \frac{U_{xi}^m}{\tau} + I_{xi} \tag{4.23}$$

$$\frac{dU_{xi}^m}{dt} = -\frac{U_{xi}^m}{\tau} - \frac{\partial E^m}{\partial V_{xi}^m} \tag{4.24}$$

Substituting the energy expression i.e. eq. (4.21) in the eq. (4.24) results into –

$$\begin{aligned}\frac{dU_{xi}^m}{dt} &= -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2} C_{xi}(1-\delta_{xd_m}\delta_{is}) \\ &\quad -\frac{\mu_2}{2}\varphi_{xi}(1-\delta_{xd_m}\delta_{is}) \\ &\quad -\mu_3\sum_{\substack{y=1\\y\neq x}}^{N}(V_{xy}^m - V_{yx}^m) + \mu_3\sum_{\substack{y=1\\y\neq i}}^{N}(V_{iy}^m - V_{yi}^m) \\ &\quad -\frac{\mu_4}{2}(1-2V_{xi}^m) + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \\ &\quad -\frac{\mu_6}{2} D_{xi}(1-\delta_{xd_m}\delta_{is})\,h(z)\end{aligned} \tag{4.25}$$

Where, the Kronecker delta δ_{ab} is defined as –

$$\delta_{ab} = \begin{cases} 1 & if\ \ a=b \\ 0 & otherwise \end{cases} \tag{4.25a}$$

By comparing the eq. (4.23) and (4.25), the connection weights $T_{xi,yj}$ and biases I_{xi} are derived as –

$$T_{xi,yj} = \mu_4\delta_{xy}\delta_{ij} - \mu_3\delta_{xy} - \mu_3\delta_{ij} + \mu_3\delta_{jx} + \mu_3\delta_{iy} \tag{4.26}$$

$$\begin{aligned} I_{xi} &= \left(-\frac{\mu_1}{2}C_{xi} - \frac{\mu_2}{2}\varphi_{xi} - \frac{\mu_6}{2}D_{xi}h(z)\right)(1-\delta_{xd_m}\delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \\ &= \begin{cases} \dfrac{\mu_5}{2} - \dfrac{\mu_4}{2} & if\,(x,i)=(d_m,s) \\ -\dfrac{\mu_1}{2}C_{xi} - \dfrac{\mu_2}{2}\varphi_{xi} - \dfrac{\mu_4}{2} - \dfrac{\mu_6}{2}D_{xi}h(z) & otherwise \end{cases}\end{aligned} \tag{4.27}$$

The terms with positive sign represent the excitatory connections and the terms with negative sign represent inhibitory connections among neurons.

Combining Shortest Paths to Various Destinations

The V^m obtained above represents the output matrix of the delay-constrained shortest path from source *s* to *m*th destination node d_m. The dimension of the V^m is (N×N). The final delay-constrained shortest path tree is obtained by the union of V^m for all the destinations $d_m \in \bar{M}$ in the multicast groups. Thus the matrix V corresponding to the tree is obtained as -

$$V = V^1 \cup V^2 \cup V^m \cup ....\cup V^M \tag{4.28}$$

With this implementation a link even if exists in multiple paths is counted only once in the tree matrix V. Similarly a node if participating into the formation of paths will be counted only once. This is explained with the help of an example graph shown as Fig. 4.2 where the source node s is '0' and the multicast group has two destination nodes as $\bar{M} = \{3,4\}$.

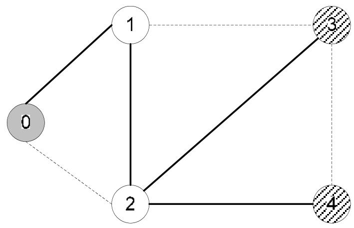


Fig. 4.2 An example graph (s = 0, $\bar{M} = \{3,4\}$)

The matrix V^1 corresponding to the path from source to first destination node *P(0,3)* is shown in Table 4.1(a). Similarly, the matrix V^2 for the path from source to second destination node *P(0,4)* is shown in Table 4.1(b). The resulting matrix V for the multicast tree *T(s,M)*, which is obtained after the union operation $(V^1 \cup V^2)$, is shown in Table 4.1(c). The links *e(0,1)* and *e(1,2)* are appearing in both the paths *P(0,3)* and *P(0,4)*. However, they are accounted only once in the resulting tree matrix V.

Table 4.1 Neuron output matrix V for the example graph (Fig. 4.2)

(a)
Matrix V^1 for *P(0,3)*

Node No	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	0	0
2	0	0	0	1	0
3	0	0	0	0	0
4	0	0	0	0	0

(b)
Matrix V^2 for *P(0,4)*

Node No	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	0	0
2	0	0	0	0	1
3	0	0	0	0	0
4	0	0	0	0	0

(c)
Tree matrix $V = V^1 \cup V^2$

Node No	0	1	2	3	4
0	0	1	0	0	0
1	0	0	1	0	0
2	0	0	0	1	1
3	0	0	0	0	0
4	0	0	0	0	0

4.3.1.2 Optimum Residual Bandwidth Shortest Path Tree

The residual bandwidth is a measure of the future use of the multicast tree and it should be maximized. With the formulation discussed in section 4.3.1.1, the HNN attempts the minimization of energy function. Therefore, the minimization of bandwidth utilization is first attempted and thereafter the residual bandwidth is calculated. During the computation, it is assumed that the capacities of all links are equal.

The bandwidth utilization $\lambda(e)$ of the link is the difference of link capacity $\phi(e)$ and available bandwidth $\alpha(e)$.

$$\lambda(e) = \phi(e) - \alpha(e) \tag{4.29}$$

The objective to minimize the bandwidth utilization for a path $P(s, d_m)$ is –

$$BU(P(s,d_m)) = \sum_{e \in P(s,d_m)} \lambda(e) = \sum_{i,j \in P(s,d_m)} \lambda_{ij} \tag{4.30}$$

This change will result into the change in the energy E^m, dynamic equations $\frac{dU_{xi}^m}{dt}$, and biases I_{xi}. Correspondingly, the equations are written as -

$$\begin{aligned} E^m = {} & \frac{\mu_1}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m,s)}}^{N} \lambda_{xi} V_{xi}^m + \frac{\mu_2}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m,s)}}^{N} \varphi_{xi} V_{xi}^m \\ & + \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{ix}^m \right\}^2 + \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x \neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \\ & + \frac{\mu_5}{2} (1 - V_{d_m s}^m) + \frac{\mu_6}{2} H(z) \end{aligned} \tag{4.31}$$

The derivative of the energy term E^m given as eq. 4.31 is expressed as-

$$\begin{aligned} \frac{dU_{xi}^m}{dt} = {} & -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2} \lambda_{xi} (1 - \delta_{xd_m} \delta_{is}) - \frac{\mu_2}{2} \varphi_{xi} (1 - \delta_{xd_m} \delta_{is}) \\ & -\mu_3 \sum_{\substack{y=1 \\ y \neq x}}^{N} (V_{xy}^m - V_{yx}^m) + \mu_3 \sum_{\substack{y=1 \\ y \neq i}}^{N} (V_{iy}^m - V_{yi}^m) - \frac{\mu_4}{2} (1 - 2V_{xi}^m) + \frac{\mu_5}{2} \delta_{xd_m} \delta_{is} \\ & - \frac{\mu_6}{2} D_{xi} (1 - \delta_{xd_m} \delta_{is}) \, h(z) \end{aligned} \tag{4.32}$$

$$I_{xi} = \left(-\frac{\mu_1}{2}\lambda_{xi} - \frac{\mu_2}{2}\varphi_{xi} - \frac{\mu_6}{2}D_{xi}h(z) \right)(1-\delta_{xd_m}\delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is}$$
$$= \begin{cases} \dfrac{\mu_5}{2} - \dfrac{\mu_4}{2} & if\,(x,i) = (d_m, s) \\ -\dfrac{\mu_1}{2}\lambda_{xi} - \dfrac{\mu_2}{2}\varphi_{xi} - \dfrac{\mu_4}{2} - \dfrac{\mu_6}{2}D_{xi}h(z) & otherwise \end{cases} \tag{4.33}$$

Having calculated the paths that are resulting into minimum bandwidth utilization from source to all the destinations in the multicast group, using eq. (4.29) to eq. (4.33) and the procedure given in section 4.3.1.1, a tree corresponding to minimum bandwidth utilization is resulted. This tree is also referred as maximum residual bandwidth tree. However, the percentage residual bandwidth is calculated after computing the sum of capacities of links in the tree as-

$$RB(T(s,M)) = 1 - \frac{BU(T(s,M))}{\sum_{i=1}^{N}\sum_{j=1}^{N}\phi_{ij}V_{ij}} \tag{4.34}$$

4.3.1.3 Shortest Path Tree Algorithm

The mapping and formulations to obtain the delay-constrained minimum cost SPT and delay-constrained maximum residual bandwidth SPT have been described in section 4.3.1.1 and 4.3.1.2 respectively. Both the formulations are identical except the computation of percentage residual bandwidth. Therefore, the flowchart to obtain the delay-constrained minimum-cost SPT is shown in Fig. 4.3. The steps of the algorithm are summarized as -

1. Read network data and set multicast group node (destination node) index $m=1$.
2. Set the destination node as d_m.
3. Initialize neurons input U^m randomly.
4. Compute neurons output V^m using sigmoid function i.e. eq. (4.22) and the initial energy $E^m(0)$ using eq. (4.21).
5. Set iteration counter $t=1$.
6. Use the fourth order Runge-Kutta method to integrate the eq. (4.25) to compute ΔU^m and update the inputs $U^m = U^m + \Delta U^m$.
7. Calculate neurons output V^m using eq. (4.22) and the energy $E^m(t)$ using eq. (4.21).
8. If $E^m(t)-E^m(t-1) > 0.0001$, then $t=t+1$, and go to step 6.

9. Calculate neuron output using hard limiter $V^m = \begin{cases} 1 \ \ for \ V^m > 0.5 \\ 0 \ \ for \ V^m < 0.5 \end{cases}$

10. If ($m < M$) then $m = m+1$, $t=0$ and go to step 2.

11. Combine V^1, V^2, ... V^m,..... V^M using eq. (4.28) to form shortest path tree and stop.

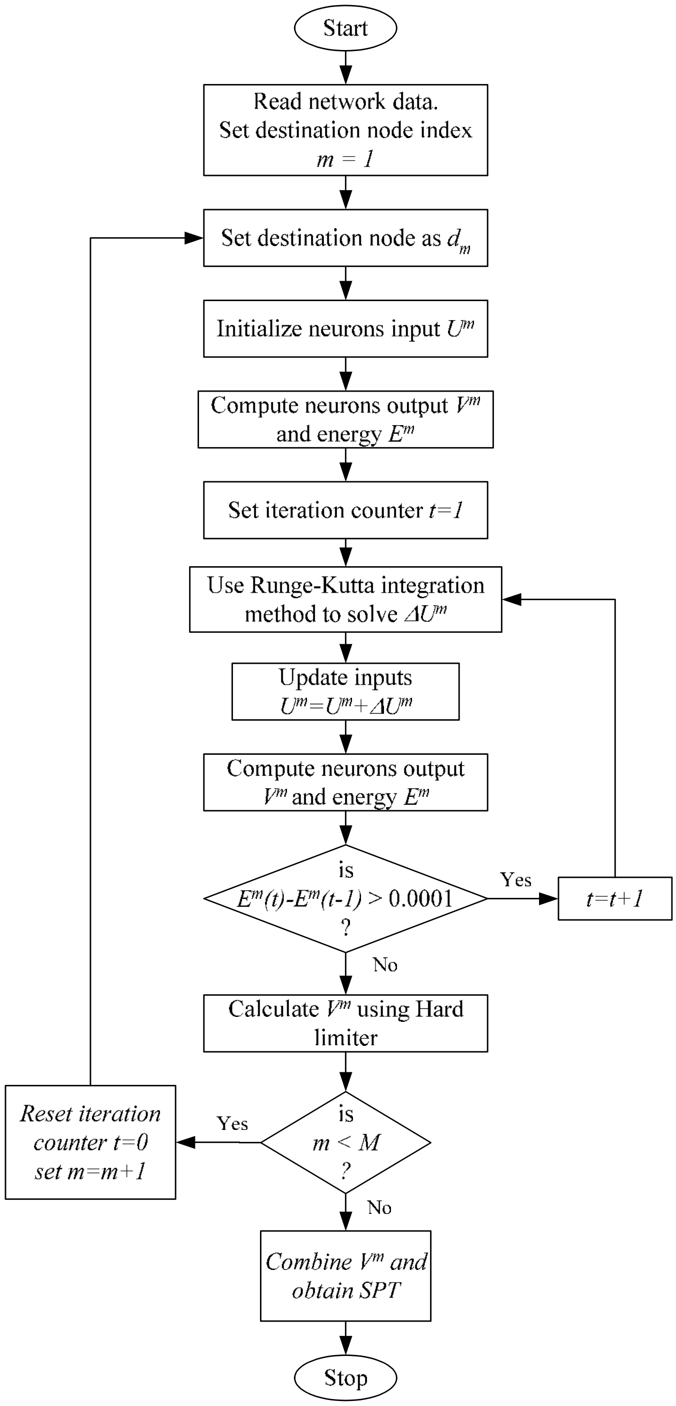


Fig. 4.3 The flowchart for computing delay-constrained shortest path tree

4.3.2 Forming Multicast Tree

In the delay-constrained minimum-cost shortest path tree formulation discussed in section 4.3.1, the shortest paths from source to various destinations are first formed and then the tree *T(s,M)* is constructed by combining the output matrices V^m for all the nodes in the multicast group $\bar{M}$. The delay constraint multicast tree is aimed to find a tree rooted at the source *s* and spanning to all the member of multicast group $\bar{M}$ such that the total cost *C(T(s,M))* of the tree is minimum or the residual bandwidth *RB(T(s,M))* is maximum and the delay from source to each destination is not greater than the required delay constraint Δ. When, the tree is obtained by minimizing the cost, it refers to delay-constrained minimum-cost multicast tree. The problem to obtain delay-constrained minimum-cost multicast tree is defined as -

$$\text{Minimize } C(T(s,M)) = \sum_{e \in T(s,M)} C(e) = \sum_{i,j \in T(s,M)} C_{ij} \tag{4.35}$$

Subjected to

End-to-end delay constraint

$$D(P(s,d_m)) = \sum_{e \in P(s,d_m)} D(e) = \sum_{i,j \in P(s,d_m)} D_{ij} \le \Delta \qquad \textit{for all } d_m \in \bar{M} \tag{4.36}$$

In forming the delay-constrained minimum cost MT, the total energy associated with the paths from source to destinations is used. For this, the common link cost should be accounted once or if the link cost has been accounted in previous path then the cost for subsequent paths should be reduced. The total energy *E* is the sum of the energy functions for delay constrained shortest paths for each destination. The total energy is expressed as -

$$E = \sum_{d_m \in \bar{M}} E^m \tag{4.37}$$

where, E^m is the energy function for the *m*th destination d_m in the multicast group as defined in the eq. (4.20).

$$E^m = E_1^m + E_2^m + E_3^m + E_4^m + E_5^m + E_{6,LP}^m \tag{4.38}$$

The all energy terms except the cost term remain unchanged. The cost E_1^m is therefore expressed as -

$$E_1^m = \sum_{\substack{x=1 \\ (x,i)\neq(d_m,s)}}^{N} \sum_{\substack{i=1 \\ i\neq x}}^{N} C_{xi} f_{xi}^m(V) V_{xi}^m \tag{4.39}$$

Where,

$$f_{xi}^m(V) = \frac{1}{1+\sum_{j=1, d_j \neq d_m, d_j \in \bar{M}}^{M} V_{xi}^j} \tag{4.39a}$$

The term $f_{xi}^m(V)$ helps in reducing the cost term E_1^m in proportions to the number of paths that are using the same link. If a link is selected by other paths, then the cost is reduced in proportion to the number of paths use the same link and therefore the link cost will be counted once.

The change in E_1^m is reflected as the change in energy E^m, dynamic equation $\frac{dU_{xi}^m}{dt}$, associated connections weights $T_{xi,yj}$ and bias terms I_{xi}. These terms are summarized as –

$$\begin{aligned} E^m = {} & \frac{\mu_1}{2} \sum_{\substack{x=1 \\ (x,i)\neq(d_m,s)}}^{N} \sum_{\substack{i=1 \\ i\neq x}}^{N} C_{xi} f_{xi}^m(V) V_{xi}^m + \frac{\mu_2}{2} \sum_{\substack{x=1 \\ (x,i)\neq(d_m,s)}}^{N} \sum_{\substack{i=1 \\ i\neq x}}^{N} \varphi_{xi} V_{xi}^m \\ & + \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i\neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i\neq x}}^{N} V_{ix}^m \right\}^2 + \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x\neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \\ & + \frac{\mu_5}{2}(1 - V_{d_m s}^m) + \frac{\mu_6}{2} H(z) \end{aligned} \tag{4.40}$$

$$\begin{aligned} \frac{dU_{xi}^m}{dt} = {} & -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2} C_{xi} f_{xi}^m(V)(1 - \delta_{xd_m}\delta_{is}) \\ & - \frac{\mu_2}{2} \varphi_{xi}(1 - \delta_{xd_m}\delta_{is}) \\ & - \mu_3 \sum_{\substack{y=1 \\ y\neq x}}^{N} (V_{xy}^m - V_{yx}^m) + \mu_3 \sum_{\substack{y=1 \\ y\neq i}}^{N} (V_{iy}^m - V_{yi}^m) \\ & - \frac{\mu_4}{2}(1 - 2V_{xi}^m) + \frac{\mu_5}{2} \delta_{xd_m}\delta_{is} \\ & - \frac{\mu_6}{2} D_{xi}(1 - \delta_{xd_m}\delta_{is}) h(z) \end{aligned} \tag{4.41}$$

$$T_{xi,yj} = \mu_4 \delta_{xy}\delta_{ij} - \mu_3\delta_{xy} - \mu_3\delta_{ij} + \mu_3\delta_{jx} + \mu_3\delta_{iy} \tag{4.42}$$

$$\begin{aligned} I_{xi} &= \left(-\frac{\mu_1}{2} C_{xi} f_{xi}^m(V) - \frac{\mu_2}{2}\varphi_{xi} - \frac{\mu_6}{2} D_{xi} h(z) \right)(1-\delta_{xd_m}\delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \\ &= \begin{cases} \dfrac{\mu_5}{2} - \dfrac{\mu_4}{2} & if\,(x,i) = (d_m, s) \\ -\dfrac{\mu_1}{2} C_{xi} f_{xi}^m(V) - \dfrac{\mu_2}{2}\varphi_{xi} - \dfrac{\mu_4}{2} - \dfrac{\mu_6}{2} D_{xi} h(z) & otherwise \end{cases} \end{aligned} \tag{4.43}$$

For maximizing the residual bandwidth, the minimization of bandwidth utilization of the tree is first attempted and thereafter the residual bandwidth is calculated. During the computation, it is assumed that the capacities of all links are equal. The objective pertaining to minimization of the bandwidth utilization by tree is -

$$BU(T(s,M)) = \sum_{e \in T(s,M)} \lambda(e) = \sum_{i,j \in T(s,M)} \lambda_{ij} \tag{4.44}$$

This change will result into the change in the energy E^m, dynamic equations $\frac{dU_{xi}^m}{dt}$, and biases I_{xi}. Correspondingly, the equations are written as -

$$\begin{aligned} E^m &= \frac{\mu_1}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m,s)}}^{N} \lambda_{xi} f_{xi}^m(V) V_{xi}^m + \frac{\mu_2}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m,s)}}^{N} \varphi_{xi} V_{xi}^m \\ &+ \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{ix}^m \right\}^2 + \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x \neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \\ &+ \frac{\mu_5}{2}(1 - V_{d_m s}^m) + \frac{\mu_6}{2} H(z) \end{aligned} \tag{4.45}$$

$$\begin{aligned} \frac{dU_{xi}^m}{dt} &= -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2} \lambda_{xi} f_{xi}^m(V)(1-\delta_{xd_m}\delta_{is}) \\ &- \frac{\mu_2}{2}\varphi_{xi}(1-\delta_{xd_m}\delta_{is}) \\ &- \mu_3 \sum_{\substack{y=1 \\ y \neq x}}^{N} (V_{xy}^m - V_{yx}^m) + \mu_3 \sum_{\substack{y=1 \\ y \neq i}}^{N} (V_{iy}^m - V_{yi}^m) \\ &- \frac{\mu_4}{2}(1 - 2V_{xi}^m) + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \\ &- \frac{\mu_6}{2} D_{xi}(1-\delta_{xd_m}\delta_{is})\, h(z) \end{aligned} \tag{4.46}$$

$$I_{xi} = \left(-\frac{\mu_1}{2}\lambda_{xi} f_{xi}^m(V) - \frac{\mu_2}{2}\varphi_{xi} - \frac{\mu_6}{2} D_{xi} h(z) \right)(1-\delta_{xd_m}\delta_{is}) - \frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is}$$

$$= \begin{cases} \frac{\mu_5}{2} - \frac{\mu_4}{2} & if\,(x,i) = (d_m, s) \\ -\frac{\mu_1}{2} C_{xi} f_{xi}^m(V) - \frac{\mu_2}{2}\varphi_{xi} - \frac{\mu_4}{2} - \frac{\mu_6}{2} D_{xi} h(z) & otherwise \end{cases} \quad (4.47)$$

The resulting tree corresponding to minimum bandwidth utilization is also referred as maximum residual bandwidth tree. The percentage residual bandwidth of the tree is calculated using eq. (4.34).

4.3.2.1Multicast Tree Algorithm

The mapping and formulations to obtain the delay-constrained minimum cost MT and delay-constrained maximum residual bandwidth MT have been described in section 4.3.2. Both the formulations are identical except the computation of percentage residual bandwidth. Therefore, the flowchart to obtain the delay-constrained minimum-cost MT is presented in Fig. 4.4. The steps of the algorithm are summarized as -

1. Read network data and initialize neurons input U randomly.
2. Compute neurons output V^m using sigmoid function i.e. eq. (4.22) and the initial energy $E^m(0)$ using eq. (4.37).
3. Set iteration counter $t=1$.
4. Use fourth order Runge-Kutta method to solve eq. (4.41) to compute ΔU and update the inputs $U=U+\Delta U$.
5. Calculate neurons output V using eq. (4.22).
6. Compute the energy value E using eq. (4.37).
7. If $E(t)-E(t-1) > 0.0001$, then $t=t+1$ and go to step 4.
8. Calculate neuron output using hard limiter $V = \begin{cases} 1 \; for \; V > 0.5 \\ 0 \; for \; V < 0.5 \end{cases}$ and stop.

4.3.3 Forming Multiobjective Multicast Tree

The delay constraint multiobjective multicast tree is aimed to find a tree rooted at the source s and spanning to all the member of multicast group M such by optimizing both the cost $C(T(s,M))$ and residual bandwidth $RB(T(s,M))$ associated with the links of the tree and the delay from source to each destination is not greater than the required delay constraint Δ. For minimization type of energy function, the optimization of residual bandwidth is attempted in terms of bandwidth utilization. The problem to obtain delay-constrained multiobjective multicast tree is therefore defined as-

minimize

$$C(T(s,M)) = \sum_{e \in T(s,M)} C(e) = \sum_{i,j \in T(s,M)} C_{ij}$$

$$BU(T(s,M)) = \sum_{e \in T(s,M)} \lambda(e) = \sum_{i,j \in T(s,M)} \lambda_{ij} \tag{4.48}$$

subjected to

$$D(P(s,d_m)) = \sum_{e \in P(s,d_m)} D(e) = \sum_{i,j \in P(s,d_m)} D_{ij} \le \Delta \quad \text{for } d_m \in \bar{M}$$

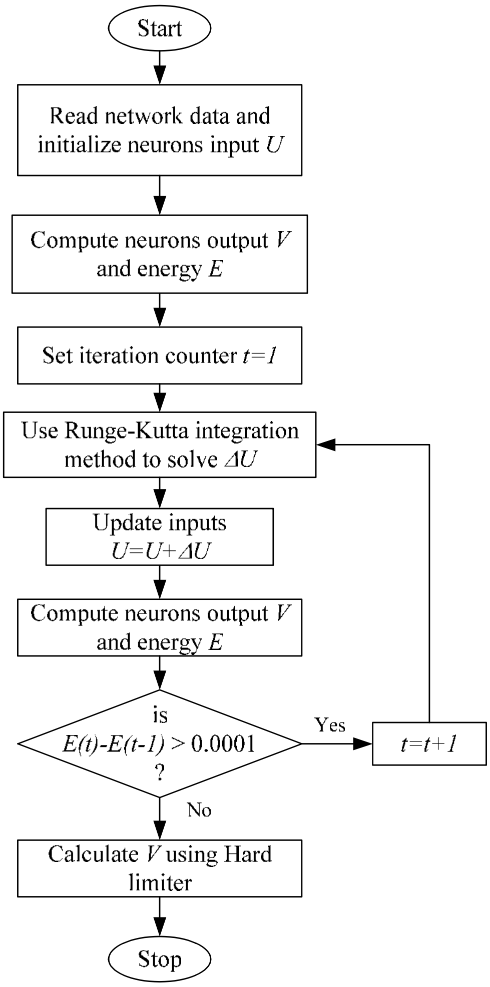


Fig. 4.4 The flowchart for computing delay-constrained multicast tree

The formulation of delay constrained multicast tree has been discussed in section 4.3.2 and the formulation of delay constrained multiobjective multicast tree is on the similar lines. The HNN works on the minimization of energy, the total energy and the associated dynamic equations are modified to account the effects of both the objectives using weighted combination of the energy terms. For the purpose, the cost term E_1^m is modified and a new term E_7^m corresponding to bandwidth is introduced. The energy is therefore expressed as -

$$E = \sum_{m \in M} E^m = \sum_{m \in M} (E_1^m + E_2^m + E_3^m + E_4^m + E_5^m + E_{6,LP}^m + E_7^m) \qquad (4.49)$$

Only the terms associated with cost E_1^m will change and the new term E_7^m associated with bandwidth utilization is introduced. These terms E_1^m and E_7^m are expressed as -

$$E_1^m = W_1 \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} C_{xi} f_{xi}^m(V) V_{xi}^m \qquad (4.49a)$$

$$E_7^m = W_2 \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} \lambda_{xi} f_{xi}^m(V) V_{xi}^m \qquad (4.49b)$$

The other energy terms have been already explained in section 4.3.2. Combining various energy terms, the E^m is resulted as -

$$\begin{aligned} E^m = {} & \frac{\mu_1}{2} W_1 \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} C_{xi} f_{xi}^m(V) V_{xi}^m + \frac{\mu_2}{2} \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} \varphi_{xi} V_{xi}^m \\ & + \frac{\mu_3}{2} \sum_{x=1}^{N} \left\{ \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{xi}^m - \sum_{\substack{i=1 \\ i \neq x}}^{N} V_{ix}^m \right\}^2 + \frac{\mu_4}{2} \sum_{i=1}^{N} \sum_{\substack{x=1 \\ x \neq i}}^{N} V_{xi}^m (1 - V_{xi}^m) \\ & + \frac{\mu_5}{2} (1 - V_{d_m s}^m) + \frac{\mu_6}{2} H(z) + \frac{\mu_7}{2} W_2 \sum_{x=1}^{N} \sum_{\substack{i=1 \\ i \neq x \\ (x,i) \neq (d_m, s)}}^{N} \lambda_{xi} f_{xi}^m(V) V_{xi}^m \end{aligned} \qquad (4.50)$$

Due to the change in the energy terms, the derivate of the input potential to the neurons is obtained as -

$$\frac{dU_{xi}^m}{dt} = -\frac{U_{xi}^m}{\tau} - \frac{\mu_1}{2} W_1 C_{xi} f_{xi}^m(V)(1-\delta_{xd_m}\delta_{is}) - \frac{\mu_2}{2}\varphi_{xi}(1-\delta_{xd_m}\delta_{is})$$
$$-\mu_3 \sum_{\substack{y=1 \\ y\neq x}}^{N} (V_{xy}^m - V_{yx}^m) + \mu_3 \sum_{\substack{y=1 \\ y\neq i}}^{N} (V_{iy}^m - V_{yi}^m) - \frac{\mu_4}{2}(1-2V_{xi}^m) + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \tag{4.51}$$
$$-\frac{\mu_6}{2} D_{xi}(1-\delta_{xd_m}\delta_{is})\, h(z) - \frac{\mu_7}{2} W_2 \lambda_{xi} f_{xi}^m(V)(1-\delta_{xd_m}\delta_{is})$$

Comparing eq. (4.51) with eq. (4.23), the connection weights and bias currents to the neurons are obtained as -

$$T_{xi,yj} = \mu_4 \delta_{xy}\delta_{ij} - \mu_3\delta_{xy} - \mu_3\delta_{ij} + \mu_3\delta_{jx} + \mu_3\delta_{iy}$$

$$I_{xi} = \left(-\frac{\mu_1}{2} W_1 C_{xi} f_{xi}^m(V) - \frac{\mu_2}{2}\rho_{xi} - \frac{\mu_6}{2} D_{xi} h(z) - \frac{\mu_7}{2} W_2 \lambda_{xi} f_{xi}^m(V) \right)(1-\delta_{xd_m}\delta_{is})$$
$$-\frac{\mu_4}{2} + \frac{\mu_5}{2}\delta_{xd_m}\delta_{is} \tag{4.52}$$
$$= \begin{cases} \dfrac{\mu_5}{2} - \dfrac{\mu_4}{2} & if\,(x,i) = (d_m, s) \\ -\dfrac{\mu_2}{2}\varphi_{xi} - \dfrac{\mu_4}{2} - \dfrac{\mu_6}{2} D_{xi} h(z) - \left(\dfrac{\mu_1}{2} W_1 C_{xi} + \dfrac{\mu_7}{2} W_2 \lambda_{xi} \right) f_{xi}^m(V) & otherwise \end{cases}$$

4.3.3.1 Multiobjective Multicast Tree Algorithm

The mapping and formulations to obtain the delay-constrained multiobjective multicast tree (MMT) has been described in section 4.3.3. Depending on the values of weights, one Pareto solution is obtained by solving the formulation discussed in 4.3.3. To obtain, the Pareto optimal front or set of Pareto-optimal solutions, different weights combinations are selected and simulation is run for each combination of the weights. After obtaining number of different Pareto-optimal solutions, the best-compromise multicast tree is obtained using the fuzzy-cardinal approach discussed in section 3.2.2. The above formulation to obtain best-compromise delay-constrained MMT is implemented through the flowchart presented in Fig. 4.5 and the steps of the algorithm are summarized as -

1. Read network data and initialize the weights as $W_1 = 1.0$, $W_2 = 0.0$, change in weight $\Delta W = 0.1$.
2. Initialize neurons input U randomly.
3. Compute neurons output V using sigmoid function i.e. eq. (4.22) and the initial energy $E(0)$ using eq. (4.49).

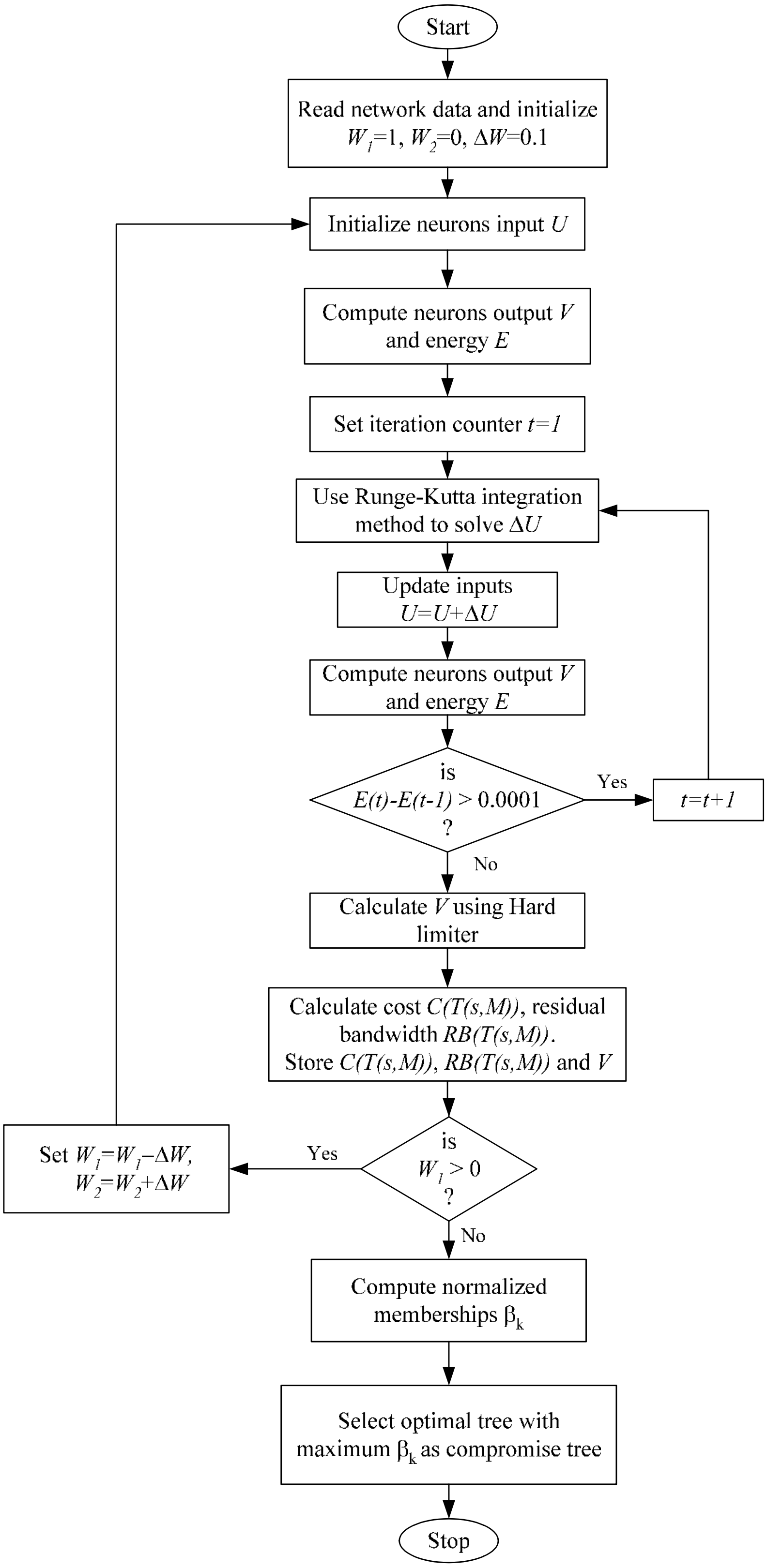


Fig. 4.5 The flowchart for computing best compromised delay-constrained multicast tree

4. Set iteration counter $t=1$.
5. Use fourth order Runge-Kutta method to solve eq. (4.51) to compute ΔU and update the inputs $U=U+\Delta U$.
6. Calculate neurons output V using eq. (4.22).
7. Compute the energy value E using eq. (4.49).
8. If $E(t)-E(t-1) > 0.0001$, then $t=t+1$ and go to step 5.
9. Calculate neuron output using hard limiter $V = \begin{cases} 1 \;\; for \; V > 0.5 \\ 0 \;\; for \; V < 0.5 \end{cases}$ and store the Pareto-solution.
10. if ($W_1>0$) then set weights as $W_1=W_1-\Delta W$, $W_2=W_2+\Delta W$ and goto step 2.
11. Calculate the normalize membership function β_k for each Pareto solution and select the solution with maximum β_k as the best compromise MMT.
12. Stop.

4.4 RESULTS AND DISCUSSION

The HNN has been used for solving unconstrained and delay constrained routing optimization problems namely cost minimization, residual bandwidth optimization and cost-residual bandwidth optimization. For the single objective cost minimization and residual bandwidth optimization, the tree has been constructed using shortest path tree (SPT) and multicast tree (MT) approaches whereas the problem of optimum cost-residual bandwidth is solved using multicast tree approach.

The neurons inputs are initialized as $-0.0001 \le U_{xi} \le 0.0001$ to prevent the neural network from adopting an undesirable state. The states of the neural network are simulated to obtain the output of the neurons V_{xi}^{m} for multicast group. The neural network dynamic equations are for various formulations are solved using fourth order Runge-Kutta method. For shortest path tree neuron outputs are recursively calculated for all values of destinations in the multicast group. The convergence of the algorithm and the outcome depend on the appropriate selection of HNN model coefficients i.e. μ_i. These coefficients are selected using the following criteria (Pornavalai et al., 1995; Ali and Kamoun, 1993),

$$2\mu_3 - \mu_4 > 0,\ \mu_2 = \mu_5,\ \ \mu_1 << 2\mu_3 \,/\, C_{xi}^{\max},\ \mu_1 < \mu_4,\ \mu_6 > \mu_1 \ \text{and}\ \mu_5 > \mu_1\, C_{xi}^{\max}.$$

The $C_{xi}^{\max}$ is the maximum cost associated with any of the link in the network. With this criteria, the following values of coefficient are selected for the simulation studies on 24-node USIP Backbone network given as Fig. A1 in the Appendix-A. The network parameters are also summarized in the Appendix A.

$$\mu_1 = 100,\ \mu_2 = 5000,\ \mu_3 = 3500,\ \mu_4 = 250,\ \mu_5 = 5000,\ \mu_6 = 250$$

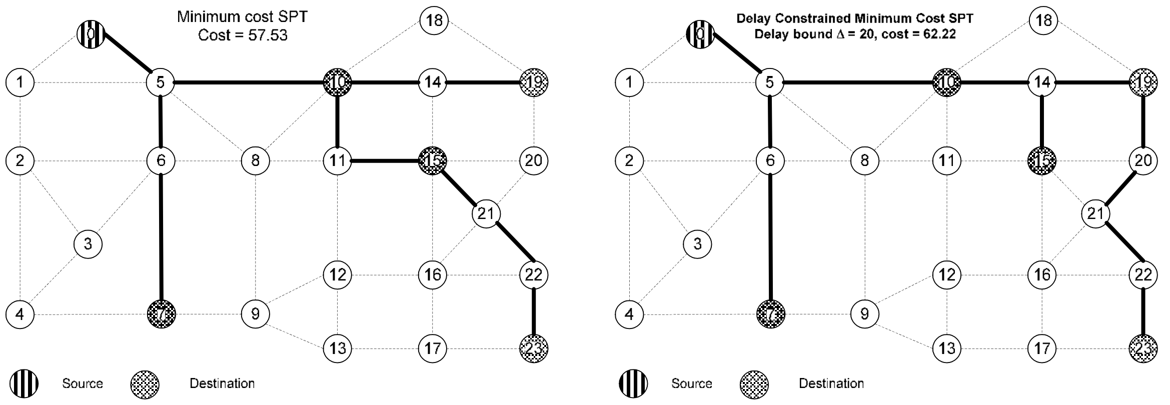
With the selection of the coefficients as above, high value of μ_2 is imposing a penalty for non-existing link being included, high μ_3 ensures the construction of a continuous path, high value of μ_5 ensures the construction of virtual link from destination to source. The higher value of μ_6 ensures the constraint satisfaction. When the difference between the energy of the system in two consecutive iterations is less than 0.00002, it is assumed that stable state is attained. The experimental investigation has been attempted to obtain optimum multicast trees for different sets of source and destinations on undirected weighted 24-node USIP network. The results are presented to form the following trees-

- Shortest Path Tree
- Multicast Tree
- Multiobjective Multicast Tree

4.4.1 Shortest Path Tree Formulation

The shortest path tree (SPT) formulation works on the concept of forming shortest paths and then the recursively obtained output matrix *V* are combined to form the tree. With the SPT algorithm, the minimum cost tree and delay-constrained minimum cost tree with source node as '0' and destinations nodes as {7, 10, 15, 19, 23} are shown in Fig. 4.6(a) and Fig. 4.6(b) respectively. For the delay-constrained tree formulation, the specified maximum delay is taken as 20. The resulting cost for minimum cost tree is 57.53 and for the delay-constrained minimum cost tree its value is 62.22.

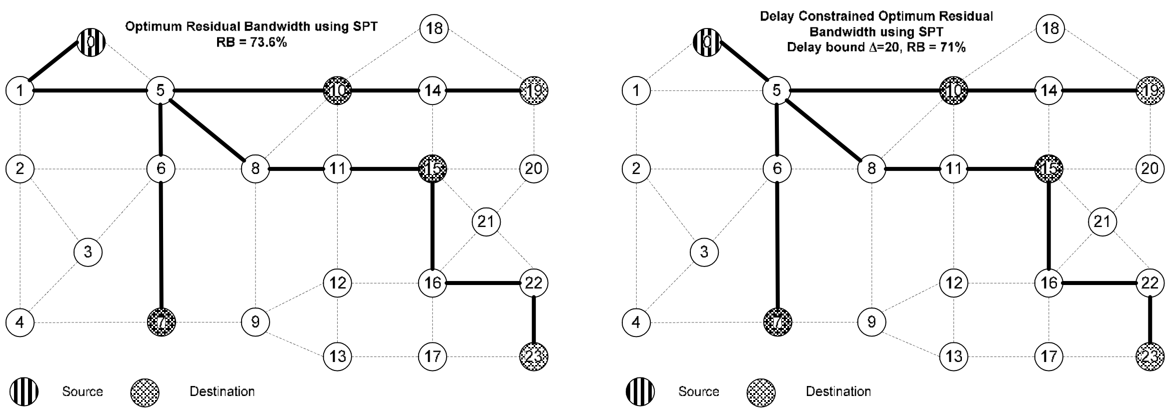
The resulted shortest path trees are different for cost minimization and for delay-constrained cost minimization. As expected, the cost of the unconstrained tree is lower than the cost of the delay bound tree. With the inclusion of delay as constraint, the model coefficients give priority to satisfy delay and form higher energy. With this the algorithm is forced to select higher cost path and thereby resulting into higher cost of the tree.



(a) minimum cost tree (b) delay constrained minimum cost tree

Fig. 4.6 The resulted trees for cost minimization using shortest path tree formulation

Using the SPT algorithm, the optimum residual-bandwidth tree and delay-constrained optimum residual-bandwidth tree for source node as ‘0’ and destinations nodes as {7,10,15,19,23} are shown in Fig. 4.7(a) and Fig. 4.7(b) respectively. The specified delay bound Δ is considered as 20. The optimum residual bandwidth values are obtained as 73.6% and 71.0% respectively.



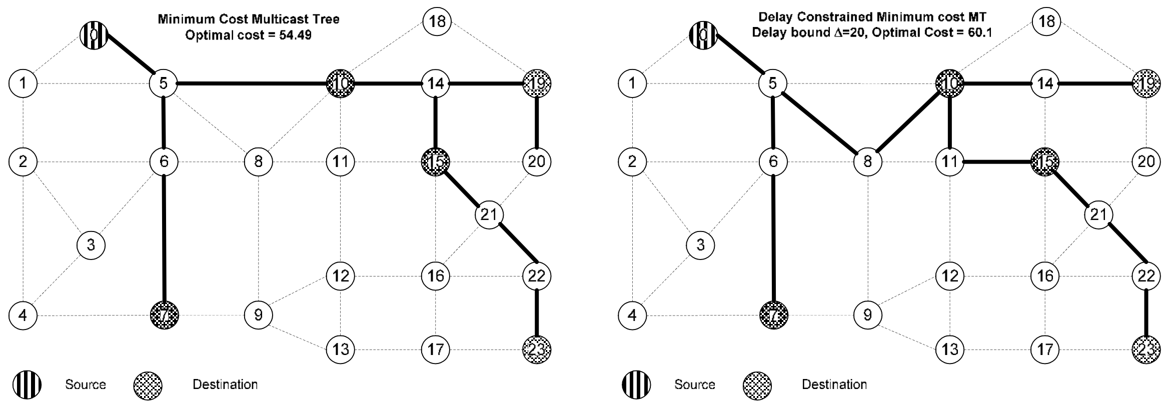
(a) optimum residual bandwidth (b) delay-constrained optimum residual bandwidth

Fig. 4.7 The resulted trees for residual-bandwidth maximization using shortest path tree formulation

The resulted value of residual bandwidth (RB) for optimum residual-bandwidth tree is higher than its value for delay-bound optimum residual-bandwidth tree. As explained, the HNN model coefficients force the algorithm to satisfy the delay constraint at the cost of residual bandwidth and thereby resulting into tree of lower residual-bandwidth. This is contrary to the minimum cost optimization where the inclusion of delay bound is resulting into higher cost. This is because the cost is being minimized and the residual bandwidth is being maximized.

4.4.2 Multicast Tree Formulation

The multicast tree formulation yields the output matrix *V* directly. The formulation is focused to yield the tree by directly giving the importance to tree measures such as tree cost *C(T(s,M))* or residual bandwidth *RB(T(s,M))*. The resulted trees using multicast tree formulation are shown in Fig. 4.8(a) and Fig. 4.8(b) for cost minimization and delay-bound cost minimization. These trees are obtained while keeping the source node and multicast group unchanged as specified in Fig. 4.6. The costs are resulted as 54.49 and 60.1 respectively. As explained earlier, the inclusion of delay bound forces the algorithm to include higher cost path and therefore the cost of delay-bound tree is higher.

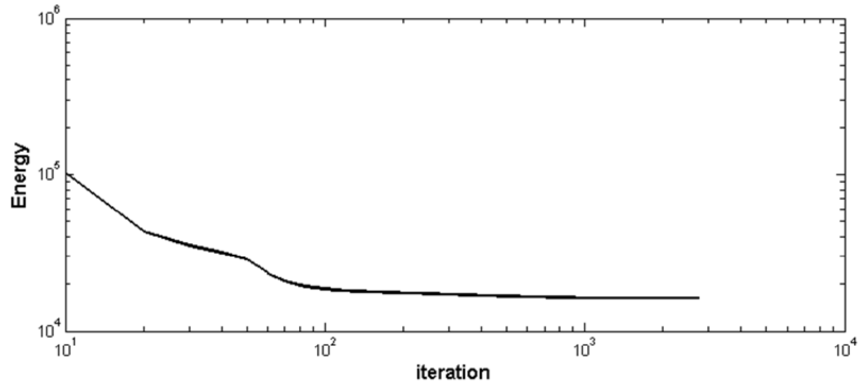


(a) minimum cost tree (b) delay constrained minimum cost tree

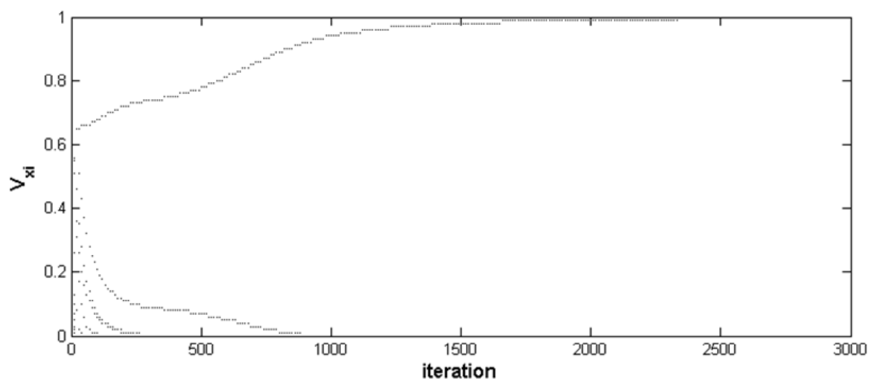
Fig. 4.8 The resulted trees for cost minimization using multicast tree formulation

The cost-minimization using MT formulation and SPT formulation is resulting the cost values as 54.49 and 57.63 respectively. For the delay-bound cost minimization the resulted optimum costs are 60.1 and 62.22 through MT and SPT formulations respectively. For both cost-minimization and delay-bound cost minimization, the MT arrangement is resulting into lower costs compared to the respective costs resulted with SPT formulation. This is because in the SPT formulation, the consideration is for the individual path $P(s,d_i)$ metric whereas in the MT formulation, the consideration is for the tree *T(s,M)* metric. The iterations expanded in forming the tree by both SPT and formulations vary between 2000-5000.

As known, the Hopfield neural network works on the minimization of the energy with the progress of iterations. During this iterative process, the neurons outputs are also getting stabilize. This process is shown in Fig. 4.9 for delay-constrained cost-minimization using multicast tree formulation. The source-multicast group and delay bound are kept same as given in Fig. 4.8(b). It is taking around 2500 iterations.



(a) monotonically decreasing energy of HNN



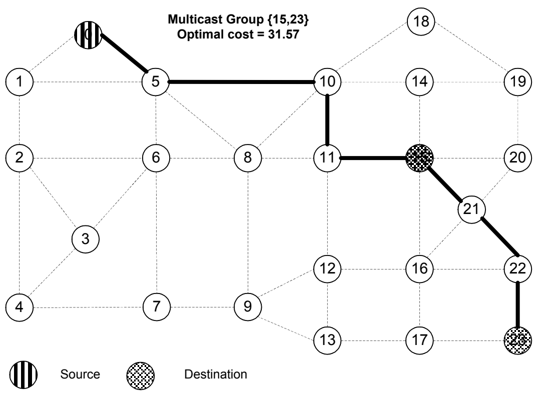
(b) Settling of neural network dynamics

Fig. 4.9 The variation in energy and neural network dynamics for delay-bound cost optimization

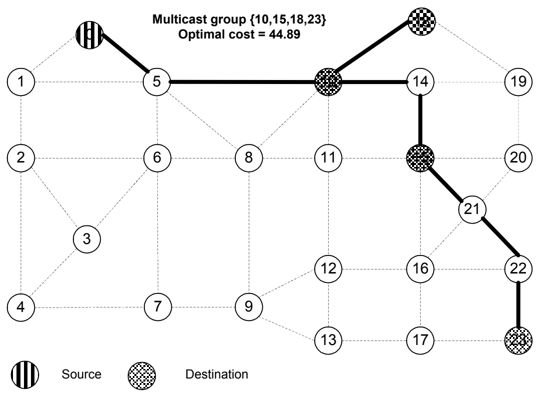
The investigation with MT formulation has been extended to obtain the delay-constrained minimum cost multicast tree for different multicast groups i.e. destination nodes. These results are summarized in Table 4.2 for delay bound Δ as 20. Correspondingly the optimal trees are shown in Fig. 4.10. It is observed that the cost of trees increases with the increasing size of multicast group. This is convincing as the size of tree is increasing and so the cost. However, the cost of multicast tree depends on the relative position of source and destinations and network parameters. The convergence is attained in 700-1100 iterations.

Table 4.2 Summary of delay-constrained minimum cost optimization using multicast tree formulation for different multicast groups (source s={0}, delay bound Δ = 20)

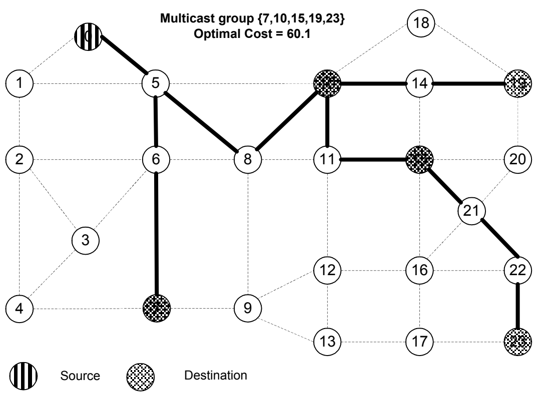
Multicast group	Cost	source to destination path $P(s,d_m)$
{15,23}	31.566717	0->5->10->11->15 0->5->10->11->15->21->22->23
{10,15,18,23}	44.892200	0->5->10 0->5->10->14->15 0->5->10->18 0->5->10->14->15->21->22->23
{7,10,15,19,23}	60.1	0->5->6->7 0->5->10 0->5->8->10->11->15 0->5->8->10->14->19 0->5->8->10->11->15->21->22->23
{7,10,15,18,20,23}	63.047867	0->5->6->7 0->5->10 0->5->10->14->15 0->5->10->18 0->5->10->14->19->20 0->5->10->21->22->23



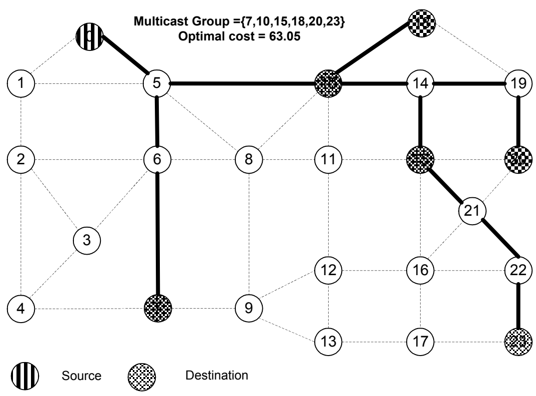
(a) multicast group {15,23}



(b) multicast group {10,15,18,23}



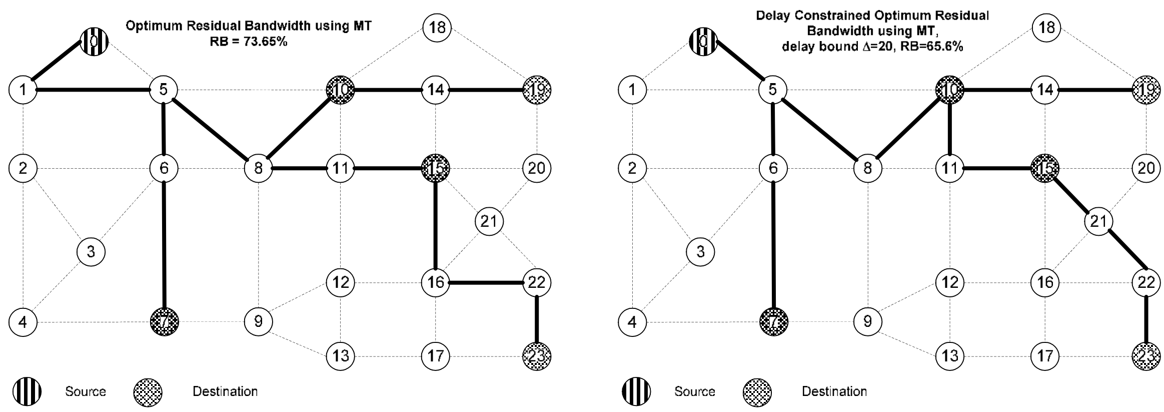
(c) multicast group {7,10,15,19,23}



(d) multicast group {7,10,15,18,20,23}

Fig. 4.10 Multicast trees for delay-constrained minimum cost optimization

The optimization of residual bandwidth is simulated using multicast tree formulation. The resulted multicast trees for optimum residual bandwidth and delay-bound residual bandwidth are shown in Fig. 4.11 for source node as ‘0’, multicast group as {7,10,15,19,23} and delay bound Δ as 20. Corresponding to the delay bound residual bandwidth optimization, the energy and HNN dynamics are shown in Fig. 4.12. The resulted optimum values of RB are 73.65% and 65.6% respectively for optimum residual bandwidth and delay bound optimum residual bandwidth.



(a) Optimum RB tree (b) delay constrained optimum RB tree

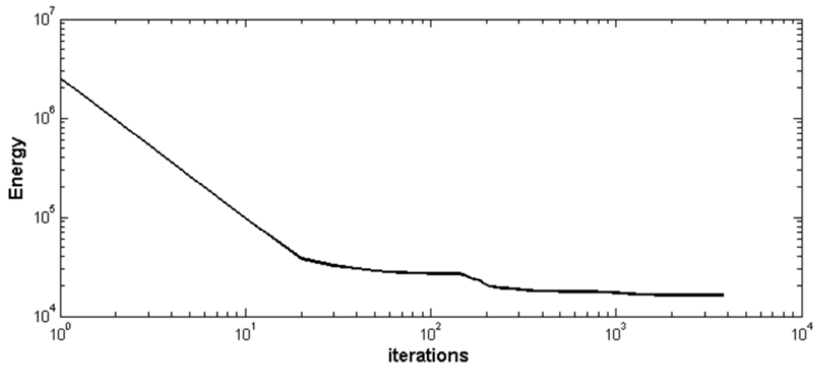
Fig. 4.11 The resulted trees for residual-bandwidth maximization using multicast tree formulation

Without the delay bound, the optimum residual-bandwidth is close to the value obtained using SPT, whereas, with delay bound the value is lower than the value resulted with SPT. The resulted RB of the respective optimum residual-bandwidth tree is higher than the RB of the respective delay bound optimum residual-bandwidth tree. The HNN dynamics as shown in Fig. 4.13 stabilizes with the progress of iterations. It is taking nearly 3800 iterations for the convergence.

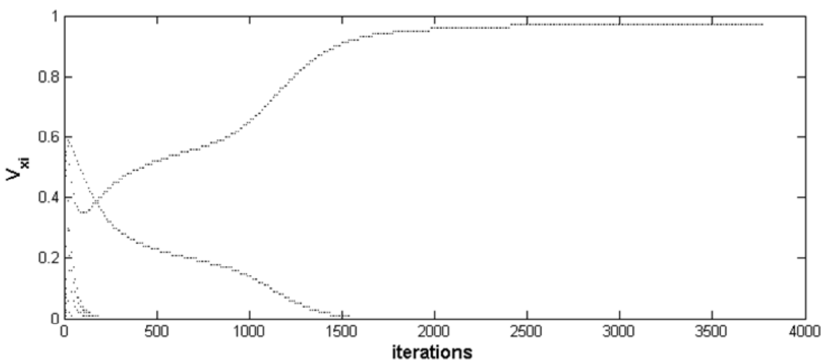
The study with MT formulation has been extended to obtain optimum residual bandwidth trees for different multicast groups and the results are summarized in Table 4.3. The correspondingly trees are shown in Fig. 4.13. It is observed that the optimal residual bandwidth decreases slightly with the increase in multicast group size.

Table 4.3 Summary of residual bandwidth optimization using multicast tree formulation for different multicast groups

Multicast group	Optimum residual bandwidth	source to destination path $P(s,d_m)$
{15,23}	77.3556	0->1->5->8->11->15 0->1->5->8->11->15->16->22->23
{10,15,18,23}	73.7403	0->1->5->8->10 0->1->5->8->11->15 0->1->5->8->10->18 0->1->5->8->11->15->16->22->23
{7,10,15,19,23}	73.6572	0->1->5->6->7 0->1->5->8->10 0->1->5->8->11->15 0->1->5->8->10->14->19 0->1->5->8->11->15->16->22->23
{7,10,15,18,20,23}	70.8949	0->1->5->6->7 0->1->5->8->10 0->1->5->8->11->15 0->1->5->8->10->18 0->1->5->8->11->15->20 0->1->5->8->11->15->16->22->23

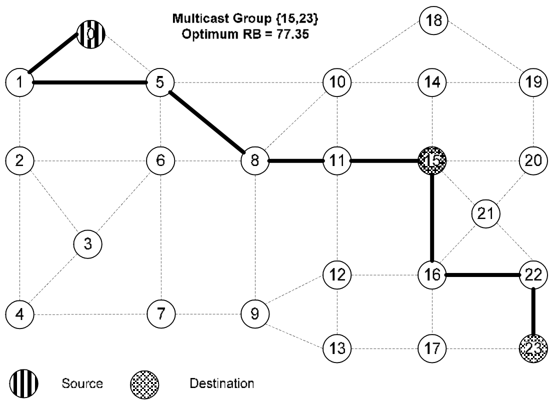


(a) monotonically decreasing energy of HNN

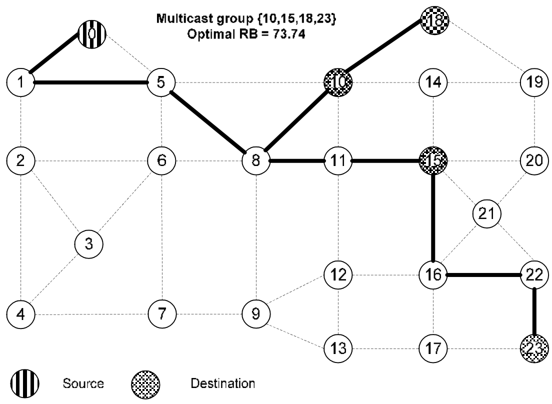


(b) settling of HNN dynamics

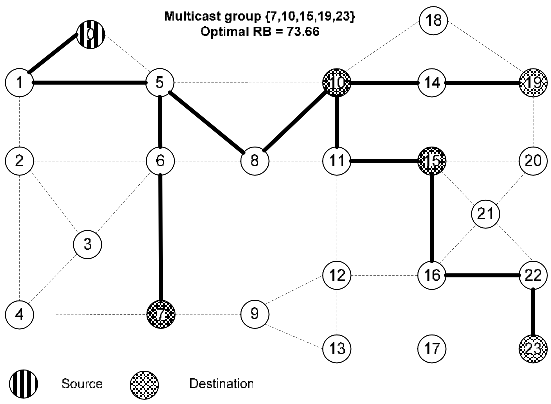
Fig. 4.12 The variation in energy and neural network dynamics for delay-constrained residual bandwidth optimization



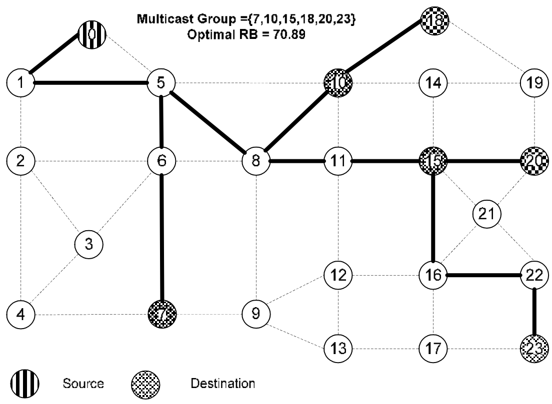
(a) multicast group {15,23}



(b) multicast group {10,15,18,23}



(c) multicast group {7,10,15,19,23}



(d) multicast group {7,10,15,18,20,23}

Fig. 4.13 Multicast trees for different multicast groups for optimum residual bandwidth

4.4.3 Multiobjective Multicast Tree Formulation

The multiobjetive multicast tree formulation has been studied for optimum cost-residual bandwidth optimization. The weighted sum approach has been used for forming the energy function and the optimization is carried out using the MT formulation. In forming the weighted single objective, a new HNN coefficient is to be associated with the bandwidth term in the energy function. The choice of the coefficients associated with cost term μ_1 and bandwidth term μ_7 will influence the result. To decide the values of μ_1 and μ_7, an experiment is conducted to study the effect of μ_1 and μ_7 for multicast group {10, 17, 23} with weights W_1 and W_2 as 0.5. These investigations are summarized in Tables 4.4, Table 4.5 and Table 4.6 and correspondingly the results are presented in Fig. 4.14, 4.15, 4.16 respectively. Table 4.4 and Fig. 4.14 correspond to equal values of μ_1 and μ_7 varying between 50 to 400 with the step size of 50. Other parameters are kept unchanged.

Table 4.4 : The optimal cost and residual bandwidths for equal values of μ_1 & μ_7

Value of (μ_1=μ_7)	Cost	Residual Bandwidth
50	86.952	69.5095
100	71.6366	68.8623
150	70.1899	70.4003
200	57.9078	67.8982
250	52.4003	72.0405
300	52.4003	72.0405
350	52.4003	72.0405
400	52.4003	72.0405

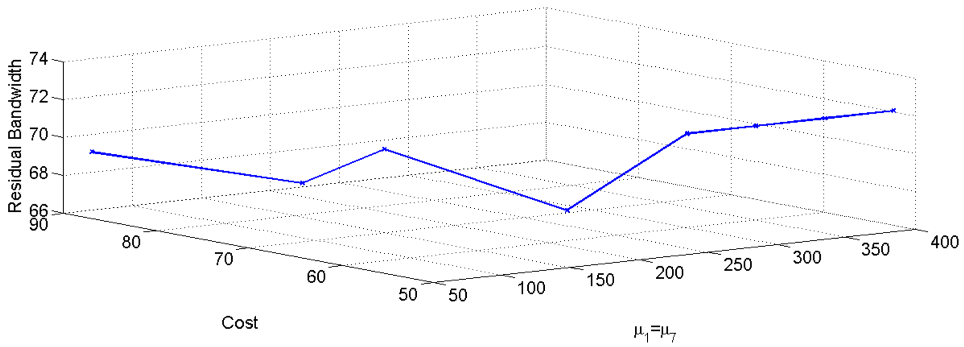


Fig. 4.14 Effect of varying HNN coefficients (μ_1=μ_7) on cost and residual bandwidth

These results suggest that the values of *(μ_1=μ_7)* between 250 to 400 are providing lower cost and higher residual bandwidth. To decide further, the μ_1 is kept fixed as 300 and the effect of varying μ_7 between 0 and 400 are summarized in Table 4.5 and Fig. 4.15. The μ_7 is varied with a step size of 50.

Table 4.5 Cost and residual bandwidths for different values of μ_7 and fixed μ_1 = 300

Value of μ_7	Cost	Residual Bandwidth
0	47.5883	69.5339
50	47.5883	69.5339
100	47.5883	69.5339
150	47.5883	69.5339
200	52.4003	72.0405
250	52.4003	72.0405
300	52.4003	72.0405
350	52.4003	72.0405
400	52.4003	72.0405

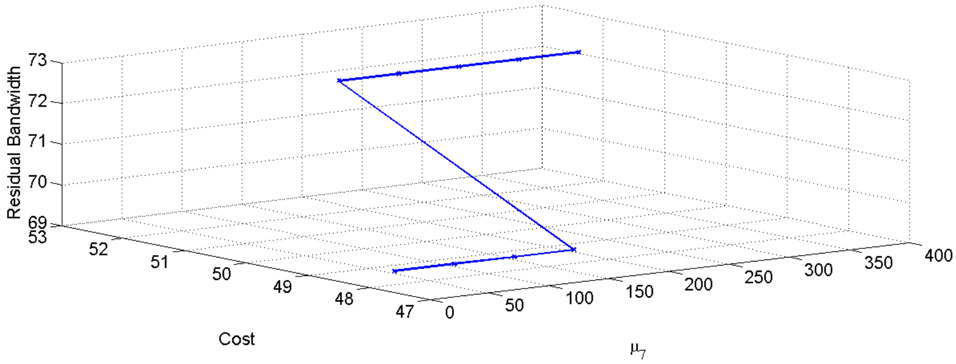


Fig. 4.15 Effect of varying μ_7 on cost and residual bandwidth

This investigation suggests that for lower values of μ_7, the optimization attempts to minimize the cost and for higher values of μ_7, the residual bandwidth is maximized. There is a range of μ_7 (200-400) for which the identical results as obtained in Table 4.4 for similar range of coefficients are obtained. To deliberate further, the μ_7 is now kept at 300 and the effect of varying μ_1 between 0 and 400 are summarized in Table 4.6 and Fig. 4.16.

Table 4.6 Cost and residual bandwidths for different valued of μ_1 and fixed μ_7=300

Value of μ_1	Cost	Residual Bandwidth
0	85.154	69.9348
50	91.7717	69.8034
100	70.1899	70.4003
150	86.2642	73.2368
200	52.4003	72.0405
250	52.4003	72.0405
300	52.4003	72.0405
350	52.4003	72.0405
400	52.4003	72.0405

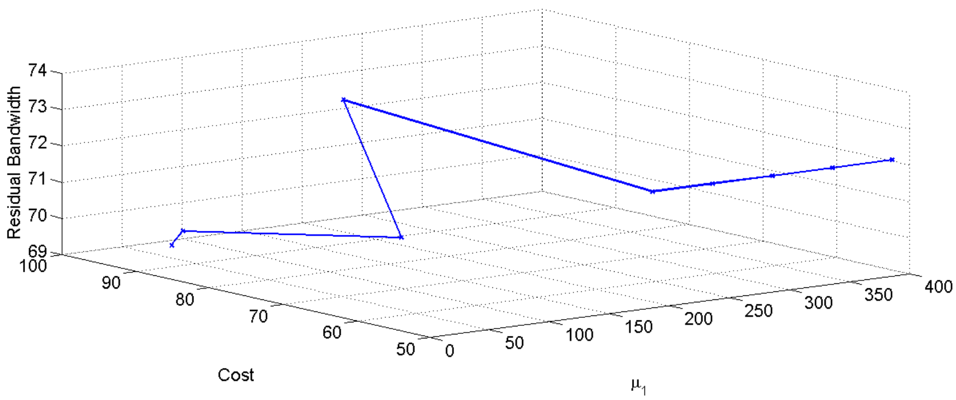
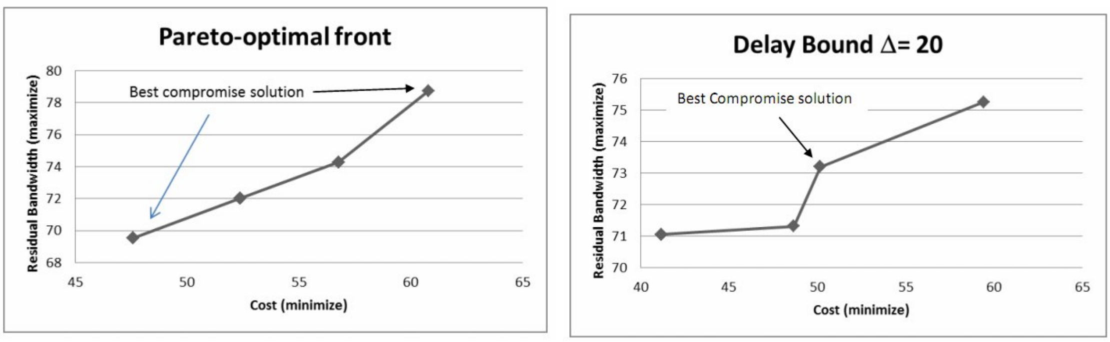


Fig. 4.16 Effect of varying μ_1 on cost and residual bandwidth

This investigation suggests that for lower values of μ_1, the optimization attempts to maximize the cost and for higher values of μ_1, cost minimization is resulted. There is a range of μ_1 (200-400) for which identical results as obtained with equal values of coefficients are obtained. This analysis suggests that the values of μ_1 and μ_7 be kept between 200 to 400. This will provide sufficient weights for both the cost minimization and residual bandwidth maximization.

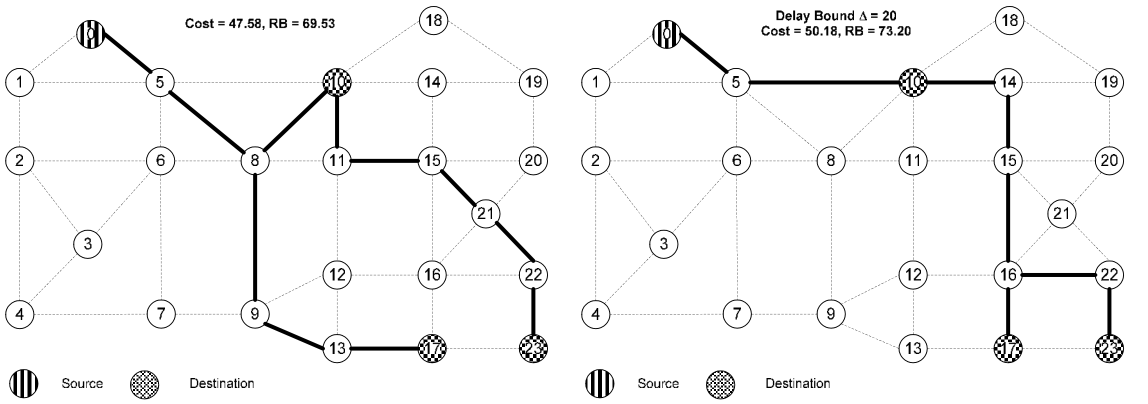
With this investigation, the values of μ_1 and μ_7 are taken as 300 and the simulation is run for different values of weights for cost-residual bandwidth optimization and delay-bound cost-residual bandwidth optimization with delay bound Δ as 20. Being the tree-structured nature of the problem, different optimal solutions are not obtained for each run or combinations of weights W_1 and W_2. The results obtained for different weights are presented in objective space. These solutions are the Pareto-optimal solutions and are nondominating to each other. This implies that the improvement in one objective is possible only at the weakening of other. The Pareto-optimal front for optimum cost-residual bandwidth and delay-bound optimum cost-residual bandwidth are shown in Fig. 4.17(a) and Fig. 4.17(b) respectively. The best compromised solutions in the objective space are indicated on the respective front.



(a) cost-residual bandwidth (b) delay bound cost residual bandwidth

Fig. 4.17 Pareto-optimal fronts for multiobjective cost-residual bandwidth optimization

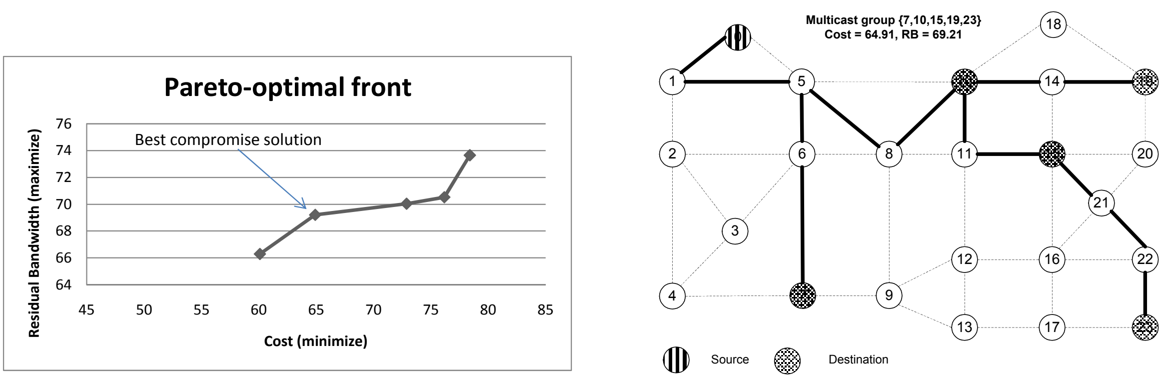
In Fig. 4.17(a), two compromise solutions are resulting due to the Pareto-front taking the shape of typical straight line nature. For both these solutions, normalized membership function β is assuming equal value. Either of the two extreme solutions can be taken as best compromised solution. The best compromised multicast trees for respective optimization are shown in Fig. 4.18(a) and Fig. 4.18(b)..



(a) cost-residual bandwidth (b) delay bound cost residual bandwidth

Fig. 4.18 Best compromised multicast trees for optimum cost-residual bandwidth

The Pareto-optimal front for cost-residual bandwidth optimization for source as node ‘0’ and multicast group as {7,10,15,19,23} is shown in Fig. 4.19(a). Corresponding to the best compromised solution indicated in Fig. 4.19(a), the best compromised tree is shown in Fig. 4.19(b). For the best compromised tree, the values of cost and residual bandwidth are resulted as 64.91 and 69.21%.



(a) Pareto-optimal front (b) best compromised tree

Fig. 4.19 Pareto-optimal front and best compromised tree for cost-residual bandwidth optimization for multicast group {7,10,15,19,23}

4.5 CONCLUDING REMARKS

The Hopfield neural network based approaches are used to obtain multicast tree for minimum cost, optimal residual bandwidth and optimum cost-residual bandwidth optimization. The respective optimization has been investigated without and with delay bound. The cost and residual bandwidth optimization has been carried out using shortest path tree formulation and multicast tree formulation. The multiobjective cost-residual bandwidth optimization has been attempted using multicast tree formulation. For the multiobjective optimization, weighted sum approach is used to form the energy function. The study has been carried out on 24-node USIP Backbone network for different multicast groups. The following conclusions are drawn from the investigations-

- The choice of Hopfield neural network parameters which are the coefficients participating in energy function, is very decisive and the optimal solution depends on these parameters. The coefficients are selected such that they impose penalty to non-existing link being included, to ensure the construction of a continuous path, to ensure the construction of virtual link from destination to source and the constraint satisfaction.

- With the selected HNN parameters, the energy and the neural network dynamics is continuously improved towards the convergence with the progress of iterations.

- Both shortest path tree and multicast tree approaches are forming the multicast tree and able to handle delay constraint. However, the multicast tree approach is yielding slightly better results due to the consideration of tree metric such as $C(T(s,M))$ instead of path metric such as $C(P(s,d_m))$.

- For increasing size of multicast groups, the tree cost increases and the residual-bandwidth value decreases slightly.

- The inclusion of delay bound is forcing the algorithm to take paths satisfying the delay bound constraint first and then optimizing the objective function term. This is resulting into higher cost and reduced bandwidth.

- For multiobjective cost-residual bandwidth optimization, each combination of weights is not always resulting into different solutions. This is due to the tree-structured nature of the problem. The tree-structured routing problem is of discrete optimization nature and thereby resulting into limited Pareto-optimal solutions.

- The algorithms are typically taking the 700-5000 iterations for the studied networks. The developed formulations are showing tendency of being trapped in local minima for large network of 100-node.

MULTICAST ROUTING USING POPULATION BASED OPTIMIZATION METHODS

5.1 GENERAL

The population based stochastic search and optimization algorithms namely genetic algorithm (GA) and particle swarm optimization (PSO) that are employed for solving QoS multicast routing problems, obtain the optimal solution by improving the initialized random feasible solutions with the progress of iterations. Many individuals of the respective algorithms search for multiple good solutions in parallel and therefore are suited for NP-complete problems such as QoS routing problems under investigation. These algorithms represent an intelligent exploitation of an adaptive search within a defined search space to solve the problem.

The GA mimics the evolutionary process of natural selection, variation, and genetics for tuning of parameters and to constitute the search. A population of solutions evolves from one generation to the next through operators namely selection, crossover and mutation to generate new solutions. This evolutional process is repeated for number of generations, and the best solution found is returned as the optimal solution. The PSO algorithm mimics the swarm intelligence and maintains a population of particles. The particles start at random locations and search for the optimal location by moving in the search space. The behaviour of the particles is influenced by their own success and from the success of peers. There is no notion of offspring generation in PSO, however, it can be considered an evolutionary algorithm (Eberhart and Shi, 1998). Due to information sharing mechanism, the PSO can be trapped into local optimum and therefore the mutation is used. (Coello *et al*., 2004).

The GA and PSO algorithms fundamentally solve the unconstrained optimization problem, whereas most of the problems are constrained optimization type. These algorithms work satisfactorily when constraints are added as penalty in the objective function and the constrained problem is solved as unconstrained optimization (Venkatraman and Yen, 2005; Mallipeddi and Suganthan, 2010).

There is some literature attributed to the use of GA and PSO for multicast routing. Ravikumar and Bajpai (1998) used GA to obtain delay bound least cost multicast tree. The GA is used to form bandwidth-delay constrained least-cost multicast tree (Zhengying *et al.,* 2001; Haghighat *et al.,* 2004). The method proposed by Randaccio and Atzori (2007) employs GA based solution to the group multicast problem by generating a set of possible trees for each session in isolation. Yen *et al.* (2008) proposed a method for forming the route, where the nodes were selected with the minimum energy consumption. The QoS constrained least cost multicast tree is obtained using PSO (Wang *et al.,* 2010). The algorithm uses tree re-shaping by tree-merging and circle elimination for improving the performance. The QoS multicast routing is converted into an integer programming problem and the problem is solved using quantum behaved PSO (Sun *et al.,* 2011).

This chapter presents tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) for QoS constrained multicast routing. A novel ordered *M*-arrays representation scheme has been proposed for constructing the tree. Each array represents a topologically connected path from destination to source. Various operations in TSGA and TSPSO are performed while preserving the tree-structured representation of ordered *M*-arrays. The constraints are added as penalty in the objective/fitness functions. The optimal multicast trees are obtained for delay and delay-jitter bound optimum cost and optimum bandwidth on various random networks. The performance of TSGA and TSPSO is compared on the basis of convergence time for varying sizes of multicast group and varying sizes of networks.

5.2 TREE STRUCTURED ENCODING SCHEME

Obtaining the multicast tree is a NP-complete problem. In the population based optimization methods, the solutions representing multicast trees are initialized. After having improvements with the progress of iterations, the optimal multicast tree is resulted at the convergence. The performance measure expressed in terms of objective or evaluation function depends on the tree and thus there is need to have an encoding or representation scheme of the solution multicast tree.

An individual or a solution is encoded using ordered *M*-arrays, where *M* represents the number of destinations in multicast group. The each array contains the node numbers forming a random loop-free route from a destination node in multicast group to source node. The order of the arrays remains unchanged or remains fixed during the iterative process. For the multicast group $\{d_1, d_2, \ldots d_m, \ldots, d_M\}$, the first array represents the path $P(d_1,s)$; second array represents the path $P(d_2,s)$ and so on. In this way *m*th array represents the path $P(d_m,s)$ and *M*th array represents the path $P(d_M,s)$. The size of an array varies depending on the number of nodes in each path and it may contain the maximum *V* nodes. Two consecutive nodes represent the existence of a link. Therefore, each array has a sequence of nodes from a destination node to the source node that are topologically connected. In this way a solution is encoded or represented as multi-array arrangement. The loop free multicast tree is obtained by testing the connectivity of each node in following array with that of preceding arrays. Therefore, the multicast tree is generated through the following two steps -

- Form solution as M-array structure.
- Form multicast tree

5.2.1 Forming solution as M-array structure

The solution *'S'* is represented by M-array, where each array represents the random path from a destination node to source node via topologically connected nodes. The random path between destination and source is generated through the following steps. Correspondingly, the flowchart is shown as Fig. 5.1.

1. Set array index $i = 1$.
2. Generation of path $P(d_i,s)$ that starts from destination node. Add the destination node 'd_i' in *i*th array and make destination node as the current node 'c_1'.
3. Generate a node 'c_2' randomly from V.
4. If node 'c_2' is already in *i*th array, go to step 3.
5. If the link exists between current node 'c_1' and generated node 'c_2', add node 'c_2' in the *i*th array and make 'c_2' as the current node 'c_1'. Otherwise go to step 3.
6. If current node 'c_1' *is* the source node '*s*', go to step 7. Otherwise go to step 3.
7. If array index $i = M$, stop. Otherwise array index $i = i+1$ and go to step 2.

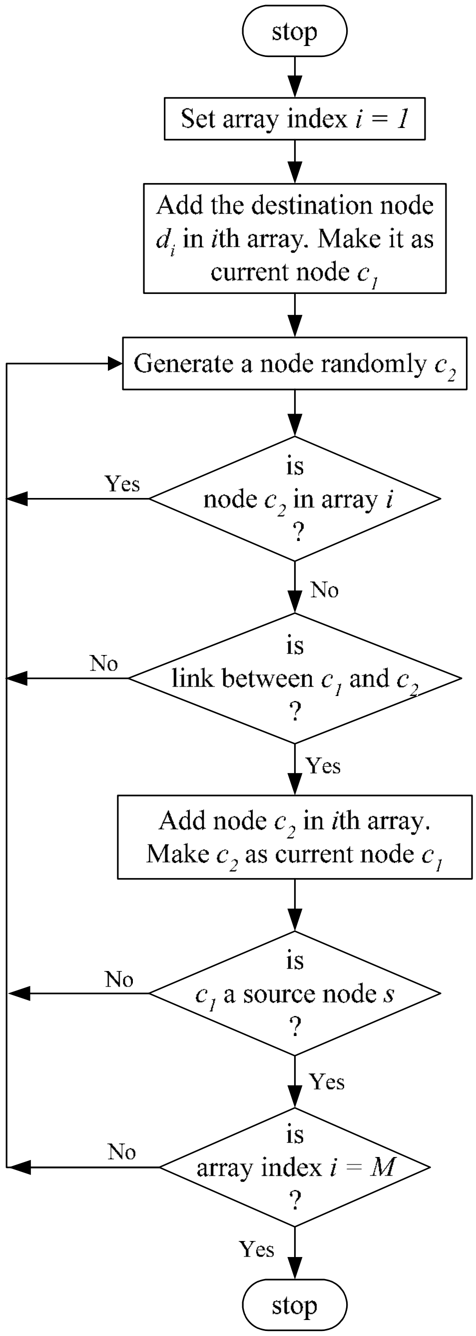


Fig. 5.1 Flowchart to generate M-array structure of solution

5.2.2 Forming multicast tree

The individual solution 'S' has been represented as ordered *M*-array structure. In this representation, each array represents a topologically connected loop free path. This structure is used to create a topologically connected tree 'T' where each node is visited only once. The tree is constructed by visiting a node on path $P(d_j,s)$ in one by one manner. The flowchart is shown in Fig. 5.2.

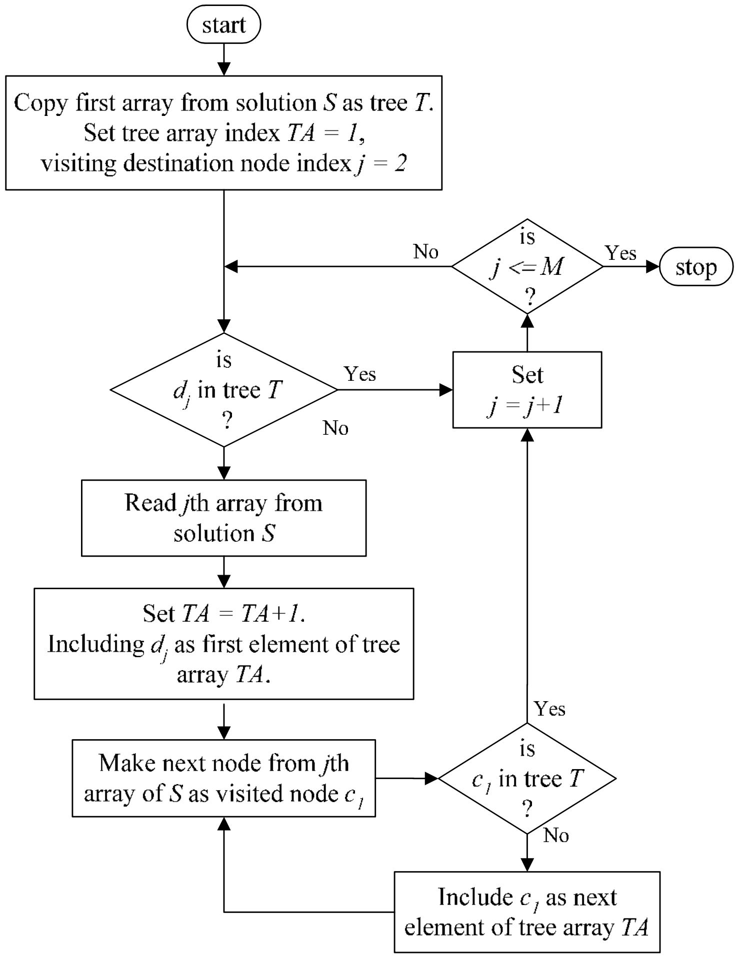


Fig. 5.2 Flowchart representing the steps to obtain multicast tree

The steps to form multicast tree from ordered *M*-array structure are summarized as:

1. Take the path from first destination node to source $P(d_1,s)$ as multicast tree *T* i.e. copy first array from '*S*' as first array of '*T*'. Set tree array index $TA = 1$, visiting destination node index $j = 2$.

2. If d_j is in multicast tree *T*, Set $j = j+1$. Otherwise go to step 5.

3. If $j \leq M$, go to step 2, else stop.

4. Read path $P(d_j,s)$ i.e. read *j*th *array* of the solution *S* from *M*-array representation.

5. Set $TA = TA+1$. Augment the multicast tree *T* by including d_j as first element in tree array *TA*.

6. Visit next node on a path $P(d_j,s)$ in one by one manner. Let the visited node is 'c_1'.

7. If node 'c_1' is not in tree *T*, include it as next element of array *TA* and go to step 6. Otherwise, set visiting array index $j = j+1$ and go to step 4.

The ordered M-array structure and formation of tree through it is explained with the help of an example network shown in Fig. 5.3. The cost is also indicated on links. The source node is numbered as ‘0’ whereas multicast group *M* members are nodes 2, 4, 6. As explained, the individual solution will have 3-array structure for three destination nodes in the example network. Only the link cost is indicated. The order of arrays remains unchanged and therefore the arrays are representing the random loop free paths P(2,0), P(4,0) and P(6,0).

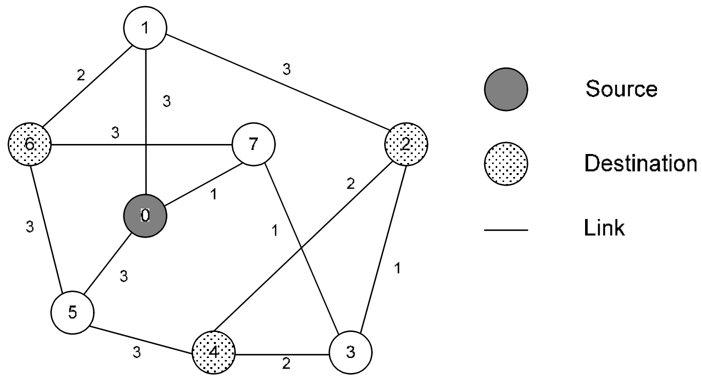


Fig. 5.3 An eight node example network topology

With the above procedure random solutions are generated. The structures of two individual solutions S_1 and S_2 and corresponding loop free trees T_1 and T_2, as obtained by the above procedure, are presented in Fig. 5.4. The first array of the multicast tree is same as that of the respective first array of the solution. The exact structure of multicast tree will depend on analyzing the path from other destination nodes to source one by one. When a node in the path from other destinations to source is found already in tree, like node ‘3’ in path P(4,0) of solution S_1, it is joined with tree instead of further traverse to source. Therefore, the elements in the subsequent tree array may be less. The arrays in the tree will not be M when a destination node is already visited. As shown, the tree T_2 is represented by only two arrays because destination node ‘6’ is already included in the generated path P(4,0) of the solution S_2.

5.3 TREE-STRUCTURED GENETIC ALGORITHM

The genetic algorithm is a population-based evolutionary algorithm. The evolutionary process uses Darwinian principle of natural selection and genetics to constitute the search. A population of solutions evolves from one generation to the next through operators namely selection, crossover and mutation. The decision variables are encoded and the fitness function based on the objective is defined to evaluate the solutions. The selection stochastically chooses good solutions with the idea to prefer better solutions to worse ones from the population as the parents. Normally, tournament selection is preferred. The crossover operator combines the two selected parents and generates offspring solutions to inherit some desirable features from parents to offspring. The mutation operator applies random perturbations on offspring to introduce some new traits that are not present in the present population. When this evolutional process is repeated for number of generations, the optimal solution is obtained.

Solution S_1

2	3	7	0		
4	3	7	0		
6	5	4	3	7	0

Solution S_2

2	1	0		
4	5	6	1	0
6	1	0		

Tree Matrix T_1

2	3	7	0
4	3		
6	5	4	

Tree Matrix T_2

2	1	0	
4	5	6	1

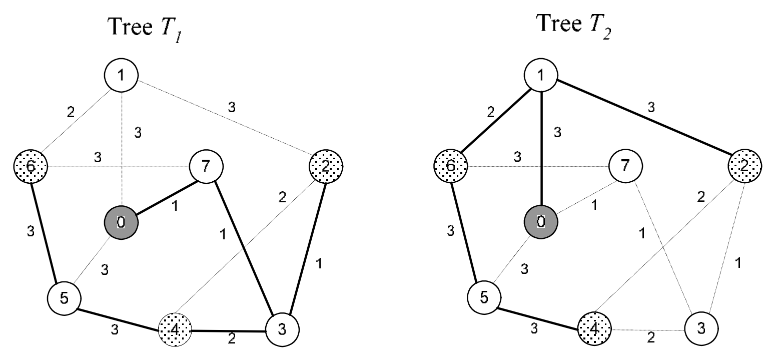


Fig. 5.4 An illustration of solutions and tree matrix and multicast tree

The developed TSGA is a genetic algorithm with the tree-structured representation scheme suited for multicast routing. As basic GA, it has an encoding tree representation scheme, initialization of population, fitness evaluation, selecting the parents in mating pool, use of tree-crossover operator to generate new multicast trees, use of tree-mutation operator suggesting random perturbation on the multicast tree and the termination condition. With reference to TSGA, these are briefly discussed herewith.

Encoding Scheme and Initialization

The encoding scheme has been described in section 5.2. A solution is encoded or represented as ordered *M*-array arrangement. Each array has a sequence of nodes from a destination node to the source node that are topologically connected. The loop free multicast tree is obtained by testing the connectivity of each node in the following array with that of the preceding arrays. With the above procedure the population of *N* individuals is created.

Fitness Evaluation

The evaluation function is directly associated with the mathematical formulation of the problem under investigation. It depends on the objective and QoS constraints. As different QoS multicast routing problems have been investigated, the fitness evaluation functions of the respective problems are formulated in separate section 5.4.

Selection of parents for forming new generation

The *selection* is intended to improve the average fitness of the population in mating pool i.e. the population participating in the next generation. The principle of "survival of fittest" suggests that the solution with higher fitness is having better chance to get copied in mating pool. This has been implemented using Tournament selection, which inherently preserve elitism. The pool size is taken equal to the size of the initial population N.

Tree-crossover to generate new multicast trees

The crossover is an important random operator and its function is to generate two offspring or 'child' solutions from two randomly selected 'parent' solutions by combining the information extracted from the parents. To perform tree-crossover operation, two individuals are selected randomly from parent population. To emulate single point crossover, a destination node is selected randomly and the paths from this randomly selected destination node to the source node between two parent solutions are swapped. The realization of tree-crossover operation for generating offspring is shown in Fig. 5.5 where the array at position 2 in parent solutions S_1 and S_2 are swapped to obtain two offspring solutions O_1 and O_2 and offspring tree. Similarly, two point crossover can be emulated by swapping two arrays from two parent population.

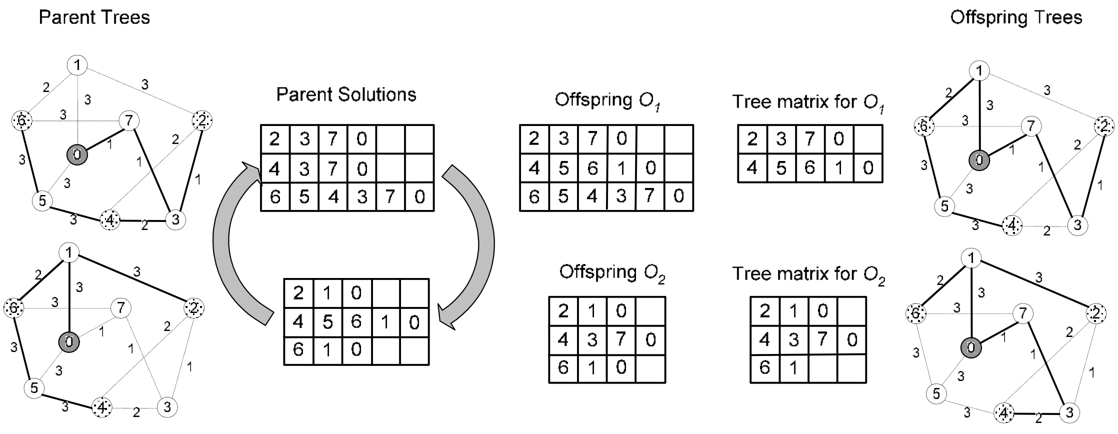


Fig. 5.5 Realizing tree-crossover operation in the proposed scheme

Tree-mutation to introduce random traits in multicast trees

Mutation is another important operator in evolutionary algorithm and alters an offspring. The mutation operation avoids the search turning into a primitive random search. In the proposed algorithm, the mutation is applied on the offspring resulted after crossover. In the M-array representation of an individual, an array representing a path from a destination node to source is selected randomly. In this array, a node is selected randomly and the sub-path from this selected not to the source is replaced by newly generated random sub path. In this way, the offspring get altered to yield altered multicast tree. The realization of mutation operation in the proposed scheme is shown in Fig. 5.6. The node '3' in 2nd array i.e. P(4,0) is selected randomly from the offspring O_2. The altered offspring O_2 is resulted by replacing the path from node '3' to source '0' in the 2nd array by new random path.

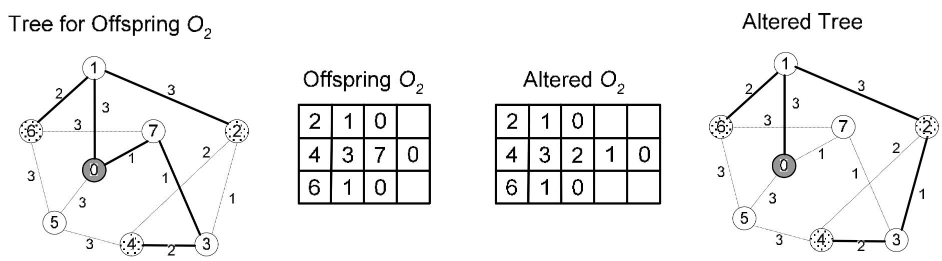


Fig. 5.6 Realizing tree-mutation operation in the proposed scheme

Termination condition

With the *selection*, the number of solutions in the mating pool having higher fitness increases with the progress of iterations. When the 80% of the solutions in mating pool attains the maximum fitness value or the maximum iterations are expanded, the search is terminated and the solution with maximum fitness is regarded as optimum multicast tree.

Having discussed various operations associated with TSGA, the flow-diagram is shown as Fig. 5.7 and the algorithm steps are summarized as follows:

1. Generate the initial random population of ordered M-arrays as per section 5.2.1.
2. Form multicast trees as per section 5.2.2 and evaluate them.
3. Form mating pool by Tournament selection.

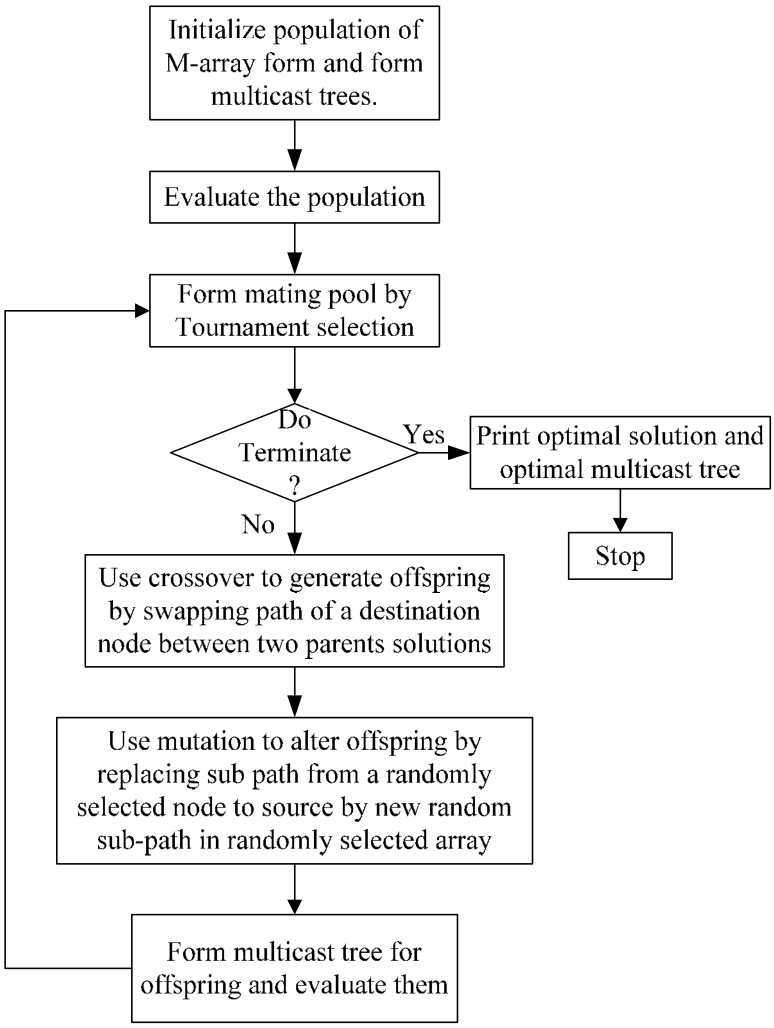


Fig. 5.7 The illustration of TSGA algorithm for multicast routing

4. If termination condition is satisfied, print the optimal solution and optimal tree and stop.
5. Apply *tree-crossover* to generate new offspring by swapping the paths from a random destination node to source between two parent solutions.
6. Apply *tree-mutation* to alter offspring by replacing the sub-path from a randomly selected node to source in a randomly selected array by new random sub-path.
7. Form multicast trees for the offspring, evaluate them and go to step 3.

5.4 TREE-STRUCTURED PARTICLE SWARM OPTIMIZATION

The Particle Swarm Optimization (PSO) is a population based stochastic search and optimization technique that mimics the swarm intelligence. It maintains a population of particles (swarm) where each particle represents a position (solution). The particles are initialized random positions (solutions) which search for the optimal location (solution) by updating their positions with the progress of iterations moving in the search space. The particles positions are evaluated on the basis of objective function. The changes in the particles velocity and positions are influenced by their own success and from the success of peers. In this way, particles interact by sharing information about their positions. The PSO updates the particle (solution) $\boldsymbol{x}_i$ at the generation t as –

$$x_i(t) = x_i(t-1) + v_i(t) \tag{5.1}$$

Where,

$$v_i(t) = w \times v_i(t-1) + C_1 \times r_1 \times (x_i^{pbest} - x_i) + C_2 \times r_2 \times (x^{gbest} - x_i) \tag{5.2}$$

In eq. (5.1), the x_i^{pbest} is the historical best position of particle x_i, the x^{gbest} is the historical global best solution viewed by entire swarm, C_1 and C_2 are the study factors that control the effect of personal and global best particles, r_1 and r_2 are two mutually independent uniformly distributed random numbers in range [0, 1] and w is the inertia weight that controls the trade-off between local and global experience.

The optimization using PSO algorithm, as explained above earlier, is realized by initializing the particles randomly, identifying their individual best positions and global best position. The new positions of the particle are obtained using their current positions, their own best positions and the global best position until the convergence is achieved. In the presented tree-structured PSO (TSPSO) algorithm, each particle is representing M arrays indicating the random paths from various destinations in the multicast group to source. With these arrays, the multicast tree is formed. At the convergence of TSPSO, the optimal multicast tree is obtained. The M-array representation of solution, construction of multicast tree and evaluation of the solution using objective function is same as explained in TSGA.

The eq. (5.1) and eq. (5.2), which are used for updating the positions of the particles for continuous variables, can not be used in the current in TSPSO form because of the tree-structured nature of the problem. However, the new positions are obtained using current position x_i, personal best position x_i^{pbest} and the global best position x^{gbest} through *tree-merge* operation. With the tree-merge operation, ordered M-arrays are randomly picked from x_i x_i^{pbest} and x^{gbest}. With *tree-merge* operation, the particle position (solution) gets updated. The tree-merge operation is illustrated in Fig. 5.8.

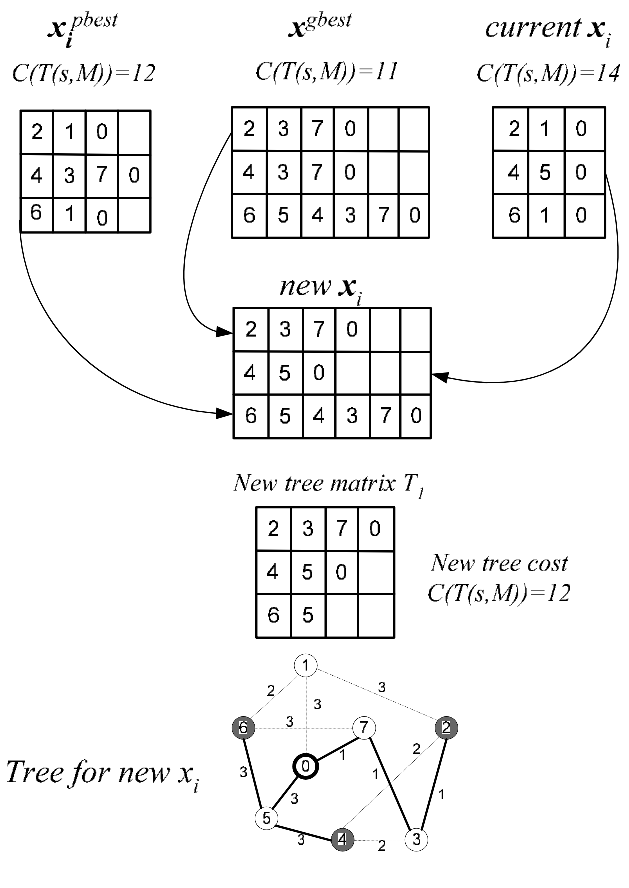


Fig. 5.8 Tree-merge operation to obtain new position of the particle

The tree-mutation as discussed in section 5.3 is applied stochastically to avoid the algorithm being trapped into local minima. The flowchart for TSPSO is shown in Fig. 5.9 and the steps are described as-

1. Specify swarm size, maximum generations t_{max} and set generation index $t = 1$.
2. Initialize the positions of the particles (solutions) randomly. As explained in *tree structured encoding* in section 5.2, each particle x_i is represented by M-arrays. Where, each array indicates the random path from a destination node to source node.

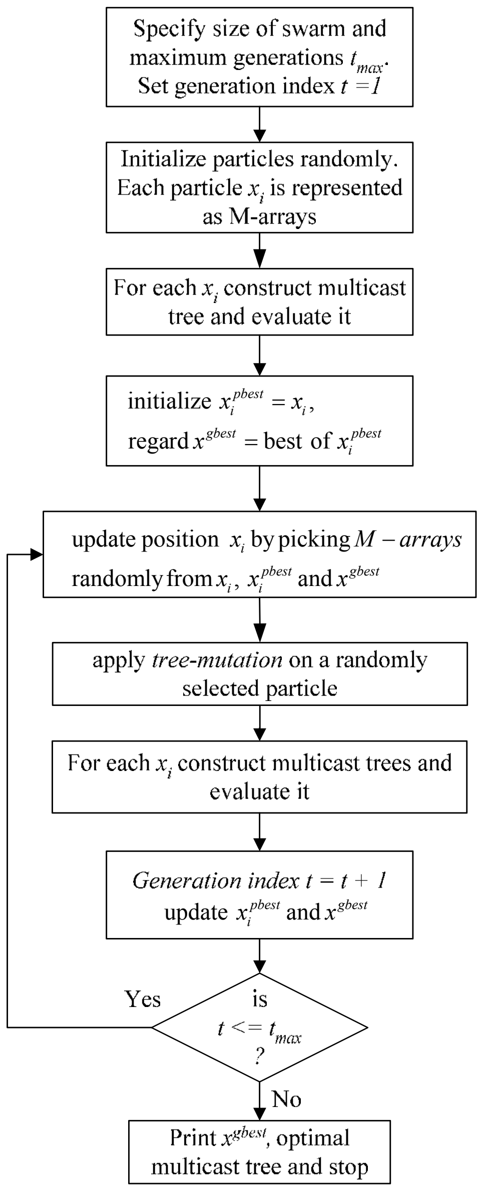


Fig. 5.9 The illustration of TSPSO algorithm for multicast routing

3. Construct the multicast trees for each x_i and evaluate the particles using objective function.
4. Initialize the local best solution x_i^{pbest} as x_i. Regard the best local best solution as the current global best solution x^{gbest}.
5. Update the position of each particle x_i by using x_i , x_i^{pbest} and x^{gbest} each of which is having M-arrays. The updated x_i is obtained by randomly picking the M-arrays from x_i , x_i^{pbest} and x^{gbest}.
6. Apply *tree-mutation* stochastically on randomly selected particle.
7. Construct the multicast trees for each x_i and evaluate the particles using objective function.
8. Update x_i^{pbest} and x^{gbest}. Set generation index $t = t+1$.
9. If ($t < t_{max}$) go to step 5, else return the optimal multicast tree corresponding to x^{gbest} and stop.

5.5 EVALUATION FUNCTIONS FOR ROUTING PROBLEMS

The formulations of QoS multicast routing problems under investigations are discussed in Chapter 3. These problems have been investigated to find the optimal tree. The investigations are carried out in this chapter to obtain QoS constrained minimum cost multicast tree and QoS constrained optimum residual bandwidth multicast tree. These trees are obtained after evaluating each solution with the consideration of objective and constraints. The evaluation function is derived by representing the constrained optimization problem as unconstrained form. This is explained as follows :

5.5.1 Function for QoS constrained minimum cost tree

The problem to find a tree rooted at the source s and spanning to all the members of M by minimizing the total cost of multicast tree and satisfying the end-to-end delay and delay jitter constraints has been defined in section 3.3. The formulation is summarized as –

$$\text{Minimize} \qquad C(T(s,G)) = \sum_{e \in T(s,G)} C(e)$$

Subjected to

$$\max_{d_i \in M}(D(P_T(s,d_i)) \le \Delta \tag{5.3}$$

$$\max_{d_i \in M}(D(P_T(s,d_i)) - \min_{d_j \in M}(D(P_T(s,d_j)) \le \delta$$

As the objective is the minimization of cost, the evaluating function EF_{cost} has been represented as sum of cost and penalty for the violation of constraints as –

$$EF_{cost} = f_{ct} + A\, f_{delay} + B\, f_{jitter} \tag{5.4}$$

Where

$$f_{ct} = C(T(s,G)) = \sum_{e \in T(s,G)} C(e) \tag{5.4a}$$

$$f_{delay} = \max\{delay_{\max} - \Delta, 0\} \tag{5.4b}$$

$$f_{jitter} = \max\{(delay_{\max} - delay_{\min}) - \delta, 0\} \tag{5.4c}$$

$$delay_{\max} = \max_{d_i \in M}(D(P_T(s,d_i)) \tag{5.4d}$$

$$delay_{\min} = \min_{d_i \in M}(D(P_T(s,d_i)) \tag{5.4e}$$

The A and B are the penalty factors for the violation of delay and delay jitter bounds. These factors are specified as 100 during simulation.

5.5.2 Function for QoS constrained optimum residual bandwidth tree

The problem to find a tree rooted at the source s and spanning to all the members of M by maximizing the residual bandwidth of the multicast tree and satisfying the end-to-end delay and delay jitter constraints has been defined in section 3.4. The formulation is summarized as –

Maximize $RB(T(s,M)) = \dfrac{\sum_{e \in T(s,M)} (\phi(e) - \lambda(e))}{\sum_{e \in T(s,M)} \phi(e)}$

Subjected to

$$\max_{d_i \in M}(D(P_T(s,d_i)) \le \Delta \tag{5.5}$$

$$\max_{d_i \in M}(D(P_T(s,d_i)) - \min_{d_j \in M}(D(P_T(s,d_j)) \le \delta$$

As the objective is to maximize the residual bandwidth, the evaluating function EF_{RB} has been derived by applying penalty for the constrained violation. For constrained violation EF_{RB} value will be reduced considerably and therefore, the feasible tree while maximizing the residual bandwidth will be resulted -

$$EF_{RB} = \frac{f_{RB}}{1 + Af_{delay} + Bf_{jitter}} \tag{5.6}$$

Where

$$f_{RB} = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} (\phi(e) - \lambda(e))}{\sum_{e \in T(s,M)} \phi(e)} \tag{5.6a}$$

$$f_{delay} = \max\{delay_{\max} - \Delta, 0\}$$

$$f_{jitter} = \max\{(delay_{\max} - delay_{\min}) - \delta, 0\}$$

$$delay_{\max} = \max_{d_i \in M}(D(P_T(s,d_i))$$

$$delay_{\min} = \min_{d_i \in M}(D(P_T(s,d_i))$$

The A and B are the penalty factors for the violation of delay and delay jitter bounds.

5.6 RESULTS AND DISCUSSION

The developed tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) are used to investigate multicast routing for minimum cost and optimum residual bandwidth. The delay and delay jitter bounds are included as constraints for these two optimizations. The performance has been investigated on 24-node USIP Backbone network shown as Fig. A1 and various other random networks. The network data are presented in Appendix A. The random networks are generated through BRITE network topology generator using Waxman model described in Appendix B.

The TSGA is based on genetic algorithm and the choice of crossover and mutation probabilities or rates is always critical for genetic based optimization algorithms. The nine combinations of crossover probabilities as 0.9, 0.7, 0.5 and mutation probabilities as 0.025, 0.05 and 0.1 are tried on 24-node USIP Backbone network. Depending on the minimum values of cost and the iterations for different run, the crossover and mutation probabilities are selected as 0.9 and 0.05 respectively. The mutation probability in TSPSO is also taken as 0.05. The initial population size in the TSGA and the initial swarm size in TSPSO are considered sixty for analyzing 24-node network. This choice of initial solutions is based on an experiment summarized in Table 5.1, where the simulation is run for different sizes of population. The TSGA involves crossover and selection whereas TSPSO updates the solution values (positions) only. When the 70% of the population or swarm attain same solution value or the optimum solution has not changed for 25 iterations, it is assumed that further improvement in the solution is not possible. This condition concurrently with the maximum number of generations is taken as termination condition for both TSGA and TSPSO.

Table 5.1 Effect of the population size on the delay and delay-jitter bound optimum cost and convergence time for 24-node USIP Backbone network

(Source s={8}, Multicast group={1,14,19,23}, delay bound Δ=15, delay jitter bound δ=5)

Population size	Cost	Maximum delay	Maximum jitter	Iterations	Convergence Time
20	74.20	12.23	4.18	30	0.05
40	63.50	12.23	2.76	58	0.14
60	**58.27**	**12.23**	**3.33**	**82**	**0.26**
80	58.27	12.23	3.33	115	0.46
100	59.05	12.23	4.18	41	0.26

As observed, the population size of sixty is resulting into minimum value of cost and also having fast convergence, the population size is taken sixty for this 24-node network.

5.6.1 Cost minimization

For the 24-node USIP Backbone network, the results obtained from the TSGA and TSPSO for cost minimization are summarized in Table 5.2 for three cases namely minimum cost, delay-bound minimum cost, delay and delay jitter bound minimum cost. Correspondingly, the optimal M-array structures and optimal multicast trees are shown in Fig. 5.10(a), Fig. 5.10(b) and Fig. 5.10(c) respectively.

Table 5.2 : Summary of minimum cost multicast tree formulation using TSGA and TSPSO (case I : minimum cost, case-II : delay bound minimum cost, case III : delay and delay-jitter bound minimum cost)

Source node s = {8}, Multicast group M = {1, 14, 19, 23}, delay bound Δ = 15, delay jitter bound δ = 5.							
	Optimization using TSGA			Optimization using TSPSO			
Case	cost	Iterations	Time (sec)	Cost	Iterations	Time	Better solution
I	42.47	79	0.45	42.47	73	**0.38**	TSPSO
II	51.96	92	0.47	51.96	79	**0.39**	TSPSO
III	58.27	41	**0.26**	58.27	55	0.30	TSGA

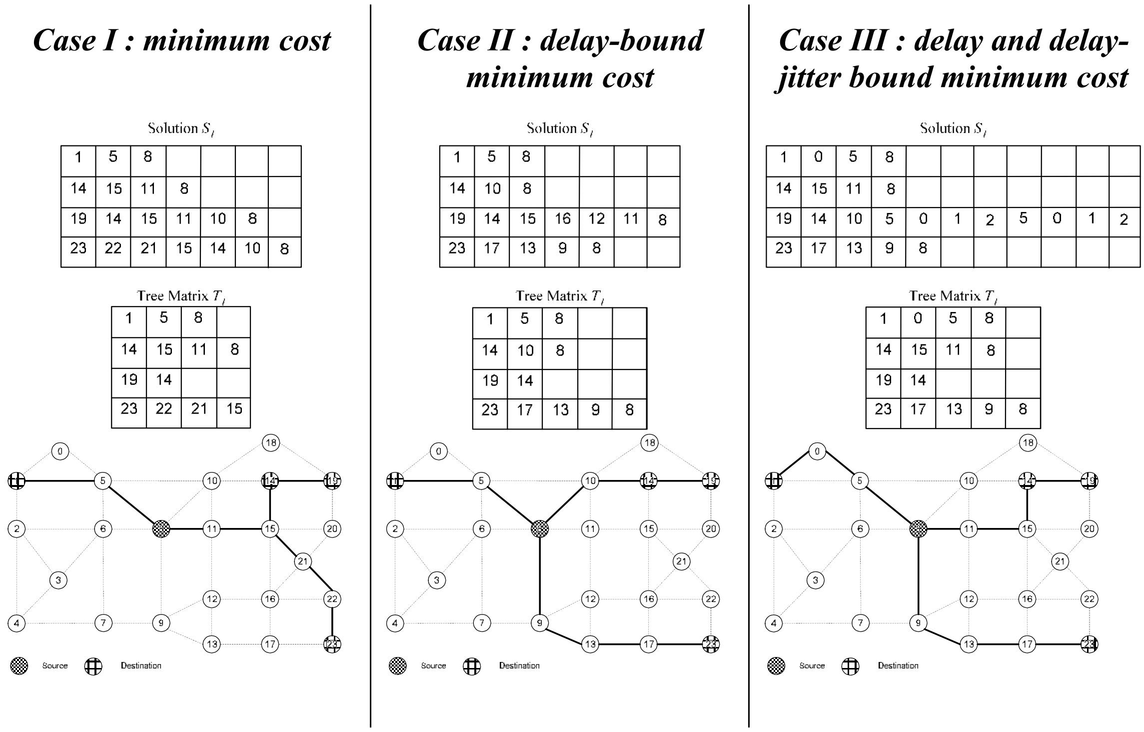


Fig. 5.10 Representation of optimal solution and multicast Tree for minimum cost routing

The optimization using TSGA and TSPSO for three cases namely case-I, case-II and case-III is resulting into same optimal costs as 42.47, 51.96 and 58.27 respectively. However, the time expended with TSPSO is less for Case-I and Case-II. For the Case-III, the TSGA is showing faster convergence.

The results obtained from TSGA and TSPSO for delay bound minimum cost multicast routing for varying delay bound Δ are shown in Table 5.3. The study has been carried out on the same source node *s* as 8 and same multicast group as {1,4,19,23}. From the study it is observed that not many different solutions are resulted due to the change in the delay bound Δ. Both TSGA and TSPSO are resulting into identical costs, however, the convergence time is less in TSPSO. With very large value of delay bound Δ, the formulation approaches towards unconstraint minimum cost optimization.

The results obtained from TSGA and TSPSO for delay and delay-jitter bound minimum cost multicast routing for varying delay-jitter bound δ are shown in Table 5.4. The source node *s* and multicast group are kept unchanged. The delay bound Δ is kept fixed as 25. The TSPSO is providing better solution (as far δ =9 and δ =12) or convergence is fast (as far δ =15).

Table 5.3 : Effect of delay bound Δ on delay-bound minimum cost routing (source s = {8}, Multicast group = {1, 4, 19, 23})

	TSGA		TSPSO	
Delay	cost	time	Cost	time
13	51.96	0.38	51.96	0.39
15	51.96	0.47	51.96	0.39
19	51.96	0.62	51.96	0.38
22	42.47	0.58	42.47	0.28
25	42.47	0.44	42.47	0.33
28	42.47	0.31	42.47	0.33

Table 5.4 : Effect of delay jitter bound δ on delay and delay-jitter bound minimum cost routing (source s = {8}, Multicast group = {1, 4, 19, 23}, delay bound Δ=25)

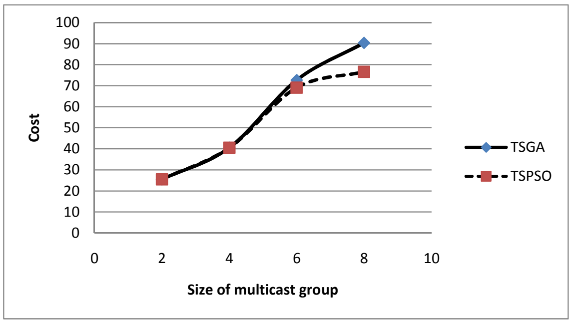
Jitter	Performance using TSGA		Performance using TSPSO	
	Cost	Time	Cost	Time
5	58.27	**0.26**	58.27	0.30
9	51.96	0.40	**48.35**	0.34
12	51.96	0.35	**45.10**	0.38
15	42.47	0.45	42.47	**0.38**

The effectiveness of TSGA and TSPSO in providing optimal solution for different multicast groups has been investigated and the results are summarized in Table 5.5. These results pertain to uniform delay bound of Δ as 20 and delay jitter bound δ as 10 and the source node *s* as 8. Correspondingly the variations in cost, convergence time, max delay and max delay jitter are shown in Fig. 5.11(a), 5.11(b) and 5.11(c), Fig. 5.11(d) respectively.

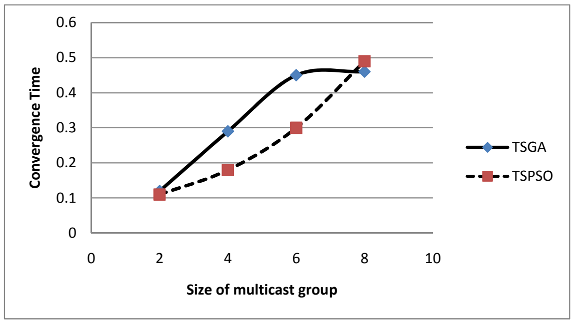
Table 5.5 : Summary of delay and delay-jitter bound minimum cost for different multicast groups (delay bound Δ = 20, delay jitter bound δ =10)

Multicast group		Performance using TSGA				Performance using TSPSO			
size	members	Cost	Max delay	Max jitter	time	Cost	Max delay	Max jitter	time
2	{0,21}	25.54	9.02	2.24	0.12	25.54	9.02	2.24	**0.11**
4	{0,7,12,22}	40.56	9.39	5.37	0.29	40.56	9.39	5.37	**0.18**
6	{0,4,7,15,19,23}	72.66	12.23	8.21	0.45	**69.10**	15.52	9.84	0.30
8	{0,3,6,10,13,16,20,23}	90.37	13.73	7.94	0.46	**76.63**	14.90	9.09	0.49

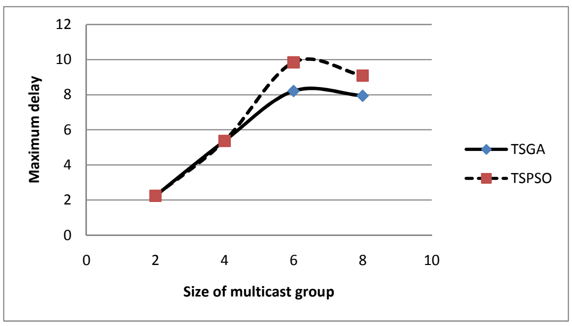
As observed from the Table 5.5, the convergence time increases with the size of the multicast group. The resulted maximum delay and maximum delay jitter also increase with the size of the multicast group because of the increase in the size of multicast tree. However, the exact value depends on the link parameters and the relative position of multicast group and source. The summarized results suggests that TSPSO gives better results as obtained for multicast group of size 6 or 8 or resulting into lower convergence time as obtained for multicast group of size 2 and 4 compared to TSGA. As shown in Fig. 5.11, the TSPSO converges quickly for small multicast group, however, convergence time for both TSGA and TSPSO is nearly same for large multicast group. As observed from Fig. 5.11(c) and Fig. 5.11(d), TSPSO method results into higher values of maximum delay and delay jitter. This suggests that TSPSO provides lower cost solution by searching paths that are of low cost but resulting delay and delay jitters close to their respective limiting values. The optimal M-array solutions for these multicast groups are shown in Table 5.6. With the help of these M-array solutions, the optimal trees are obtained.



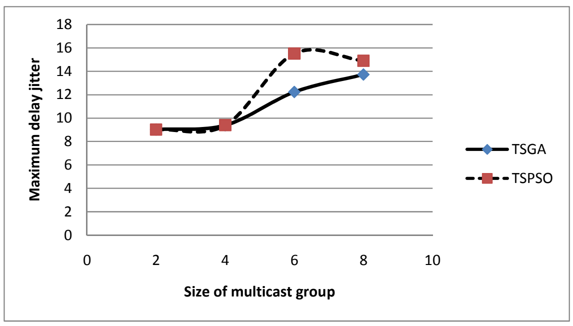
(a) cost



(b) convergence time



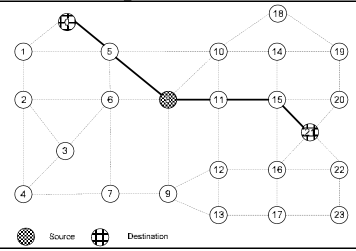
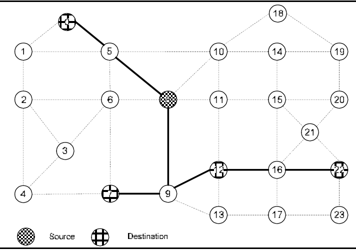
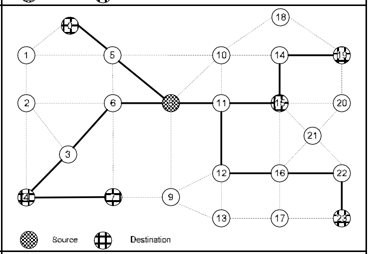
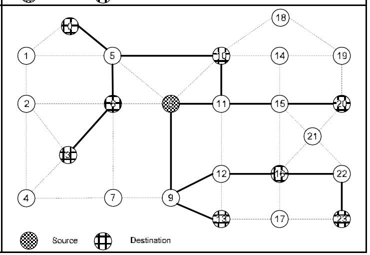
(c) Maximum delay



(d) Maximum delay jitter

Fig. 5.11 Comparative performance of TSGA and TSPSO for different sizes of multicast groups on 24-node network

Table 5.6 The structure of optimal solution and tree for different multicast groups

<table>
<tr><th>Multicast group</th><th>Optimal Solution</th><th>Optimal tree</th></tr>
<tr><td>{0,21}</td><td>0 5 8
21 15 11 8</td><td>
</td></tr>
<tr><td>{0,7,12,22}</td><td>0 5 8
7 9 8
12 9 13 17 23 22 21 20 15 11 8
22 16 12 11 8</td><td>
</td></tr>
<tr><td>{0,4,7,15,19,23}</td><td>0 5 8
4 3 6 8
7 4 2 6 8
15 11 8
19 14 15 11 8
23 22 16 12 11 8</td><td>
</td></tr>
<tr><td>{0,3,6,10,13,16,20,23}</td><td>0 5 10 11 8
3 6 5 10 8
6 5 10 8
10 11 12 16 17 13 9 7 4 3 6 5 8
13 9 8
16 12 9 8
20 15 11 8
23 22 16 15 14 10 5 0 1 2 4 7 9 8</td><td>
</td></tr>
</table>

The performance is also investigated on 100 node random network generated using BRITE topology based on Waxman model. The network data is given in Appendix-A. The simulation is carried out with the initial population or swarm size of 200. The performance is studied for various sizes of the multicast groups. For the given size of multicast group, the multicast group is selected randomly. However, the results from TSGA and TSPSO are obtained for same multicast groups. The effect of varying delay-bound Δ for delay-bound minimum cost multicast routing is summarized in Table 5.7. The analysis is carried out where the delay bound Δ is taken as (15+size of multicast group). The need of varying Δ is experienced because of the large network and the random choice of multicast group. The cost and convergence time are shown in Fig. 5.12(a) and Fig. 5.12(b) respectively. The TSPSO is resulting into lower cost and lower convergence time for delay-constrained minimum cost routing. Similarly, the effects of varying delay bound Δ and delay jitter bound δ are summarized in Table 5.8. The delay jitter bound δ is taken as half of the delay bound Δ for different combinations of multicast groups. The delay bound is taken as (20+size of multicast group). The resulting cost and convergence time are shown in Fig. 5.13 (a) and 5.13 (b) respectively. The TSPSO is resulting into lower cost and slightly fast convergence compared to respective values for TSGA.

Table 5.7 Effect of delay bound for varying size of multicast group on 100 node random network

Size of multicast group	Delay bound Δ	Performance using TSGA		Performance using TSPSO	
		Cost	Time	Cost	time
5	20	83.42	8.02	69.55	3.22
10	25	134.37	13.85	137.42	6.87
15	30	156.89	19.63	132.62	8.23
20	35	202.08	40.46	163.49	14.53
25	40	240.57	35.03	192.49	20.27
30	45	270.39	37.61	246.4	21.33
35	50	314.66	59.35	266.08	42.06
40	55	342.31	74.99	299.56	41.67

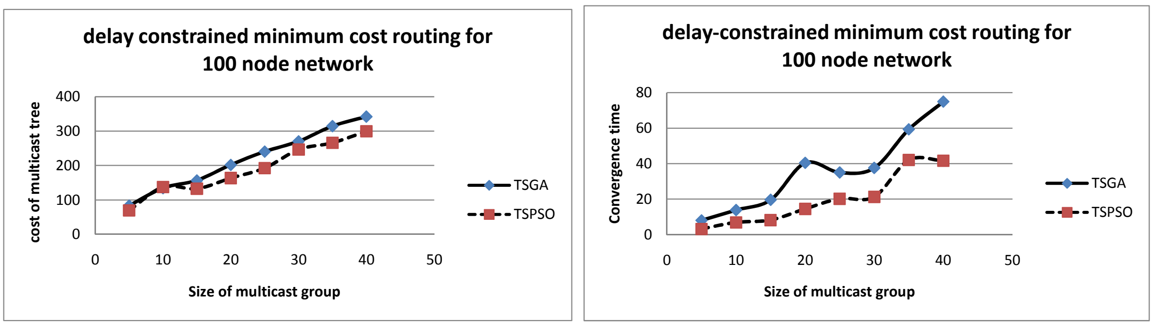


Fig. 5.12 Effect of varying multicast group on 100-node random networks for delay-constrained minimum cost for varying delay bound Δ

The Table 5.7 and 5.8 summarizes the results for delay bound minimum cost and delay and delay jitter bound minimum cost routing. The performance of TSGA and TSPSO for these constraint optimizations are shown in Fig. 5.14(a) and Fig. 5.14(b) respectively. These results confirm that cost increases when both the constraints are applied simultaneously. Both TSGA and TSPSO are yielding higher costs for delay and delay jitter bound minimum cost optimization in comparison to the respective delay bound minimum cost optimization.

Table 5.8 Effect of delay and delay jitter bounds for varying size of multicast group on 100 node random network

Size of multicast group	Delay bound Δ	Delay jitter bound δ	Performance using TSGA		Performance using TSPSO	
			Cost	Time	Cost	time
5	25.0	12.5	88.74	4.02	99.07	4.18
10	30.0	15.0	146.56	15.03	142.59	13.83
15	35.0	17.5	156.89	19.63	151.57	11.74
20	40.0	20.0	219.03	24.58	192.6	19.70
25	45.0	22.5	243.47	47.48	240.01	29.80
30	50.0	25.0	273.01	63.11	253.35	49.90
35	55.0	27.5	333.87	77.39	284.44	70.01
40	60.0	30.0	370.82	51.68	320.7	52.53

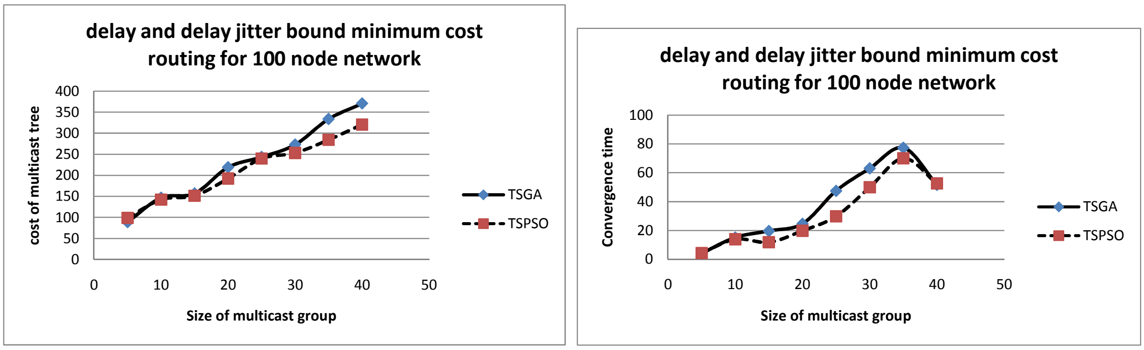


Fig. 5.13 Effect of varying multicast group size on 100-node random networks for delay and delay jitter bound minimum cost routing for varying Δ and δ

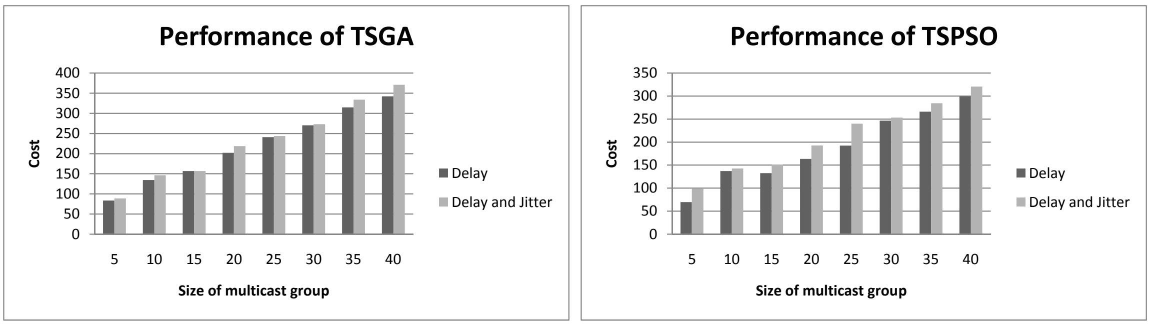
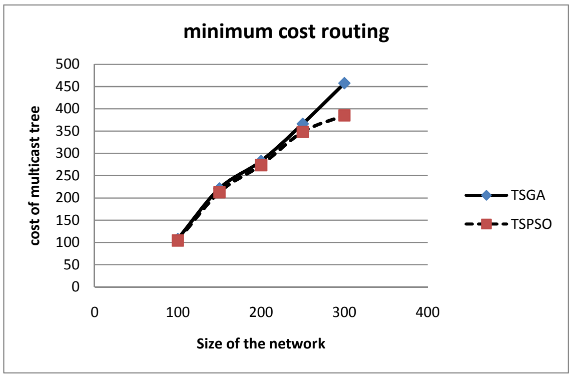


Fig. 5.14 The performance of TSGA and TSPSO for handling constrained minimum cost routing

The performance for different size random networks is summarized in Table 5.9 for minimum cost optimization. These random networks are generated using BRITE network topology generator. The multicast group size is taken as 10% of the network size. The performance is also shown in Fig. 5.15. With both TSGA and TSPSO, convergence time increases considerably with the increase in the network size. The rate of increase of convergence time is higher in TSPSO. However, it is resulting into slightly reduced cost.

Table 5.9 TSGA and TSPSO performance for minimum cost routing for different networks (multicast group size in 10% network size)

Network size (N)	Size of Multicast group	Performance using TSGA		Performance using TSPSO	
		Cost	Time	Cost	Time
100	10	107.65	9.22	104.59	7.30
150	15	221.23	30.26	212.92	22.85
200	20	282.82	62.38	273.76	58.42
250	25	366.37	179.73	348.32	175.24
300	30	457.74	292.18	385.37	339.11



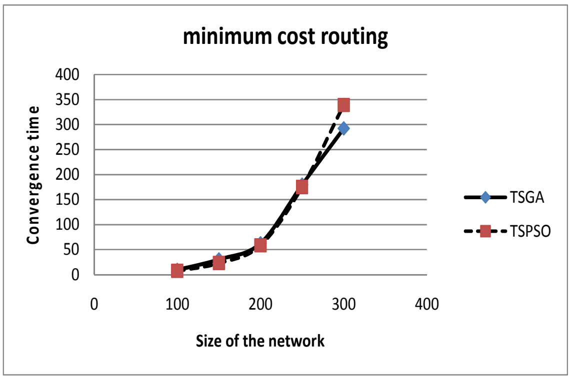


Fig. 5.15 Minimum cost routing for varying network size

5.6.2 Residual Bandwidth Optimization

For the 24-node USIP Backbone network, the analysis is carried out using the same parameters such as crossover and mutation probability, maximum numbers of iterations, termination criterion etc. The results obtained from the TSGA and TSPSO for residual bandwidth optimization are summarized in Table 5.10 for three cases namely optimum residual bandwidth, delay-bound optimum residual bandwidth, delay and delay jitter bound optimum residual bandwidth. Correspondingly, the optimal multicast trees are shown in Fig. 5.16(a), Fig. 5.16(b) and Fig. 5.16(c) respectively.

The optimization using TSGA and TSPSO is resulting into same value of residual bandwidth as 69.74 for delay bound optimum residual bandwidth. The resulted optimum residual bandwidths for case-I and case-III are slightly higher through TSPSO. For all these cases, the time expended with TSPSO is significantly less compared to TSGA.

Table 5.10 : Summary of optimum residual bandwidth multicast tree formulation using TSGA and TSPSO (case I : optimum residual bandwidth, case-II : delay bound optimum residual bandwidth, case III : delay and delay-jitter bound optimum residual bandwidth)

Source node s = {8}, Multicast group M = {1, 14, 19, 23}, delay bound Δ = 15, delay jitter bound δ = 5.							
	Optimization using TSGA			Optimization using TSPSO			
Case	Residual Bandwidth	Iterations	Time (sec)	Residual Bandwidth	Iterations	Time	Better solution
I	72.69	90	0.77	**74.89**	38	0.08	TSPSO
II	69.74	54	0.50	69.74	40	**0.08**	TSPSO
III	61.90	40	0.46	**65.96**	35	0.08	TSPSO

Case I : optimum RB | ***Case II : delay-bound optimum RB*** | ***Case III : delay and delay-jitter bound optimum RB***

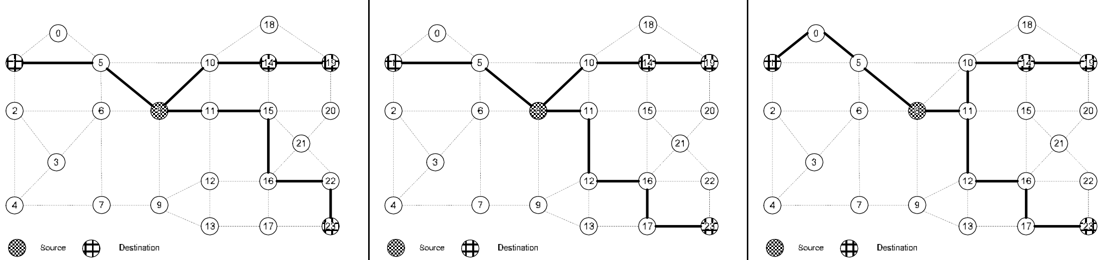


Fig. 5.16 Representation of multicast Tree for residual bandwidth optimization

The results obtained from TSGA and TSPSO for delay bound residual bandwidth optimization for varying delay bound Δ are shown in Table 5.11. The study has been carried out on the same source node *s* as 8 and same multicast group as {1,4,19,23}. The delay bound is varied from 13 to 28 in the step of 3, however, not many solutions are resulted. The simulation through TSPSO is resulting into slightly higher residual bandwidth and simulation converges very fast compared to TSGA. With large delay bound Δ, the formulation approaches towards unconstraint residual bandwidth optimization.

The results obtained from TSGA and TSPSO for delay and delay-jitter bound optimum residual bandwidth routing for varying delay-jitter bound δ are shown in Table 5.12. The delay bound Δ is kept fixed as 25 and the delay-jitter bound δ is varied from 3 to

21 with step size of 3. The TSGA is resulting in better residual bandwidth for jitter as 6 and 12. However, for other cases TSPSO is providing better solution or fast convergence.

Table 5.11 : Effect of delay bound Δ on delay-bound optimum residual bandwidth routing (source s = {8}, Multicast group = {1, 4, 19, 23})

	TSGA		TSPSO	
Delay	Residual Bandwidth	time	Residual Bandwidth	time
13	64.92	0.53	64.92	**0.08**
16	69.74	0.48	**72.55**	0.09
19	69.74	0.47	**74.89**	0.09
22	71.99	0.45	**74.89**	0.08
25	71.99	0.48	**74.89**	0.08
28	71.99	0.67	**74.89**	0.08

Table 5.12 : Effect of delay jitter bound δ on delay and delay-jitter bound optimum residual bandwidth routing (source {8}, Multicast group={1,4,19,23}, delay bound Δ=25)

Jitter	Performance using TSGA		Performance using PSPSO	
	Residual Bandwidth	Time	Residual Bandwidth	Time
3	66.78	0.74	**68.24**	0.09
6	**63.17**	0.22	61.90	0.08
9	62.29	0.39	**66.91**	0.08
12	**72.55**	0.47	68.35	0.07
15	66.91	0.40	**74.89**	0.09
18	74.89	0.44	74.89	**0.08**
21	71.99	0.44	**74.89**	0.08

For the 100 node random network, the effect of varying delay-bound Δ for delay-bound residual bandwidth optimization is summarized in Table 5.13. The analysis is carried out for various multicast groups selected randomly. The delay bound Δ is varied as (15+size of multicast group). The resulted residual bandwidths and convergence time are shown in Fig. 5.17(a) and Fig. 5.17(b) respectively.

The Fig. 5.17 suggests that TSPSO is resulting into higher residual bandwidth for various multicast groups compared to the respective values obtained using TSGA. The convergence time increases with the increase in size of multicast group. For small size of multicast group, both TSGA and TSPSO are having nearly same convergence time. However, the TSGA converges slowly for large multicast group. The variation in the convergence time for TSPSO is smooth and showing direct relation with the size of multicast group.

Table 5.13 Effect of delay bound on delay bound optimum residual bandwidth routing for varying size of multicast group on 100 node random network

Size of multicast group	Delay bound Δ	Performance using **TSGA**		Performance using **TSPSO**	
		Residual Bandwidth	Time	Residual Bandwidth	time
5	20	63.25	3.35	65.63	2.65
10	25	62.52	4.68	65.31	3.74
15	30	61.36	6.43	65.14	5.82
20	35	62.58	13.42	64.93	7.98
25	40	64.62	42.54	65.64	10.37
30	45	62.49	20.2	65.83	14.18
35	50	65.79	43.79	67.74	18.41
40	55	62.64	50.81	64.66	23.79

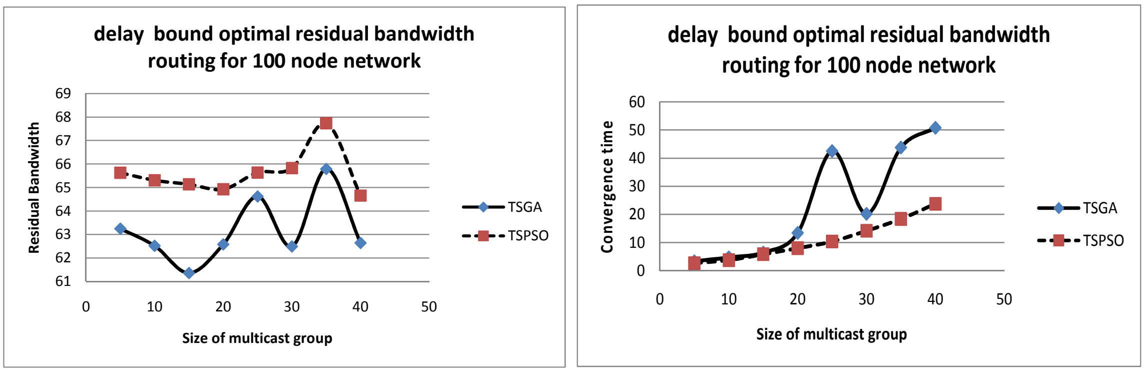
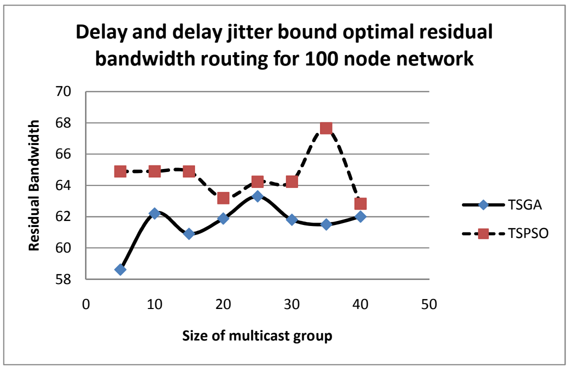


Fig. 5.17 Effect of varying multicast group on 100-node random networks for delay-constrained optimum bandwidth for varying delay bound Δ

For the 100-node random network, the effects of varying delay bound Δ and delay jitter bound δ are summarized in Table 5.14. The delay jitter bound δ is taken as half of the delay bound Δ for different combinations of multicast groups. The delay bound is taken as (25+size of multicast group). The resulting residual bandwidth and convergence time are shown in Fig. 5.18 (a) and 5.18 (b) respectively.

Table 5.14 Effect of delay and delay jitter bounds for varying size of multicast group on 100 node random network

Size of multicast group	Delay bound Δ	Delay jitter bound δ	Performance using **TSGA**		Performance using **TSPSO**	
			Residual Bandwidth	Time	Residual Bandwidth	time
5	30	15	58.61	2.7	64.89	1.84
10	35	17.5	62.2	8.25	64.9	4
15	40	20	60.9	14.4	64.89	8.4
20	45	22.5	61.88	12.99	63.19	8.22
25	50	25	63.3	30.42	64.23	13.78
30	55	27.5	61.8	25.58	64.24	13.2
35	60	30	61.5	35.05	67.65	16.97
40	65	32.5	62	54.45	62.83	18.29



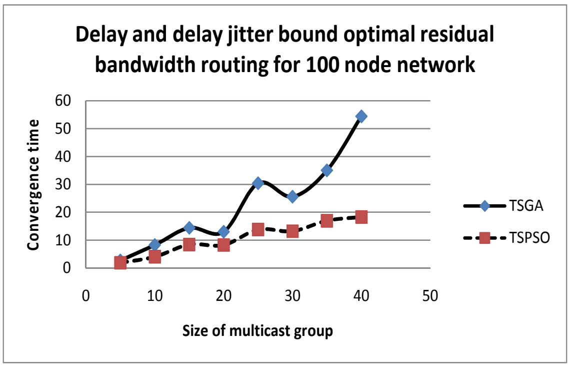
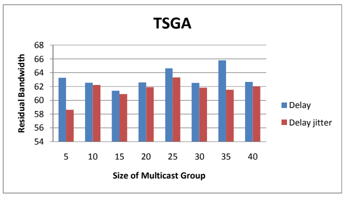


Fig. 5.18 Effect of varying multicast group size on 100-node random networks for delay and delay jitter bound optimal residual bandwidth routing for varying Δ and δ

As observed from the Fig. 5.18, the higher residual bandwidth is resulted through TSPSO. The convergence time increases smoothly with the increase in the size of multicast group. For TSGA, the convergence time increases exponentially for large multicast group.

The comparative performance of TSGA and TSPSO for QoS constrained optimum residual bandwidth routing are shown in Fig. 5.19. With both the algorithms, the obtained residual bandwidth for different multicast groups decreases with the inclusion of delay-jitter constraints.



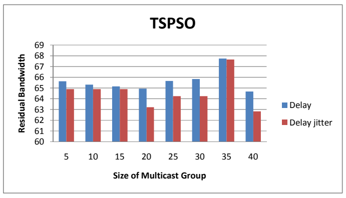


Fig. 5.19 The performance of TSGA and TSPSO for handling constrained optimum residual bandwidth routing

5.7 CONCLUDING REMARKS

A tree-structured encoding scheme is presented and two algorithms namely tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) are proposed to obtain the optimal multicast tree for QoS constrained minimum cost and QoS constrained optimal residual bandwidth optimization. The end-to-end delay and delay jitter are considered as the QoS constraints. The study has been carried out on 24-node USIP Backbone network and various random networks obtained through BRITE network generator. The following conclusions are drawn from the investigations-

- The proposed encoding scheme representing the solution as ordered M-array structure is effective in providing loop free tree. The effectiveness of this encoding has been tested through TSGA and TSPSO.

- Both the developed algorithms TSGA and TSPSO are effective in handling both delay bound and delay jitter bound for minimum cost and optimum residual bandwidth routing.

- With the inclusion of constraints, the cost of resulting optimal tree increases. Whereas the residual bandwidth decreases with the inclusion of constraints.

- The TSPSO yields the optimal tree that is having lower cost compared to the optimal tree resulted by TSGA. This suggests that TSPSO forms the tree through low cost paths which are resulting into delay and delay jitter close to the limiting values.

- The TSPSO yields the optimal tree having slightly higher residual bandwidth.

- The convergence time for both TSGA and TSPSO increases with the increase in size of multicast group. The TSPSO converges fast compared to the convergence attained by TSGA.
- For both TSGA and TSPSO, the convergence time increases with the increase in the size of the network.

MULTIOBJECTIVE MULTICAST ROUTING USING POPULATION BASED OPTIMIZATION METHODS

6.1 GENERAL

The QoS multicast routing problem involves various QoS constraints and objectives such as cost, delay, delay jitter, residual bandwidth, packet-loss-probability etc. These objectives and constraints are in conflict like various other real world problems. The consideration of various objectives simultaneously in multicast routing will provide more realistic solution and will provide better choice to the decision maker. In contrast to the single objective optimization which is providing single solution, the solution to multiobjective optimization yields a set of non-dominated solutions called Pareto optimal set. All Pareto optimal solutions are equally good.

The multiobjective optimization has been investigated by various researchers. The classical methods can be classified as *a priori*, *post-priori* and *interacting* type (Veldhuizen and Lamont, 2000; Deb 2001). In a priori methods like weighted sum approach, the preference to an objective is first decided. Correspondingly, the weights are assigned for different objectives. The weighted sum of objectives is formulated to be optimized by single objective optimization. The post-priori methods do not require the preference of the objectives but require the knowledge of parameters for good solution. The ε-constraint method, a post-priori method, optimizes one objective at a time and constrains all others within the limit decided by decision maker. The interactive methods interact with decision maker during the process of optimization. Although, the interaction provides flexibility, the methods lose simplicity.

These classical methods although give one Pareto-solution, the typical single objective optimizer has to run many times. These methods are not effective in obtaining Pareto-optimal solutions in non-convex region. Even the uniformity in Pareto-optimal solutions can not be maintained. Further, these classical methods require higher degree of problem knowledge. The population based nature of the optimization methods like GA and PSO make them suitable for multiobjective optimization. These methods due to inherently exploring multiple solutions in parallel and their ability to be used with discrete search space can provide number of equally good solutions in single run.

One of early work to solve multiobjective optimization using evolutionary approach is attributed to Fogal *et al.* (1966). The weighted sum approach was used to handle multiple objectives using evolutionary method. However, the development of Vector Evaluated Genetic Algorithm (VEGA) by Schaffer (1985) is a pioneer work in solving the multiobjective optimization problems using evolutionary population based methods. The population is divided subpopulation groups that are equal to number of objectives being optimized. The fitness is assigned to each individual in the subpopulation. Goldberg (1989) suggested ranking of non-dominated solutions. Fonseca and Fleming (1993) presented Multiobjective GA (MOGA) where rank 1 was given to all non-dominated solutions whereas other solutions were ranked according to number of solutions that they were dominating. Srinivas and Deb (1994) presented Non-Dominated Sorting GA (NSGA). The population was sorted in order of non-dominance and subgroups or fronts were formed. All the fronts were assigned ranks in order of their non-dominance. The Strength Pareto Evolutionary Algorithm (SPEA) was proposed by Zitzler and Thiele (1999). The SPEA used elitism by using an external repository to maintain non-dominated solutions and assigned fitness to these Pareto solutions. The NSGA-II, proposed by Deb *et al.* (2002), uses elitism to NSGA. The population and offspring are combined and the parent population for next iteration is made using non-dominated crowding sort procedure.

The PSO has also been used for multiobjective optimization. Ray and Liew (2002) used Pareto ranking scheme. The non-dominated solutions are used as leaders. Less crowded leaders were selected using crowding density. The methods used external repository to store non-dominated solutions, which were used as leaders (Coello and Salazar, 2002; Coello *et al*., 2004). The Vector Evaluated PSO (VEPSO) has been proposed by Peropoulous and Vrahatis (2002). The VEPSO is along the lines of VEGA. Different swarms were used for different objectives. The information between different swarms was shared during the calculation of particle velocity and particle position. Ho *et al.* (2005) extended the original PSO and also used external archive to store Pareto solutions and assigned fitness to all Pareto solutions.

Roy and Das (2004) proposed MOGA based approach to optimize delay, bandwidth consumption and residual bandwidth utilization. Araújo and Garrozi (2010) considered three objectives for multicast routing. All objectives were combined into one function by using weighting factors depending on the importance of the objectives and used single objective optimization. Ant colony was used to simultaneously optimize cost, delay and average delay (Pinto et al., 2005). The SPEA was used for routing (Zitzler and Thiele, 1999).

This chapter presents two algorithms namely tree-structured multiobjective GA (TSMGA) and tree-structured multiobjective PSO (TSMPSO) for QoS constrained multiobjective multicast routing. These algorithms use tree-structured representation that has been discussed in Chapter-V. After obtaining the Pareto-optimal front, the best compromise solution is obtained using fuzzy cardinal priority ranking. The multicast trees are obtained for cost-residual bandwidth and cost-residual bandwidth-packet loss probability optimization under delay and delay jitter constrains. The performance of the developed algorithms TSMGA and TSMPSO is compared on the basis of convergence time for varying sizes of multicast group and varying sizes of networks.

6.2 TREE-STRUCTURED MULTIOBJECTIVE GENETIC ALGORITHM

The proposed tree-structured multiobjective genetic algorithm (TSMGA) for QoS multiobjective multicast routing is based on an elitist non-dominated sorting genetic algorithm (NSGA-II). The NSGA-II has the following features-

- It uses fast non dominated sorting techniques to provide the solutions as close as possible to the Pareto-optimal solutions.
- It uses crowding distance to provide diversity in solution.
- It uses elitism to preserve the best solutions of current population in next generation.

These are helpful for the multi-objective optimization to obtain the multiple non-dominated solutions in single run. The following terms, which are used in NSGA-II, are helpful for the better understanding of the TSMGA. The terms are –

- Constraints handling
- Non-dominance sorting and ranking of the population.

- Crowding distance computation
- Selection to support elitism

Constraints handling

The constraints are taken care of using the concept of nondomination. A solution x_i is said to constrain-dominate a solution x_j if either of the following conditions are satisfied:

- Solution x_i is feasible and solution x_j is infeasible.
- Solutions x_i and x_j are both infeasible; however, solution x_i has a smaller constraint violation than x_j.
- Solutions x_i and x_j are both feasible, and solution x_i dominates solution x_j.

Non-dominated sorting and ranking

The concept of dominance has been explained in detail in Chapter-3. A solution is said to dominate another solution, if it is no worse than other in all objectives and better than in at least one objective. A solution is said to be non-dominated if it is not dominated by any other solution. With non-dominated sorting, the population is divided in different dominance levels or fronts and is ranked. The solutions of a population P_i, which do not dominate each other but dominate all other solutions of P_i, are kept in the first front i.e. set F_1. After ignoring group of classified individuals, next dominance level is formed after checking the dominance i.e. among the solutions which are in P_i but not in $F = F_1$ i.e. the solutions ($P_i \cap F$) are sorted and the solutions which do not dominate each other but dominate all other solutions, are kept in the second front i.e. set F_2. Similarly, the solutions belonging to P_i but not belonging to $F = F_1 \cup F_2$ i.e ($P_i \cap F$) are sorted. The solutions which do not dominate each other but dominate all of the other solutions are kept in the next front i.e. set F_3. This process is repeated until all solutions in P_i are assigned one of the front. This process is shown in Fig. 6.1. Subsequently, these generated fronts are assigned ranks in order of the dominance level i.e. F_1, F_2, F_3 are assigned ranks 1, 2, 3 respectively. The lower rank number suggests higher is the fitness. Such ranking and sorting suggests that all the solutions of a front are equally good solutions. However, solutions at higher dominance level are better compared to the solutions at lower dominance level. The nature of population and corresponding sorting and ranking are explained in Fig. 6.2 for min-min type problem. The four fronts namely F_1, F_2, F_3, F_4 and F_5 have been created after non-dominance sorting.

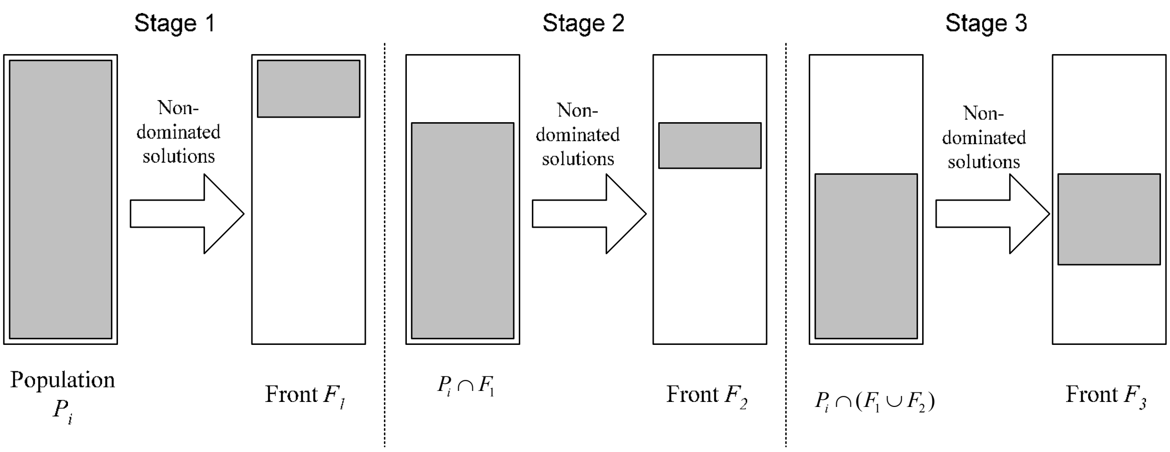


Fig. 6.1 Process to create Pareto fronts through non-dominating sorting

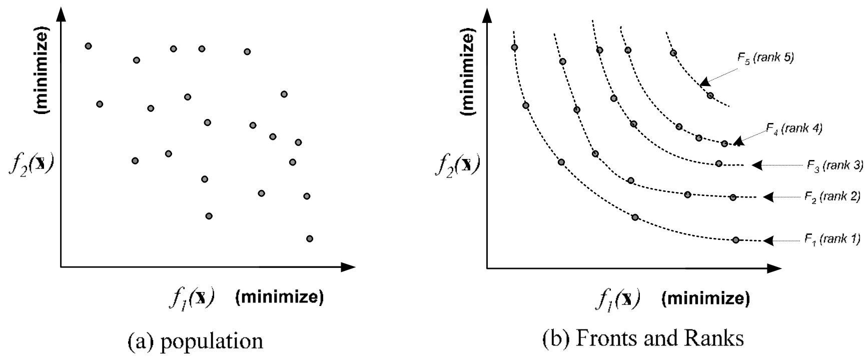


Fig. 6.2 population and fronts in objective space for min-min type problem

Crowding Distance Computation

To estimate the density of solutions surrounding a particular point in the population, the average distance of the two points on either side of this point is considered along each of the objectives. This quantity, referred as crowding distance, serves as an estimate of the size of the largest cuboid enclosing the point without including any other point in the population. It aims to obtain uniform spread of the solutions over the Pareto front and is indicative of density around a solution. The crowding-distance computation requires sorting of a given population into different non-dominating fronts and ranking.

For each front, the population is sorted according to each objective function value in ascending order of magnitude. Once this is done, the two boundary solutions with the largest and smallest objective value are assigned distance values of infinity. All other solutions lying in between these two solutions are then assigned a distance value calculated by the absolute normalized distance between each pair of adjacent solutions. The total crowding distance of a solution is the sum of the crowding distances with respect to each objective. As shown in Fig. 6.3 for bi-objective min-min problem, the crowding distance *cd(x)* for *i*th solution is the sum of crowding distances $cd_1(x)$ and $cd_2(x)$ for two objectives. In general, *cd(x)* for *i*th solution is the sum of side-lengths of the cuboid, defined by the nearest neighbours in the same front.

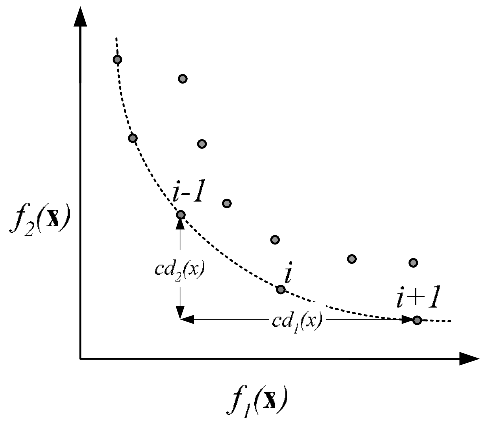


Fig. 6.3 Crowding distance for a solution

This crowded-distance is helpful in deciding the winner among the randomly selected solutions. If the solutions are in the same non- dominated front, the solution with a higher crowding distance wins. Otherwise, the solution with the lowest rank is selected. Therefore, for a given front, the solution located in a lesser-crowded region is considered better.

Preserving Elitism

The Elitism is a procedure to retain the best individuals of the current iteration (generation) for generating offspring solutions in the next iteration. As explained, the rank numbers to various non-dominating fronts are indicative of the relative superiority of the solutions. However, the solutions belonging to same front are equally good. To implement elitism, the combined population of parent and offspring solutions is created. The combined population is sorted and ranked as per the non-dominance levels and stored in different fronts. The parent population for next generation is created by taking populations from the fronts in order of their ranks until the total number of solutions in parent population is just greater than the original size of parent population. To make the size exactly equal, some of the solutions of the last included front, which are having least crowding distances, are rejected.

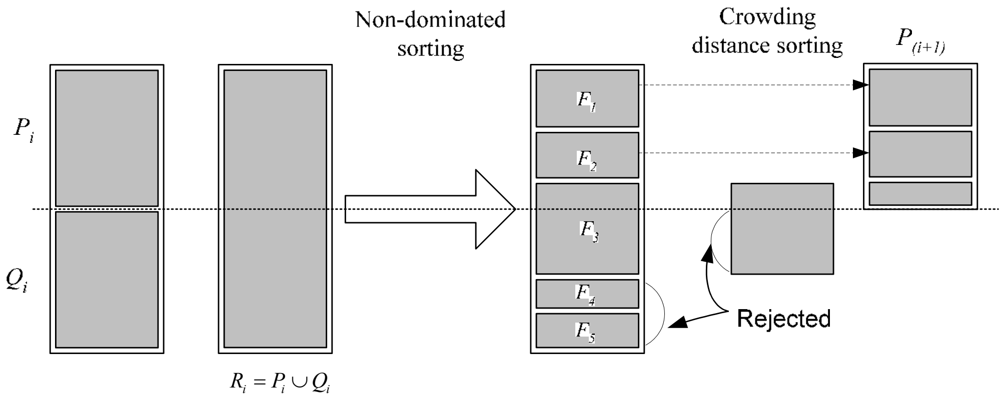


Fig. 6.4 Schematic representation to maintain elitism and population selection

This is shown in Fig. 6.4 where P_i and Q_i denote the parent and offspring population at *i*th generation. The combined population $R_i = P_i \cup Q_i$ of size $2N$ is created. The combined population R_i is sorted and ranked in different fronts. To create parent population $P_{(i+1)}$ for next generation, initially the solutions belonging to the set F_1 are considered. If size of F_1 is smaller than N, then all of the solutions in F_1 are included in $P_{(i+1)}$. The remaining solutions in $P_{(i+1)}$ are filled up from the rest of the non-dominated fronts in order of their ranks i.e. from F_2, F_3 and so on. This process continues until the condition is reached where $(F_1 \cup F_2 \cup F_3 \cup ....F_{k-1} \cup F_k) < N$ and $(F_1 \cup F_2 \cup F_3 \cup ....F_k \cup F_{k+1}) > N$ i.e no more front can be accommodated without exceeding the total number of solutions in $P_{(i+1)}$. To make the size exactly equal to N, some solutions from the last non-dominated front F_{k+1} are discarded from being included into $P_{(i+1)}$ on the basis of crowding distance. The solutions of the last non-dominated front F_{k+1} are sorted according to their crowding distances and, subsequently, the solutions having higher crowding distances are included. The inclusion of solutions with higher crowding distances provides diversity.

6.2.1 Tree-Structured Genetic Algorithm Steps

Let P_i is the parent population, Q_i is the offspring population and R_i represent the total population of the *i*th generation. F_k is the *k*th front. As discussed, the solutions in front F_1 are better than those of F_2, and so on. The convergence of TSMGA will provide Pareto-optimal solutions. The best compromise solution is obtained using fuzzy-cardinal priority ranking as discussed in section 3.2.2. The algorithm steps are shown in Fig. 6.5 and are summarized as follows:

1. Specify population size N, maximum generations i_{max} and set generation index $i = 1$.

2. Initialize the population P_i randomly. As explained in *tree structured encoding* in section 5.2, each solution is represented by M-ordered arrays. Where, each array indicates the random path from a destination node to source node.

3. Construct the multicast trees for each solution and evaluate them.

4. Use binary tournament selection and apply *tree-crossover* and *tree-mutation* operation to create offsprings Q_i of size N.

5. Construct the multicast trees for each offspring solution and evaluate them.

6. Create a combined population $R_i = P_i \cup Q_i$ of size $2N$.

7. Assign each population in R_i to the fronts $F_1, F_2, F_3, \ldots F_k \ldots F_t$ the basis of nondominance and rank them.

8. Calculate the crowding distance of solutions in each front F_k.

9. Assign half of the combined population R_i using crowded non-dominated sorting to $P_{(i+1)}$.

10. Use *tree-crossover* and *tree-mutation* to recombine the population $P_{(i+1)}$ and assign that to $Q_{(i+1)}$. Create multicast trees for $Q_{(i+1)}$ and evaluate them.

11. Increment the generation index ($i = i + 1$).

12. if ($i \leq i_{max}$) go to step 6. Otherwise, print solutions of F_1 front as Pareto-optimal solutions and optimal multicast trees.

13. Compute normalized membership function β for each Pareto-optimal solution in the front F_1. Select the solution with maximum β as the best compromise solution and stop.

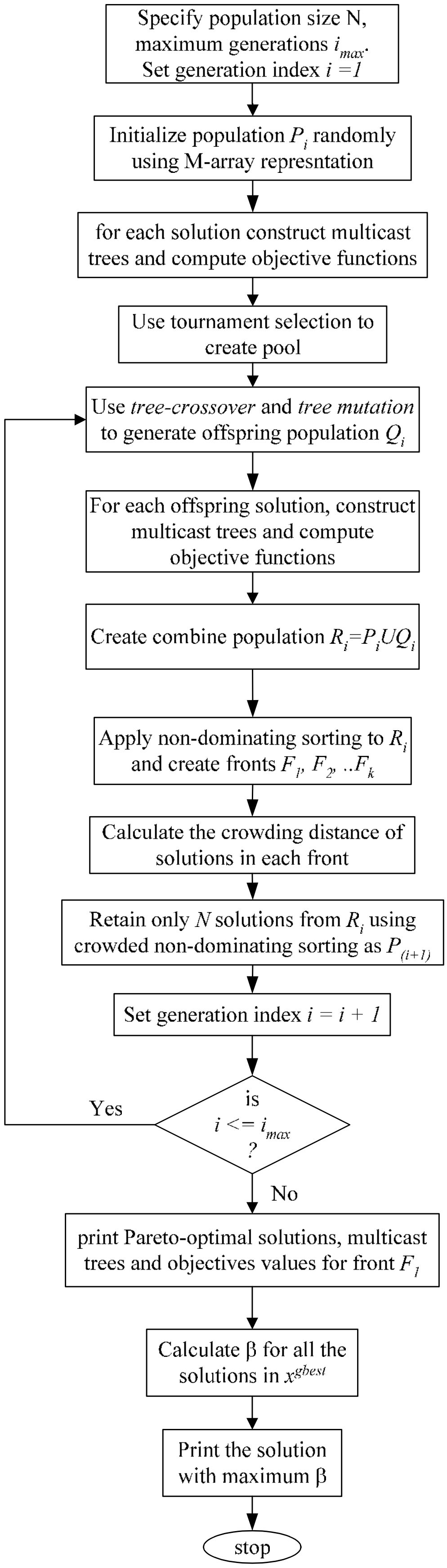


Fig. 6.5 Flowchart for tree-structured multiobjective genetic algorithm

6.3 TREE-STRUCTURED MULTIOBJECTIVE PARTICLE SWARM OPTIMIZATION

The particle swarm optimization has been reviewed in chapter-5 where the development of TSPSO was discussed. The following considerations are made in developing the tree-structured multiobjective particle swarm optimization (TSMPSO):

- There is no notion of offspring generation in PSO and therefore, the size of the swarm remains unchanged.
- The particles change their positions during the iterative search process. This search process results into optimal position.
- In PSO, the leaders such as x^{gbest} and x_i^{pbest} are used to guide the search instead of using selection as in genetic algorithm. The x^{gbest} is the global best position attained by entire swarm. Whereas, the x_i^{pbest} is the personal best position that has been attained by the individual particles. These leaders x^{gbest} and x_i^{pbest} get updated in each iteration.
- The multiobjective optimization should return multiple non-dominated solutions at the convergence of optimization process. Therefore, the $\mathbf{x}^{gbest}$ will be a set of solutions instead of a single solution. This set of solutions will be a set of nondominated solutions, which will get updated in each iteration. As all solutions in the $\mathbf{x}^{gbest}$ are equally good, any of the nondominated solution can be opted as leader.

$$x^{gbest} = rand(\mathbf{x}^{gbest}) \tag{6.1}$$

- In multiobjective optimization, the solutions are compared on the basis of nondominance. Therefore x_i^{pbest} is taken as the nondominated value between x_i and x_i^{pbest}. Therefore, x_i^{pbest} is representing one solution. Instead of managing a set of nondominated positions that the individual particles have acquired, this one solution nomenclature of x_i^{pbest} will help save the memory requirement.

$$x_i^{pbest} = \text{nondominate}(x_i, x_i^{pbest}) \tag{6.2}$$

Therefore, the equations for TSMPSO will have the same structure as given by eq. (5.1) and eq. (5.2). However, the leaders are selected by eq. (6.1) and eq. (6.2). The combined structure is shown as eq. 6.3 as -

$$v_i(t) = x_i(t-1) + w \times v_i(t-1) + C_1 \times r_1 \times (x_i^{pbest} - x_i) + C_2 \times r_2 \times (x^{gbest} - x_i) \qquad (6.3)$$

Where,

$$x_i^{pbest} = \text{nondominate}(x_i, x_i^{pbest})$$
$$x^{gbest} = rand(\mathbf{x}^{gbest})$$

- In TSPSO, updated values of particles (solutions) x_i are resulted by *tree-merging* of x_i, x^{gbest} and x_i^{pbest}.
- The *tree-mutation* is used to avoid the algorithm being trapped in the local minima. The TSMPSO algorithm at the convergence provides a set of Pareto-optimal solutions and the best compromise solution is obtained by fuzzy cardinal priority ranking.

With the above considerations, the steps of the algorithm are summarized as follows:

1. Specify swarm size, maximum iterations t_{max} and set iteration index $t = 1$.

2. Initialize the positions of the particles (solutions) randomly. As explained in *tree structured encoding* in section 5.2, each particle x_i is represented by M-ordered arrays. Where, each array indicates the random path from a destination node to source node.

3. Construct the multicast trees for each x_i and evaluate the particles using objective function.

4. Initialize the local best solution x_i^{pbest} as x_i.

5. Regard the nondominated solutions among the x_i^{pbest} as the current global best solutions x^{gbest}.

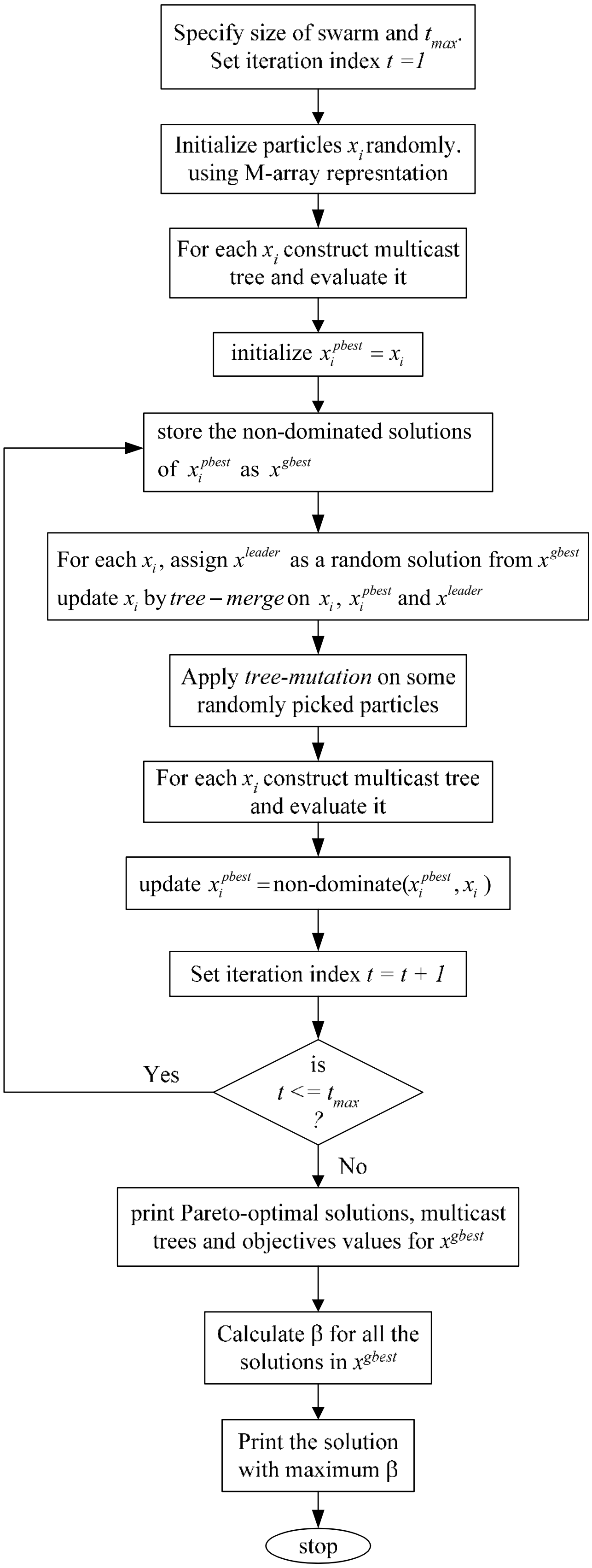


Fig. 6.6 Flowchart for tree-structured multiobjective particle swarm optimization

6. Update the position of each particle x_i by *tree-merge* operation on x_i , x_i^{pbest} and a randomly picked solution from nondominated set x^{gbest}. The *tree-merge* operation randomly picks M-arrays to yield new position of each particle x_i.

7. Attempt *tree-mutation* on randomly picked particles. In this a node is selected randomly in path from destination to source and the current sub-path between this randomly selected node and source is replaced by new random sub-path.

8. Construct the multicast trees for each x_i and evaluate the objective functions.

9. Update x_i^{pbest} as the nondominated value between x_i^{pbest} and x_i as per the following considerations :

   if x_i dominates x_i^{pbest} , $x_i^{pbest} = x_i$

   if x_i^{pbest} dominates x_i , $x_i^{pbest} = x_i^{pbest}$

   if x_i and x_i^{pbest} are non-dominated, x_i^{pbest} is picked randomly between the two.

10. Set iteration index $t = t+1$.

11. If ($t < t_{max}$) go to step 5, else return optimal multicast trees corresponding to x^{gbest}.

12. Compute normalized membership β for each Pareto-optimal solution in the front F_1. Select the solution with maximum β as the best compromise solution and stop.

6.4 MULTIOBJECTIVE MULTICAST ROUTING PROBLEMS

The multiobjective multicast routing has been attempted to investigate the cost-residual bandwidth optimization and cost-residual bandwidth-packet loss probability optimization under delay and delay jitter constraints. The formulations of these problems are discussed in section 3.4.2. The formulation is presented herewith for the ready reference.

Cost-Residual Bandwidth Optimization

The delay and delay jitter bound cost-residual bandwidth optimization is attempted to optimize twin conflicting objectives cost and residual bandwidth while satisfying delay and delay jitter bounds simultaneously. It is expressed as –

Optimize $f(\mathbf{x})$ =[minimize $f_1(\mathbf{x})$, maximize $f_2(\mathbf{x})]^T$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \le \Delta$ (6.4)

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i))\right) \le \delta$$

Where, $f_1(\mathbf{x})$ and $f_2(\mathbf{x})$ are expressed as -

$$f_1(\mathbf{x}) = C(T(s,M)) = \sum_{e \in T(s,M)} C(e) \quad (6.4a)$$

$$f_2(\mathbf{x}) = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)} \quad (6.4b)$$

Cost-Residual Bandwidth-Packet Loss Probability Optimization

The conflicting objectives cost, residual bandwidth and packet loss probability associated with tree are considered for multiobjective optimization while satisfying the delay and delay jitter bounds. The structure of the optimization problem is –

Optimize $f(\mathbf{x})$ = [minimize $f_1(\mathbf{x})$, maximize $f_2(\mathbf{x})$, minimize $f_3(\mathbf{x})]^T$

Subjected to $DB(T(s,M)) = \max_{d_i \in M}(D(P(s,d_i)) \le \Delta$ (6.5)

$$DJ(T(s,M)) = \left(\max_{d_i \in M}(D(P(s,d_i)) - \min_{d_i \in M}(D(P(s,d_i))\right) \le \delta$$

Where, $f_1(\mathbf{x})$, $f_2(\mathbf{x})$ and $f_3(\mathbf{x})$ are expressed as –

$$f_1(\mathbf{x}) = C(T(s,M)) = \sum_{e \in T(s,M)} C(e) \quad (6.5a)$$

$$f_2(\mathbf{x}) = RB(T(s,M)) = \frac{\sum_{e \in T(s,M)} \alpha(e)}{\sum_{e \in T(s,M)} \phi(e)} \quad (6.5b)$$

$$f_3(\mathbf{x}) = PL(T(s,M)) = \max_{d_i \in M}(PL(P(s,d_i))) \quad (6.5c)$$

6.5 RESULTS AND DISCUSSION

The developed tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSMPSO) are used to investigate multicast routing for cost-residual bandwidth optimization and for cost-residual bandwidth-packet loss probability optimization. The delay and delay jitter bounds are included for these multicast routing optimizations. The effectiveness of the developed algorithms has been tested on 15-node Bellcore network, 24-node USIP Backbone network and random networks. The effects of change in multicast group size have also been studied. The Pareto optimal solutions and correspondingly the best compromise solution have been obtained.

6.5.1 Cost-Residual Bandwidth Optimization

For the 15-node Bellcore network shown in Fig. 6.7, the link cost is randomly assigned integer value between 2 and 10, available bandwidth on links is assigned random integer between 70 Kbps and 170 Kbps while taking the link capacity as 200 Kbps. The link delay is randomly assigned integer values between 1 and 5. The optimization methods have been attempted for the source node as '0' and four destination nodes as 7, 9, 11, 14. The delay and delay jitter bounds are considered as 15 and 4 respectively.

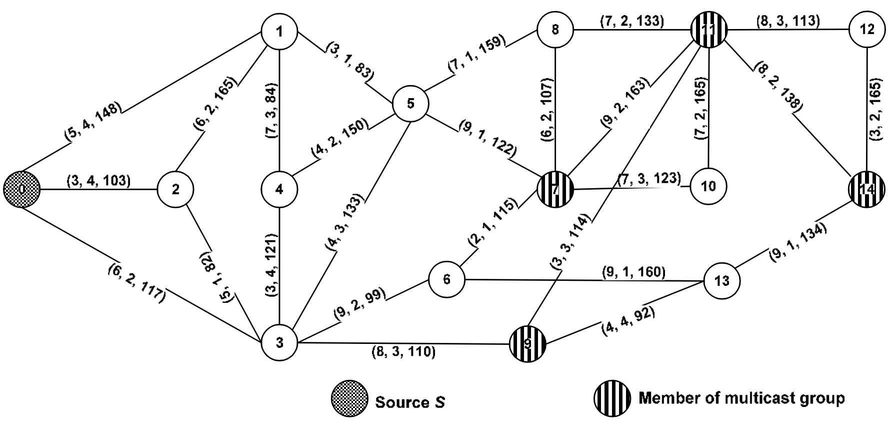


Fig. 6.7 A 15-node Bellcore network topology

The initial Pareto fronts that are resulted with TSMGA are shown in Fig. 6.8. The ranks of the "front 1", "front 2" and "front 3" are 1, 2 and 3 respectively. The tree cost and available bandwidth for these fronts are summarized in Table 6.1. The "front 1" is closest to the vertices whereas the "front 3" is farthest to the vertices.

Table 6.1 : Cost and available bandwidth for Initial Pareto fronts for 15-node Bellcore network

front 1		*front 2*		*front 3*	
Tree cost	*Residual Bandwidth*	*Tree cost*	*Residual Bandwidth*	*Tree cost*	*Residual Bandwidth*
39	63.92	45	60.00	46	60.79
41	65.14	46	63.63	47	62.38
51	66.72	50	64.67	49	62.78
53	68.28	51	64.88	53	62.90
		54	66.15	55	63.61
		56	66.20	57	65.10
		64	67.05	58	65.30
				69	65.42
				73	66.20

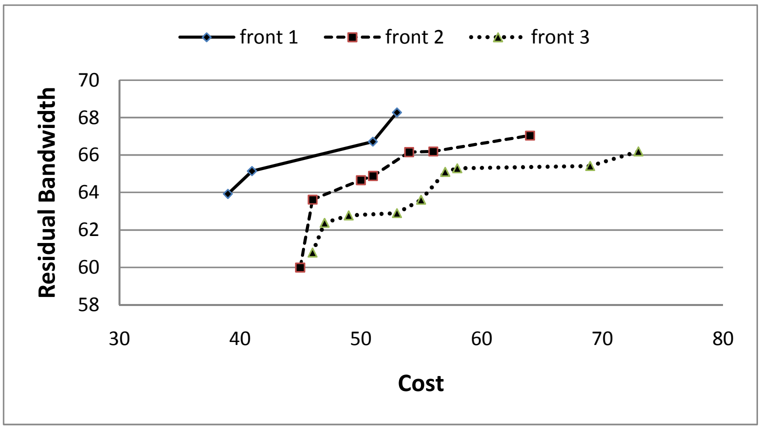
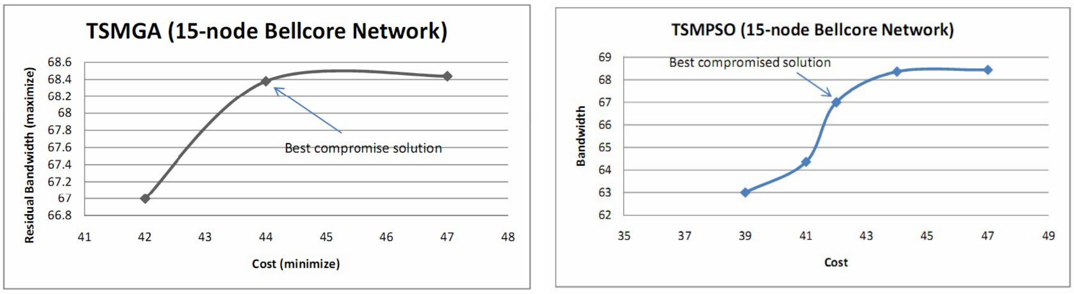


Fig. 6.8 Initial Pareto fronts for 15-node Bellcore network

The initialization procedure is similar for TSMGA and TSMPSO. However, TSPSO does not require non-dominating sorting and ranking. Therefore, the only *front 1*, marked in Fig. 6.8, will be resulted with TSMPSO.

The resulted Pareto optimal fronts as obtained using TSMGA and TSMPSO are shown in Fig. 6.9(a) and Fig. 6.9(b) respectively. The best compromise solution is also marked in Fig. 6.9(a) and Fig. 6.9(b). It is observed that TSMPSO results into five solutions in the Pareto optimal front whereas TSMGA results into three solutions in Pareto optimal front. The spread in Pareto front for TSMPSO is large. The solution structure and the encoded tree as obtained from TSMPSO are shown in Fig. 6.10(a) and corresponding optimal multicast tree on the network topology is presented in Fig. 6.10(b).



(a) performance of TSMGA (b) performance of TSMPSO

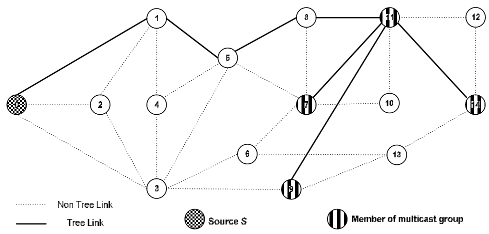
Fig. 6.9 Pareto optimal front for 15-node Bellcore network

Best Compromised Solution Structure

7	11	8	5	1	0		
9	11	8	7	6	3	2	0
11	12	14	13	9	3	0	
14	11	8	7	5	1	2	0

Best Compromised Tree Structure

7	11	8	5	1	0
9	11				
14	11				



(a) encoded best compromised solution and tree **(b)** Best compromise optimal tree

Fig. 6.10 Representation of optimal compromised solution and multicast tree

The cost-residual bandwidth optimization has been attempted using TSMGA and TSMPSO for delay and delay jitter bounds as 30 and 15 respectively. The performance has been simulated for source node s as '0' and two multicast groups as {7,10,15} and {7,10,15,19,23}. The Pareto-optimal fronts resulted for {7,10,15} are shown in Fig. 6.11 and the Pareto optimal fronts resulted for multicast group {7,10,15,19,23} are shown in Fig. 6.12. For the multicast group {7,10,15} identical front is resulted by both. For the multicast group {7,10,15,19,23}, an additional point is resulted with TSMPSO, the spread is not significant. Even, the same best compromised solution is obtained for both the fronts. The best compromised trees for these multicast groups are illustrated in Fig. 6.13(a) and Fig. 6.13(b) respectively. For the compromised tree shown in Fig. 6.13(a), the cost and residual bandwidth values are 38.39 and 70.03 respectively. The cost and residual bandwidth are obtained as 62.03 and 70.02 respectively for compromised tree shown in Fig. 6.13(b).

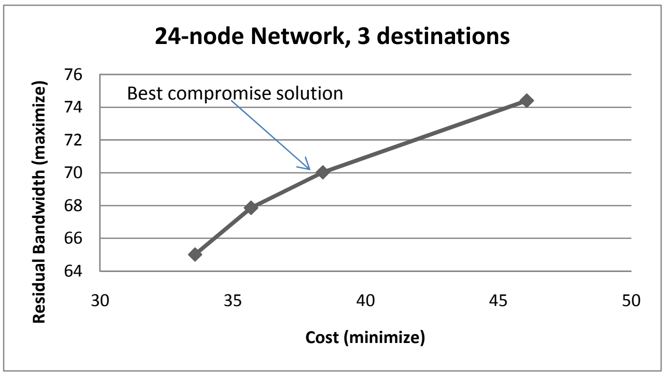


Fig. 6.11 The Pareto-optimal front for multicast group {7,10,15}

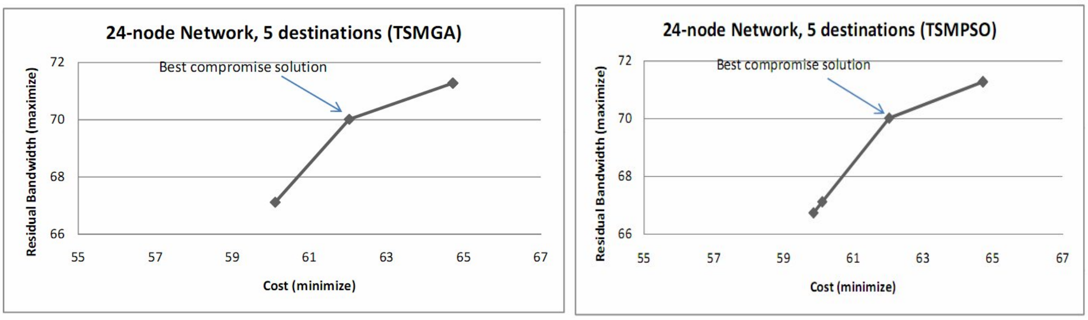
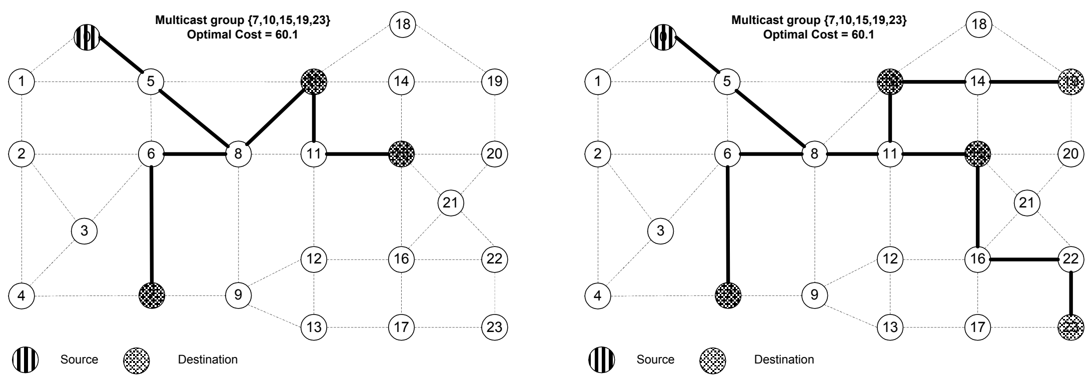


Fig. 6.12 The Pareto-optimal front for multicast group {7,10,15,19,23}



(a) For multicast group {7,10,15} (b) For multicast group {7,10,15,19,23}

Fig. 6.13 The optimal multicast trees for 24-node USIP Backbone Network

The study is carried out on 100-node random network, whose details are given in Appendix-A. The study is carried out with delay and delay jitter bounds as 100 and 25 respectively. The population size N is taken as 200. The simulation is attempted for the random multicast group size of 10%.

The Pareto optimal front obtained from TSMGA is shown in Fig. 6.14. For comparison, the initial "*front 1*" resulted with initial population is also represented. As expected, the Pareto optimal front is closest to the vertices. The Pareto-optimal front resulted from TSMPSO is shown in Fig. 6.15.

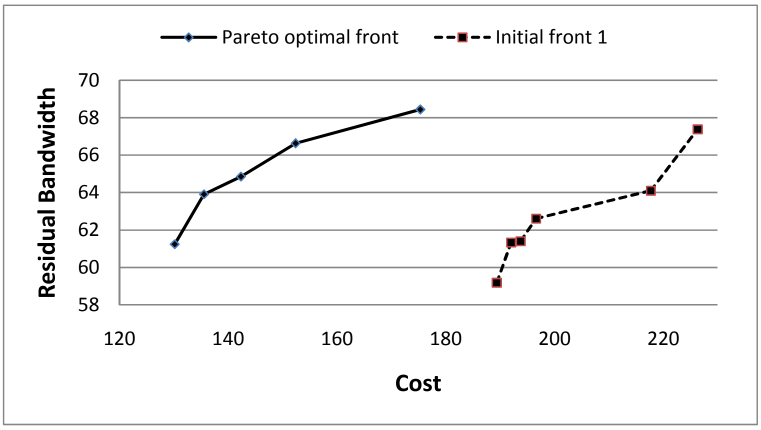


Fig. 6.14 Pareto optimal front and initial Pareto "front 1" through TSMGA

Fig. 6.15 Pareto-optimal front using TSMPSO for 100 node network

The data for Pareto optimal fronts obtained from TSMGA and TSMPSO is summarized in Table 6.2 and these fronts are presented in Fig. 6.16 for comparison. The Pareto optimal front obtained from TSMPSO is having wide spread and also closer to the vertices.

Table 6.2 : Summary of Pareto optimal fronts from TSMGA and TSMPSO for 100-node network

TSMGA		**TSMPSO**	
Tree cost	*Residual Bandwidth*	*Tree cost*	*Residual Bandwidth*
130.2	61.2	115.2	60.9
135.6	63.9	121.6	62.0
142.4	64.9	122.4	64.8
152.4	66.6	132.9	66.2
175.4	68.4	144.2	66.3
		147.3	66.8
		155.5	66.9
		162.6	67.4
		166.1	67.8

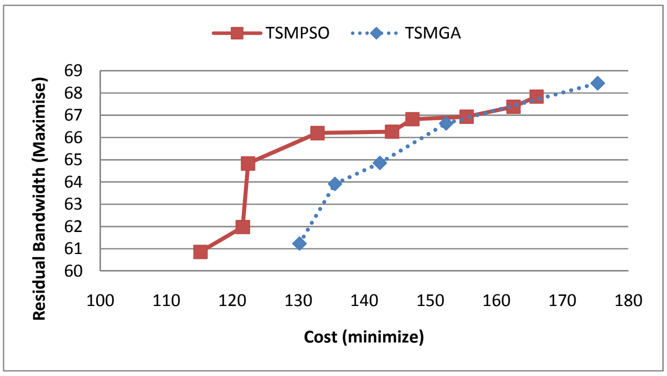


Fig. 6.16 Pareto optimal fronts from TSMGA and TSMPSO for 100-node network

The effect of the change in the multicast group size on the convergence time has been studied with both TSMGA and TSMPSO on 100-node network. Correspondingly, the results are summarized in Table 6.3 and are depicted in Fig. 6.17. In this study, the multicast group size is varied and the destination nodes forming the multicast group are selected randomly. The other parameters are kept unchanged. The convergence time increases as the size of the multicast group increases and both TSMGA and TSMPSO exhibit near linear relationship. For all such cases, the TSMPSO is showing slightly faster convergence.

Table 6.3 : Summary of convergence time for different sizes of multicast group

Multicast group size	Time (sec) TSMPSO	Time (sec) TSMGA
5	1.53	2.34
10	2.98	4.32
15	6.0	6.04
20	5.7	6.98
25	8.69	10.05
30	9.95	11.57

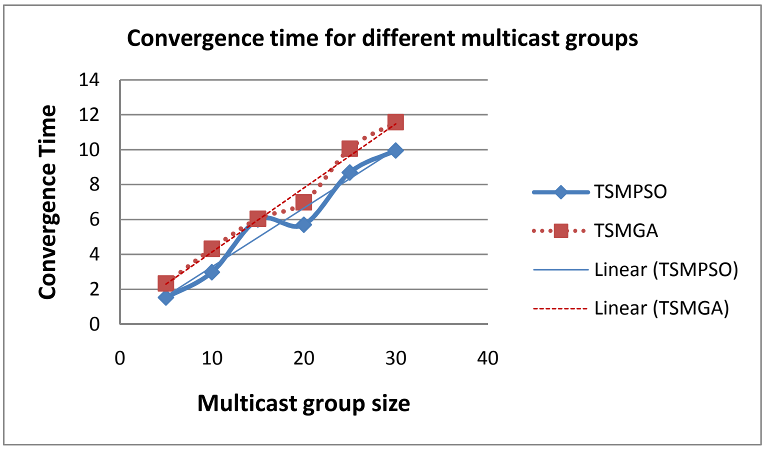


Fig. 6.17 Effect of multicast group on the convergence time on 100 node network

The convergence time resulted by TSMGA and TSMPSO for varying sizes of the random networks is summarized in Table 6.4 and the variation is shown in Fig. 6.18. The random networks are obtained using Waxman model as per the procedure explained for 100-node network. For each network, the multicast group size is kept as 20%. The delay and delay jitter are taken as 100 and 25 respectively. The convergence time can be approximated as exponential relation with the network size.

Table 6.4 Summary of convergence time for different sizes of random networks

Network size	Time (sec) TSMPSO	Time (sec) TSMGA
25	0.34	0.44
50	1.48	1.57
75	3.37	3.89
100	5.89	6.98
125	10.58	14.43
150	18.37	22.80

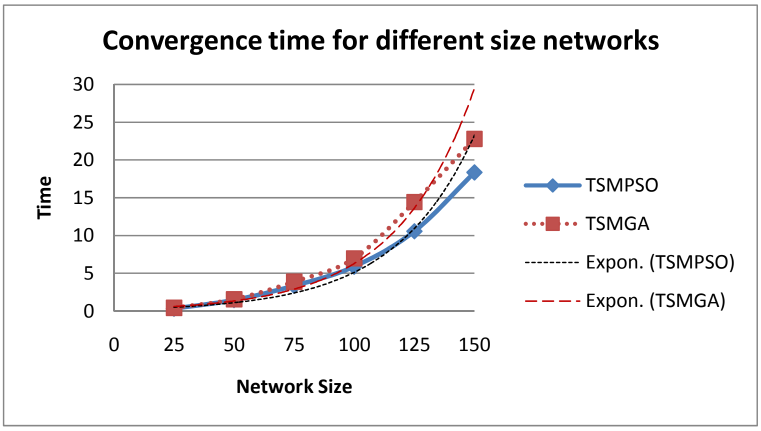


Fig. 6.18 The effect of network size on convergence time

6.5.2 Cost-residual bandwidth-packet loss probability optimization

The effectiveness of TSMGA and TSMPSO has been studied to solve tri-objective cost-residual bandwidth-packet loss probability optimization. The tree cost is minimized, residual bandwidth is maximized and the packet loss probability is minimized. The performance is studied on 24-node USIP Backbone network and a 100-node network.

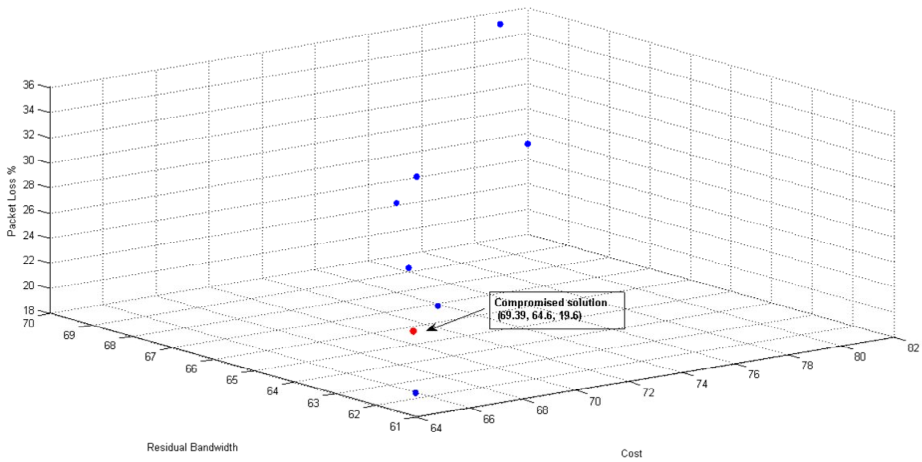
For the 24 node network, the multicast group of five destination nodes {7,10,15,19,23} has been considered while source node is selected as 0. The delay and delay jitter bounds are considered as 30 and 10 respectively. The set of Pareto-optimal solutions resulted from TSMGA and TSMPSO are summarized in Table 6.5. The Pareto-fronts are sorted on Packet loss probability for clarity of analyzing them.

These Pareto-optimal solutions corresponding to TSMGA and TSMPSO are illustrated in objective space as Fig. 6.19(a) and Fig. 6.19(b) respectively. The best-compromised solution obtained from fuzzy cardinal priority ranking is specified in the Table 6.5 and also marked in Fig. 6.19. As the system is identical, some solutions are bound to be common from TSMGA and TSMPSO. The best compromised optimal solutions are different, although these solutions are resulted from both the algorithms. This is because, the best compromised solution depends on the spread of the front and the normalized membership function. As the spread of solutions from TSMPSO is large, the different solution is obtained as best compromised solution.

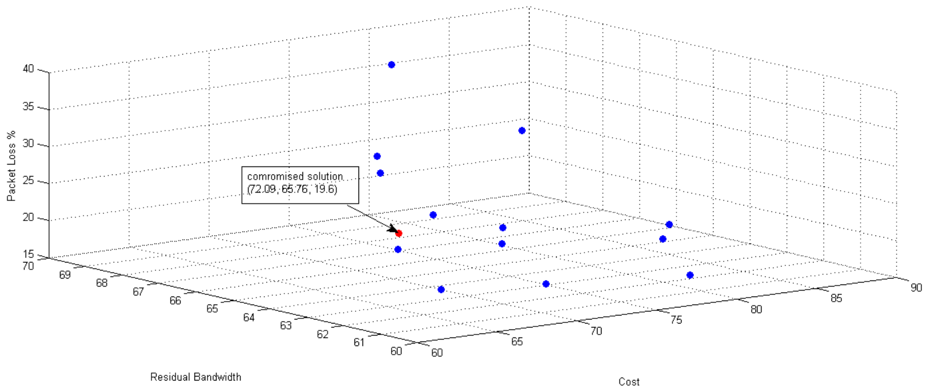
Table 6.5 : The set of Pareto-optimal solutions for 24-node network

TSMGA			TSMPSO		
Cost	**Residual bandwidth**	**Packet loss probability**	**Cost**	**Residual Bandwidth**	**Packet loss probability**
64.24	61.19	19.6	78.87	60.78	17.68
69.39	**64.6**	**19.6**	84.02	63.76	17.68
72.09	65.76	19.6	86.72	64.77	17.68
79.77	68.56	27.21	70.23	60.95	18.85
69.69	65.21	29.13	75.39	64.38	18.85
72.38	66.47	29.13	78.08	65.54	18.85
64.53	61.55	29.13	64.24	61.19	19.6
80.07	69.43	35.83	69.39	64.6	19.6
			72.09	**65.76**	**19.6**
			85.77	68.36	26.52
			64.53	61.55	29.13
			69.69	65.21	29.13
			72.38	66.47	29.13
			80.07	69.43	35.83

To further understand the Fig. 6.19, the Pareto optimal solutions for TSMPSO are plotted between residual bandwidth and cost in Fig. 6.20 for various values of packet loss probability. For each value of packet loss, the set of residual-bandwidth and cost is resulting as Pareto front. The similar fronts are obtained for bi-objective problem discussed in section 6.5.1.



(a) solutions from TSMGA



(b) solutions from TSMPSO

Fig. 6.19 Pareto-optimal solutions in objective space for 24-node network

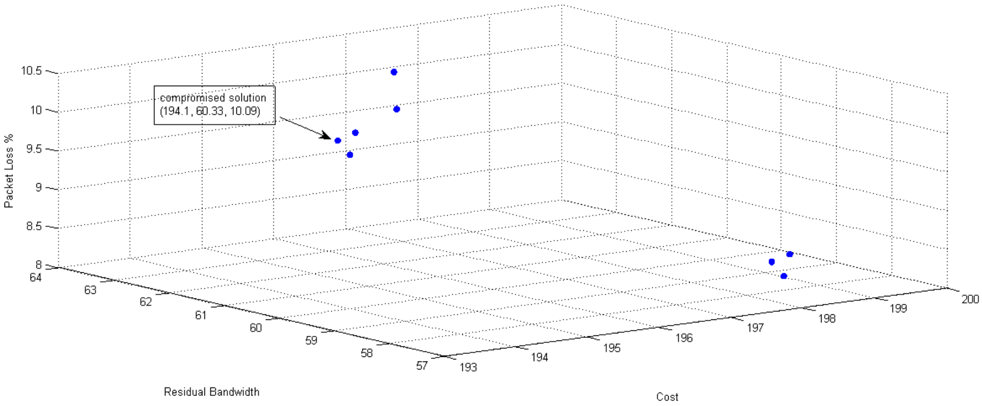
Fig. 6.20 Illustration of Pareto-optimal solutions for Fig. 6.19(b) as cost-residual bandwidth characteristics for different packet-loss probability

The performance cost-residual bandwidth-packet loss probability optimization is carried out on 100 node network with the multicast group size of 20 i.e. 20% of the network size. The simulation using TSMGA and TSMPSO is carried out considering delay bound of 50 and delay jitter 25. The obtained results are summarized in Tables 6.6. Correspondingly the solutions are presented in objective space as Fig. 6.21(a) and Fig. 6.21(b) respectively. The TSMPSO results into large Pareto-optimal set and characterizes large spread compared to TSMGA.

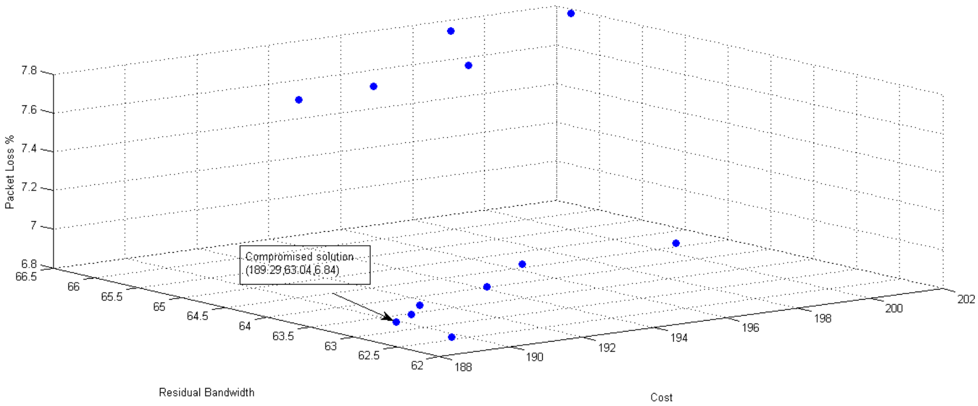
Table 6.6 : The set of Pareto-optimal solutions for 100-node network

TSMGA			**TSMPSO**		
Cost	**Residual bandwidth**	**Packet loss probability**	**Cost**	**Residual Bandwidth**	**Packet loss probability**
198.18	57.61	8.28	189.12	62.33	6.84
198.61	58.39	8.28	**189.29**	**63.04**	**6.84**
199.06	58.65	8.28	190.17	63.23	6.84
193.64	59.55	10.09	191.02	63.49	6.84
194.07	**60.33**	**10.09**	193.59	63.79	6.84
194.52	60.6	10.09	195.92	64.35	6.84
195.77	61.48	10.09	200.22	64.36	6.84
197.16	63.34	10.09	190.66	64.75	7.78
			192.99	64.86	7.78
			196.26	65.12	7.78
			198.59	66.3	7.78
			201.98	66.32	7.78

The Fig. 6.21(a) and Fig. 6.21(b) are represented as bi-objective representation showing variation between Residual bandwidth and cost for different packet loss probability values as Fig. 6.22(a) and Fig. 6.22(b) respectively. The respective best compromised solutions are also marked in Fig 6.22(a) and Fig 6.22(b).



(a) solutions using TSMGA



(b) solutions using TSMPSO

Fig. 6.21 Pareto-optimal solutions in objective space for 100-node network

(a) TSMGA

(b) TSMPSO

Fig. 6.22 Illustration of Pareto-optimal solutions for Fig. 6.21 as cost-residual bandwidth characteristics for different packet-loss probability

6.6 CONCLUDING REMARKS

The tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSMPSO) method are presented to determine optimum multiobjective multicast routing. The topological assisted ordered M-array structure is used to represent the multicast tree. The solution structure is preserved during the simulation. Using these developed methods, bi-objective cost-residual bandwidth and tri-objective cost-residual bandwidth-packet loss probability optimizations are investigated under delay and delay-jitter constraints. The effectiveness of these proposed algorithms is tested on various networks and the best compromised solution is obtained using fuzzy cardinal priority ranking. The following conclusions are drawn from the study –

- The developed algorithms TSMGA and TSMPSO, which are emulating elitist multiobjective evolutionary approach, are providing Pareto-optimal solutions with good diversity. These algorithms are flexible to handle multiple objectives and constraints. Due to the typical tree-structured nature of the problem, continuous Pareto front is not possible.

- Both TSMGA and TSMPSO are resulting different set of Pareto-optimal solutions for the same network. However, few solutions may be common in the Pareto-optimal sets resulted by TSMGA and TSMPSO. Depending on the spread and the optimal solutions, the best compromised solution may differ. However, both these solutions will be non-dominating to each other.

- It is observed that TSMPSO is resulting into a set of Pareto optimal solutions with better spread.

- For both TSMGA and TSMPSO, the convergence time increases almost linearly with the increase in the size of multicast group. However, the TSMPSO is showing faster convergence.

- For both TSMGA and TSMPSO, the convergence time increases almost exponentially with the increase in the size of network. However, the TSMPSO is showing faster convergence.

CONCLUSIONS AND SCOPE FOR FUTURE WORK

7.1 GENERAL

The use of QoS multicast routing has become imperative for sending the information to a subset of the users for resource intensive computer networks. The various QoS parameters such as cost, bandwidth utilization, residual bandwidth, delay, delay jitter and packet loss probability etc. are in conflict and therefore the routing algorithm has to be attempted with multiobjective formulation. The investigations carried out on the QoS multicast routing have been directed to develop multicast routing algorithms for constructing multicast tree for various combinations of constraints and objectives. The multicast routing algorithms have been developed using Hopfield neural network (HNN) and population based search and optimization algorithms namely genetic algorithm and particle swarm optimization. The main contributions of this thesis include –

- Development of multicast routing for optimum cost, optimum residual bandwidth and multiobjective optimum cost-residual bandwidth using Hopfield neural network described in Chapter-4.
- Development of tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) for QoS constrained multicast routing described in Chapter-5.
- Development of tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSMPSO) method for QoS constrained multiobjective multicast routing described in Chapter-6.

The main findings of the research work carried out in this thesis are summarised in the following section.

7.2 SUMMARY OF IMPORTANT FINDINGS

In Chapter-4, the Hopfield neural network based approaches are used to obtain multicast tree for minimum cost, optimal residual bandwidth and optimum cost-residual bandwidth optimization. The respective optimization method has been investigated without and with delay bound. The cost and residual bandwidth optimization has been carried out using shortest path tree formulation and multicast tree formulation. The multiobjective cost-residual bandwidth optimization has been attempted using multicast tree formulation. For the multiobjective optimization, weighted sum approach is used to form the energy function. The study has been carried out on 24-node USIP Backbone network for different multicast groups. The following conclusions are drawn from the investigations-

- The choice of Hopfield neural network parameters participating in energy function, is very decisive and the optimal solution depends on these parameters. These coefficients should be selected such that they impose penalty to non-existing link being included, to ensure the construction of a continuous path, to ensure the construction of virtual link from destination to source and the constraint satisfaction.

- Both shortest path tree and multicast tree approaches are forming the multicast tree and able to handle delay constraint. However, the multicast tree approach yields slightly better results due to the consideration of tree metric instead of path metric.

- With the increase in size of multicast groups, the tree cost increases and the residual-bandwidth value decreases slightly.

- The inclusion of delay bound forces the algorithm to take paths satisfying the delay bound constraint. For constraints violation, the neuron output gets penalized and the HNN stabilizes to other path. This is resulting into higher cost and reduced bandwidth.

- For multiobjective cost-residual bandwidth optimization, each combination of weights is not always result into different solutions. This is due to the tree-structured nature of the problem. Due to this tree-structured nature, limited Pareto-optimal solutions are obtained.

- The Hopfield neural network requires mapping of the optimization problem. In the mapping of the routing problem, the tree formulation is implied and therefore it does not require any encoding scheme as needed for population based search and optimization algorithms.

- The large convergence time and iterations are needed with HNN in forming the multicast tree. The algorithms also exhibits tendency of being trapped in local minima when multicast routing problems are obtained for large network.

In Chapter-5, a tree-structured encoding scheme is presented and two algorithms namely tree-structured genetic algorithm (TSGA) and tree-structured particle swarm optimization (TSPSO) method are proposed to obtain the optimal multicast tree for QoS constrained minimum cost and QoS constrained optimal residual bandwidth optimization. The end-to-end delay and delay jitter are considered as the QoS constraints. The constraints are taken care by adding penalty terms to the objective function. The study has been carried out on 24-node USIP Backbone network and various random networks. The following conclusions are drawn from the investigations-

- The proposed encoding scheme representing the solution as ordered M-array structure is effective in providing loop free tree. The effectiveness of this encoding has been tested through TSGA and TSPSO.
- Both the developed algorithms TSGA and TSPSO are effective in handling both delay bound and delay jitter bound for minimum cost and optimum residual bandwidth routing. These formulations are cable to handle complexity arising due to different routing objectives and constraints.
- With the inclusion of constraints, the cost of resulting optimal tree increases. Whereas the residual bandwidth decreases with the inclusion of constraints.
- The TSPSO method yields the optimal tree that is having lower cost compared to the optimal tree resulted by TSGA. This suggests that TSPSO forms the tree through low cost paths which are resulting into delay and delay jitter close to the limiting values.
- The TSPSO method yields the optimal tree having slightly higher residual bandwidth.
- The convergence time for both TSGA and TSPSO increases with the increase in

size of multicast group and the size of the network.

- Due the process of updating the leader x_i^{pbest}, which inherently include elitism, the formulation of TSPSO is simple and its convergence is fast.

In Chapter-6, the tree-structured multiobjective genetic algorithm (TSMGA) and tree-structured multiobjective particle swarm optimization (TSMPSO) method are presented to understand the multiobjective multicast routing. These algorithms employ the concept of non-dominance for both objectives and constraints. The topological assisted ordered M-array structure is used to represent the multicast tree. The solution structure is preserved during the simulation. Using these developed methods, bi-objective cost-residual bandwidth and tri-objective cost-residual bandwidth-packet loss probability optimizations are investigated under delay and delay-jitter constraints. The effectiveness of these proposed algorithms is tested on various networks and the best compromised solution is obtained using fuzzy cardinal priority ranking. The following conclusions are drawn from the study –

- The developed algorithms TSMGA and TSMPSO, which are emulating elitist multiobjective evolutionary approach, are providing Pareto-optimal solutions with good diversity. These algorithms are flexible to handle multiple objectives and constraints. Due to the typical tree-structured nature of the problem, continuous Pareto front is not possible.

- Both TSMGA and TSMPSO are resulting different set of Pareto-optimal solutions for the same network. However, few solutions may be common in the Pareto-optimal sets resulted by TSMGA and TSMPSO. Depending on the spread and the optimal solutions, the best compromised solution may differ. However, both these solutions will be non-dominating to each other.

- It is observed that TSMPSO method results into a set of Pareto optimal solutions with better spread.

- For both TSMGA and TSMPSO, the convergence time increases almost linearly with the increase in the size of multicast group. However, the TSMPSO is showing faster convergence.

- For both TSMGA and TSMPSO, the convergence time increases almost exponentially with the increase in the size of network. However, the TSMPSO is showing faster convergence.

7.3 SCOPE FOR FURTHER WORK

The research work is a continuous process. An end of a research project is a beginning to a lot of other avenues for further work. As a consequence of the investigations carried out in this thesis on multicast routing and multiobjective multicast routing algorithms using Hopfield neural network (HNN) and population based search and optimization methods namely genetic algorithm and particle swarm optimization, the basic objectives have been brought to a successful conclusions and the following aspects are identified for further research work in this area.

- Work for QoS multicast routing has been presented for the static networks, where nodes including the source and multicast group remain unchanged. There is a scope to investigate the effectiveness of the methodologies for the dynamic networks, wireless and mobile ad-hoc networks.

- The source specific tree formation has been investigated with only one source node. There is a scope to investigate the routing tree formulation for multi-source multi-destination formulation.

- The performance of the HNN model depends on the judicious choice to the model coefficients, which is a difficult task. The methodology to decide these coefficients or a methodology independent to the model coefficients shall be investigated for multicast routing. The mechanism to accelerate the convergence of HNN for multicast routing may be investigated for its suitability to large network.

- The solution of multiobjective optimization based on HNN has been attempted using weighted sum approach. The method based on HNN and combining the concept of Pareto nondominance can be investigated.

- The solution in the population based search and optimization based algorithms has been represented as ordered M-arrays where each array is representing a random path between source and destinations. This formulation is found effective with the developed single-objective multicast routing algorithms TSGA and TSPSO and the developed multiobjective multicast routing algorithms TSMGA and TSMPSO. The other efficient scheme to represent the multicast tree can be investigated.

LIST OF PUBLICATIONS

International Journals

- Sushma Jain and J.D. Sharma, “Tree Structured Encoding based Multiobjective Multicast Routing Algorithm”, International Journal of Physical Sciences, Vol. 7, No.10, pp. 1622-1632, 2012..

- Sushma Jain and J.D. Sharma, “QoS Constraints Multicast Routing for Residual Bandwidth Optimization using Evolutionary Algorithm”, International Journal of Computer Theory and Engineering, Vol. 3, No. 2, pp. 211-216, 2011.

- Sushma Jain and J.D. Sharma, “Delay Bound Multicast Routing Using Hopfield Neural Network”, International Journal of Computer Theory and Engineering, Vol. 2, No. 3, pp 384-389, 2010.

International Conferences

- Sushma Jain and J.D. Sharma “Delay Bound Multicast Routing Algorithm Using Evolutionary Programming”, IEEE International Conference on Computer Network, ICON 2008, December 2008, New Delhi.

REFERENCES

Ahn C.W. and Ramakrishna R.S. (2002), "A genetic algorithm for shortest path routing problem and the sizing of populations", IEEE Transactions on Evolutionary Computation, vol. 6, no. 6, pp. 566-579.

Ahn C.W. and Ramakrishna R.S. (2004), "QoS Provisioning Dynamic Connection-Admission Control for Multimedia Wireless Networks Using a Hopfield Neural Network", IEEE Transactions on Vehicular technology, vol. 53, no. 1, pp. 106-117.

Ali M.K. and Kamoun F. (1993), "Neural networks for shortest path computation and routing in computer networks", IEEE Transactions on Neural Networks, vol. 4, no. 6, pp 941-953.

Alrabiah T. and Znati T. (2001), "Delay-constrained, low-cost multicast routing in multimedia networks", Journal of Parallel and Distributed Computing, vol. 61, vol. 9, pp. 1307-1336.

Araújo A.F.R. and Garrozi C. (2010), "MulRoGA: A multicast routing genetic algorithm approach considering multiple objectives", Applied Intelligence, vol. 32, no. 3, pp. 330-345.

Araújo F., Ribeiro B. and Rodrigues L. (2001). A neural network for shortest path computation. IEEE Transactions on Neural Networks, 12, 1067–1073.

Awerbuch B., Barnoy A. and Gopal M. (1991), "Approximate distributed Bellman-Ford algorithms", Proceedings of IEEE INFOCOM, pp. 1206-1213.

Balakrishnan S., Kannan P.S., Aravindan C. and Subathra P. (2003), "On-line emission and economic load dispatch using adaptive Hopfield neural network", Applied soft computing, vol. 2, no. 4, pp. 297-305.

Ballardie A., Crowcroft J. and Francis P. (1993), "Core Based Trees (CBT) - an architecture for scalable interdomain multicast routing", ACM SIGCOMM pp.85-95.

Barathkumar K. and Jaffe J. (1983), "Routing to multiple destinations in computer networks", IEEE Transactions on Communications, vol. 31, no. 3, pp. 343-351.

Bauer F. and Varma A. (1996), "Distributed algorithms for multicast path setup in data networks", IEEE/ACM Transactions on Networking, vol. 4, no. 2, pp. 181-191.

Bertsekas D. and Gallager R. (1992), "Data networks", Prentice-Hall, 2nd ed.

Boeringer D.W. and Werner D.H. (2004), "Particle swarm optimization versus genetic algorithms for phased array synthesis", IEEE Transactions on Antennas and Propagation, vol. 52, no. 3, pp. 771- 779.

Cai Z. and Wang Y. (2006), "A Multiobjective Optimization-Based Evolutionary Algorithm for Constrained Optimization", IEEE Transactions on Evolutionary Computation, vol. 10, no. 6, pp. 658-675,.

Casner S. and Deering S. (1992), "First IETF Internet audiocast",. ACM Computer Communications Review, vol. 22, no. 3, pp. 92-107.

Chakraborty D., Chakraborty G. and Shiratori N. (2003), "A dynamic multicast routing satisfying multiple QoS constraints", International Journal of Network management, vol.13, pp. 1-15.

Chen R.M. (2011), "Reducing network and computation complexities in neural based real-time scheduling scheme", Applied Mathematics and Computation, vol. 217, no. 13, pp. 6379–6389.

Chen S. and Nahrstedt K. (1998), "An overview of quality-of-service routing for the next generation high-speed Networks", problems and solutions. IEEE Network. vol. 12, no. 6, pp. 64-79.

Coello C. A. and Lechuga S. M. (2002), "MOPSO: A Proposal for Multiple Objective Particle Swarm Optimization", Proceedings of the Congress on Evolutionary Computation, IEEE Press, vol. 1, pp. 1051–1056.

Coello C.A.C., Pulido G.T. and Lachuga M.S. (2004), "Handling multiple objectives with particle swarm optimization", IEEE Transactions on Evolutionary Computation, vol. 8, no. 3, pp. 256-279.

Dauer J. P. and Stadler W. (1986), "Survey of Vector Optimization in infinite-Dimensional Spaces", Part-II. Journal of Optimization Theory and Applications, vol. 51, no. 2, pp. 205–241.

Deb K. (2001), "Multiobjective optimization using evolutionary algorithms", John Wiley & Sons.

Deb K., Pratap A., Agarwal S. and Meyarivan T. (2002), "A fast elitist multi-objective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation, vol. 6, no. 2, pp. 182-197.

Deering S. (1996), "The PIM architecture for wide-area multicast routing", IEEE/ACM Transaction on Networking, vol. 4, pp.153-162.

Deering S. and Cheriton D. (1990), "Multicast routing in datagram internetworks and extended LANs", ACM Transactions on Computer Systems, vol. 8, no. 2, pp. 85-110.

Deo N. and Pang C.Y. (1984), "Shortest-path algorithms: taxonomy and annotation", Networks, vol. 14, no. 2, pp. 275–323.

Divakaran D.M., Moorthy H.A. and Gonsalves T.A. (2005), "Effective bandwidth utilization for multimedia data over low bandwidth links", Proceedings of National Conference on Communication", IIT Madras.

Eberhart R.C. and Shi Y. (1998), "Comparison between genetic algorithms and particle swarm optimization", Proceedings of the seventh annual conference on Evolutionary Programming (Springer-Verlag), pp. 611–616.

Elbeltagi E., Hegazy T. and Grierson D. (2005), "Comparison among five evolutionary-based optimization algorithms", Advanced Engineering Information, vol. 19, no. 1, pp. 43– 53.

Farina M. and Amato P. (2004), "A fuzzy definition of optimality for many criteria optimization problems", IEEE Transactions on System, Man and Cybernetics, vol. 34, no. 3, pp. 315–326.

Feng B. and Douligers C. (2001), "A neural network method for minimum delay routing in packet-switched networks", Computer Communications, vol. 24, no. 10, pp. 933-941.

Feng G. and Yum T.S.P. (1999), "Efficient multicast routing with delay constraints", International Journal of Communication Systems, vol. 12, pp 181-195.

Fogel L.J., Owens A.J. and Walsh M.J. (1966), "Artificial Intelligence through Simulated Evolution", New York: John Wiley.

Fonseca C. M. and Fleming P. J. (1993), "Genetic Algorithms for Multiobjective Optimization: Formulation, Discussion and Generalization. In Genetic Algorithms",: Proceedings of Fifth International Conference, Published by Morgan Kaufmann, pp. 416–423.

Fonseca C.M. and Fleming P.J. (1998), "Multiobjective optimization and multiple constraint handling with evolutionary algorithms", IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, vol. 28, no. 1, pp. 26-77.

Fuller R. and Carlsson C. (1996), "Fuzzy multiple criteria decision making: Recent developments", Fuzzy Sets and System, vol. 78, no. 2, pp. 139–153.

Gallager R., Humblet P. and Spira P. (1983), "A distributed algorithm for minimum weight spanning trees", ACM Transactions on Programming Languages & Systems, vol. 5, no. 1, pp. 66-77.

Garey M. and Johnson D. (1979), "Computers and intractability : A Guide to the theory of NP-completeness", New York: W.H. Freeman and Co.

Gelenbe E, Ghanwani A. and Srinivasan V. (1997), "Improved Neural Heuristics for Multicast Routing", IEEE Journal on Selected Areas in Communication, vol. 15, no. 2, pp. 147-155.

Ghanwani A. (1998), "Neural and delay based heuristics for the Steiner problem in networks", European Journal of Operational Research, vol. 108, no. 2, pp. 241-265.

Gilbert E.N. and Pollak H.O. (1968), "Steiner minimal trees", SIAM Journal of Applied Math, vol. 16, no,1 pp. 1-129.

Goldberg D. (1989), "Genetic algorithm in search, optimization & machine learning", Addison Wesley.

Gong Y.J., Zhang J., Liu Q., Huang R.Z., Chung H.S. and Shi Y.H. (2012), "Optimizing the Vehicle Routing Problem With Time Windows: A Discrete Particle Swarm Optimization Approach", IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, vol. 42, no. 2, pp. 254-267.

Gonsalves T.A. and Tobagi F.A. (1989), "Comparative performance of voice/data local area networks", IEEE Journal on Selected Areas of Communication, vol. 7, no. 5, pp. 657-669.

Haghighat A.T., Faez K., Dehghan M., Mowlaei A. and Ghahremani Y. (2004), "GA-based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing", Computer Communications, vol. 27, no. 1, pp. 111-127.

Hemminger T.L., Coulston C. and Pomalaza-Raez C.A. (2002), "Routing strategies for multicast packet radio networks", International Journal of Smart Engineering System Design, vol. 4, no. 3, pp. 215-223.

Ho S., Yang S., Ni G., Lo E. W. and Wong H. C. (2005), "A particle swarm optimization-based method for multiobjective design optimizations", IEEE Transactions on Magnetics, vol. 41, pp. 1756–1759.

Holland J.H. (1975), Adaptation in natural and artificial system, MIT press, Cambridge.

Hong S.M., Wang S.B. and Cai B.Y. (2003), "A bandwidth constrained QoS routing optimization algorithm", Proceedings of International Conference on Communication Technology, vol. 1, pp. 491 – 494.

Hopfield J.J. and Tank D.W. (1985), "Neural computation of decisions in optimization problems", Biological Cybernetics, vol. 52, pp 141-152.

Huang J., Du X., Yang Z. and Cheng W. (2003), "Available bandwidth-based real-time multicast routing distributed algorithm", International Conference on Computer Networks and Mobile Computing.

Huang L., Han H. and Hou J. (2007), "Multicast routing based on the ant system", Applied Mathematical Sciences, vol. 1, no. 57, pp. 2827-2838.

Hwang F. and Richards D. (1992), "Steiner tree problems", Networks, vol. 22, no. 1, pp. 55-89.

Ishibuchi H. and Murata T. (1998), "A multi-objective genetic local search algorithm and its application to flowshop scheduling", IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews. vol. 28, no. 3, pp. 392-403.

Jensen M.T. (2003), "Reducing the run-time complexity of multiobjective EAs: the NSGA-II and other algorithms", IEEE Transactions on Evolutionary Computation, vol. 7, no. 5, pp. 503-515.

Jia X. (1998), "A distributed algorithm of delay bounded multicast routing for multimedia applications in wide area networks", IEEE/ACM Transactions on Networking, vol. 6, no. 6, pp.828-837.

Jiang X. (1993), "Distributed path finding algorithm for stream multicast", Computer Communications, vol. 16, no. 12, pp. 767-775.

Kadirire J. (1994), "Minimising packet copies in multicast routing by exploiting geographic spread", Computer Communication Review, vol. 24, no. 3, pp. 47-62.

Kennedy J. and Eberhart R.C. (2001), "Swarm Intelligence", Morgan Kaufmann Publishers, San Francisco, California.

Kennedy M.P. and Chua L.O. (1987), "Unify the Tank and Hopfield linear programming circuit and the canonical programming circuit of Chua and Lin", IEEE Transactions on circuits and systems, vol. 34, no. 2, pp. 210-214.

Kompella V., Pasquale J. and Polyzos G. (1993), "Multicast routing for multimedia communication", IEEE/ACM Transactions on Networking, vol. 1, no. 3, pp. 286-292.

Kompella V., Pasquale J. and Polyzos G. (1993), "Two distributed algorithms for the constrained Steiner tree problem", Proceedings of International Conference on Computer Communications and Networking, pp. 343-349.

Kou L., Markowsky G. and Berman L. (1981), "A fast algorithm for Steiner trees", Acta Informatica, vol. 15, no. 2, pp. 141-145.

Kumar R., Mishra M. and Sarje A.K. (2010), "A simplified analytical model for end-to-end delay analysis in MANET", International Journal of Computer Application (IJCA) special issue on Mobile Ad-Hoc network, pp. 195-199.

Kun Z., Yong Q. and Hong Z. (2006), "Dynamic multicast routing algorithm for delay and delay variation-bounded Steiner tree problem", Knowledge Based Systems, vol. 19, no. 7, pp.554–564.

Leong W.F. and Yen G.G. (2008), "PSO-Based Multiobjective Optimization With Dynamic Population Size and Adaptive Local Archives", IEEE Transactions On Systems, Man, And Cybernetics—Part B: Cybernetics, vol. 38, no. 5, pp. 1270-1293.

Li B.B., Wang L. and Liu B. (2008), "An Effective PSO-Based Hybrid Algorithm for Multiobjective Permutation Flow Shop Scheduling", IEEE Transactions On Systems, Man, And Cybernetics—Part A: Systems And Humans, vol. 38, no. 4, pp. 818-831.

Liang C., Hong J.I., Yi L. and Xi L. (2011), "Multi-path routing based on load-balance for cognitive packet networks", The Journal of China Universities of Posts and Telecommunications, vol. 18, no. 5, pp. 71–75.

Lin X.H., Kwok Y.K. and Lau V.K.N. (2003), "A genetic algorithm based approach to route selection and capacity flow assignment", Computer Communications, vol. 26, no. 9, pp. 961-974.

Liu B., Wang L. and Jin Y.H. (2007), "An Effective PSO-Based Memetic Algorithm for Flow Shop Scheduling", IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, vol. 37, no. 1, pp. 18-27.

Mala C. and Rajagopalan N. (2011), "Simulated study of QoS multicast routing using particle swarm optimization", Worlds Applied Programming, vol. 1, no. 3, pp. 176-182.

Mallipeddi R. and Suganthan P.N. (2010), "Ensemble of Constraint Handling Techniques", IEEE Transactions on Evolutionary Computation, vol. 14, no. 4, pp. 561-579.

Masip-Bruin X., Yannuzzi M., Domingo-Pascual J., Fonte A., Curado M., Monteiro E., Kuipers F., Van Mieghem P., Avallone S., Ventre G., Aranda-Gutierrez P., Hollick M., Steinmetz R., Iannone L. and Salamatian K., (2006), "Research challenges in QoS routing", Computer Communications, vol. 29, no. 5, pp. 563-581.

Mehdi G. and Ali M. (2009), "Motion planning in order to optimize the length and clearance applying Hopfield neural network", Expert Systems with Applications, vol. 36, no. 3, pp. 4688–4695.

Mohemmed A.W., Sahoo N.C. and Geok T.K. (2008), "Solving shortest path problem using particle swarm optimization", Applied Soft Computing, vol. 8, no. 4, pp. 1643–1653.

Moy J. (1994), "MOSPF: analysis and experience RFC 1585"

Nesmachnow S., Cancela H. and Alba E. (2007), "Evolutionary algorithms applied to reliable communication network design", Engineering Optimization, vol. 39, no. 7, pp.831–855.

Noronha C. and Tobagi F. (1994), "Optimum routing of multicast streams", Proceedings of IEEE INFOCOM, vol.2, pp. 865-873.

Oliveira C.A.S. and Pardalos P.M. (2005), "A survey of combinatorial optimization problems in multicast routing", Computers and Operations Research, vol. 32, pp. 1953-1981.

Parsa M. and Zhu Q. (1998), "An iterative algorithm for delay-constrained minimum-cost multicasting", IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 461–474.

Parsopoulos K. E. and Vrahatis M. N. (2002), "Recent Approaches to Global Optimization Problems Through Particle Swarm Optimization", Natural computing, vol. 2, no. 3, pp. 235–306.

Paul P. and Raghavan S.V. (2002), "Surveys of multicast routing algorithms and protocols", 15th international Conference on Computer communication. pp. 902-926.

Pham N., Malinowski A. and Bartczak T. (2011), "Comparative Study of Derivative Free Optimization Algorithms", IEEE Transactions on Industrial Informatics, vol. 7, no. 4, pp. 592-600.

Pinto D., Barán B. and Fabregat R. (2005), "Multi-objective multicast routing based on ant colony optimization", Proceeding of Artificial Intelligence Research and Development, IOS Press Amsterdam, Netherlands, pp. 363-370.

Pornavalai C., Chakraborty G. and Shiratori N. (1995), "A neural network approach to multicast routing in real-time communication networks", in Proceedings of International Conference on Network Protocols, pp. 332 – 339.

Ramanathan S. (1996), "An algorithm for multicast tree generation in networks with asymmetric links", Proceedings of IEEE INFOCOM, pp. 337-344.

Randaccio L.S. and Atzori L. (2007), "Group multicast routing problem: A genetic algorithms based approach", Computer Networks, vol. 51, no. 14, pp. 3989–4004, 2007.

Rauch H.E. and Winarske T. (1988), "Neural network implementation of the shortest path algorithm for routing communication traffic", IEEE Con. Sys. Mag., pp. 26-30.

Ravikumar C. P. and Bajpai R. (1998), "Source-based delay-bounded multicasting in multimedia networks", Computer Communications, vol. 21, no. 2, pp.126-132.

Ray T. and Liew K. M. (2002), "A Swarm Metaphor for Multiobjective Design Optimization", Engineering Optimization, vol.34, no. 2, pp.:141–153.

Ribeiro C.C.C., Martins S.L. and Rosseti I.C.M., (2007), "Metaheuristics for optimisation problems in computer communications", Computer Communications, vol. 30, no. 4, pp. 656-669.

Rouskas and Baldine I. (1996), "Multicast routing with end-to-end delay and delay variation constraints", Proceedings of IEEE INFOCOM, vol.1, pp. 353-360.

Roy A. and Das S.K. (2004), "QM2RP: A QoS-based mobile multicast routing protocol using multi-objective genetic algorithm", Wireless networks, vol. 10, no. 3, pp. 271–286.

Salama H.F., Reeves D.S. and Viniotis Y. (1997), "Evaluation of multicast routing algorithms for real time communication on high speed networks", IEEE Journal on Selected Areas in Communications, vol. 15, no. 3, pp. 332-345.

Schaffer J. D. (1985), "Multiple Objective Optimization with Vector Evaluated Genetic Algorithms", 1st International Conference on Genetic Algorithms, pp. 93–100.

Shen J., Xu F. and Zheng P. (2005), "A tabu search algorithm for the routing and capacity assignment problem in computer networks", Computers and Operations Research, vol. 32, no. 11, pp. 2785-2800.

Shen Y.J. and Wang M.S. (2008). Broadcast scheduling in wireless sensor networks using fuzzy Hopfield neural network. Expert Systems with Applications, vol. 34, no. 2, pp. 900–907.

Smith V.R. (1983), "The computation of nearly minimal Steiner trees in graphs", International Journal of Mathematical Education in Science and Technology, vol. 14, no. 1, pp. 15-23.

Srinivas N. and Deb K. (1994), "Multiobjective Optimization Using Nondominated Sorting in Genetic Algorithms", Evolutionary Computation, vol. 2, no. 3, pp.:221–248.

Sriram R., Manimaran G. and Murthy C.S.R. (1998), "Algorithms for delay-constrained low-cost multicast tree construction", Computer Communications, vol. 21, no. 18, pp. 1693-1706.

Sriram R., Manimaran G. and Murthy C.S.R. (1998), "Preferred link based delay-constrained least-cost routing in wide area networks", Computer Communications, vol. 21, no. 18, pp. 1655-1669.

Stadler W. (1979), "A Survey of Multicriteria Optimization or the Vector Maximum Problem", .JOTA, vol. 29, no. 4, pp.1–52.

Stalling W. (1998), "High-Speed Networks: TCP/IP and ATM Design", Principles. Englewood Cliffs, NJ: Prentice-Hall.

Striegel A. and Manimaran G. (2002), "A survey of QoS multicasting issues", IEEE Communications Magazine, vol. 40, no. 6, pp. 82-87.

Sun J., Fang W., Wu X., Xie Z. and Xu W. (2011), "QoS multicast routing using a quantum-behaved particle swarm optimization algorithm", Engineering Applications of Artificial Intelligence, vol. 24, no. 1, pp. 123–131.

Takahashi H. and Matsuyama A. (1980), "An approximate solution for the Steiner problem in graphs", Mathematica Japonica, vol. 24, no. 6, pp. 573-577.

Tanaka Y. and Huang P. (1993), "Multiple destination routing algorithms", IEICE Transactions on Communications, vol. 76, no. 5, pp. 544-552, 1993.

Tsai S.J, Sun T.Y., Liu C.C., Wu W.C. and Chiu S.Y. (2010), "An improved multi-objective particle swarm optimizer for multi-objective problems", Expert Systems with Applications, vol. 37, pp. 5872–5886.

Tsenga S.Y., Linb C.C., Huangc Y.M. (2008), "Ant colony-based algorithm for constructing broadcasting tree with degree and delay constraints", Expert Systems with Applications, vol. 35, pp. 1473–1481.

Van Veldhuizen D. A. and Lamont G. B. (2000), "Multiobjective Evolutionary Algorithms: Analyzing the State-of-the-Art", Evolutionary Computation, vol.8, no. 2, pp.:125–147.

Venkataram P., Ghosal S. and Vijaykumar B.P. (2002), "Neural network based optimal routing algorithm for communication networks", Neural Networks, vol. 15, no. 10, pp. 1289-1298.

Venkatraman S. and Yen G.G. (2005), "A Generic Framework for Constrained Optimization Using Genetic Algorithms", IEEE Transactions on Evolutionary Computation, vol. 9, no. 4, pp. 424- 435.

Vijayalakshmia K. and Radhakrishnanb S. (2008), "Artificial immune based hybrid GA for QoS based multicast routing in large scale networks (AISMR)", Computer Communications, vol. 31, no. 17, pp. 3984–3994.

Vijaykumar B.P. and Venkataram P. (2003), "Reliable multicast routing in mobile networks : a neural network approach", IEE Proceedings on Communications, vol. 150, no. 5, pp. 377-384.

Waitzman D., Partridge C. and Deering S. (1988), "Distance vector multicast routing protocol RFC 1075"

Wang B. and Hou J.C. (2000), "Multicast routing and its QoS extension: problems, algorithms, and protocols", IEEE Network, vol. 14, no. 1, pp. 22-36.

Wang C.F., Lai B.R. and Jan R.H. (1999), "Optimum multicast of multimedia streams", Computers and Operations Research, vol. 26, no. 5, pp. 461-480.

Wang H., Fang J., Wang H. and Sun Y.M. (2004), "TSDLMRA: an efficient multicast routing algorithm based on Tabu search", Computer Applications, vol. 27, no. 2, pp. 77–90.

Wang H., Meng X., Li S. and Xu H. (2010), "A tree-based particle swarm optimization for multicast routing", Computer Networks, vol. 54, no. 15, pp. 2775–2786.

Wang L., Liu W. and Xi H. (2009), "Delay constrained multicast routing using noisy chaotic neural network", IEEE Transactions on Computers, vol. 53, no. 1, pp. 82-89.

Wanga X., Caob J., Chengb H., and Huanga M. (2006), "QoS multicast routing for multimedia group communications using intelligent computational methods", Computer Communications, vol. 29, no. 12, pp. 2217–2229.

Waxman B. (1988), "Routing of multipoint connections", IEEE Journal on Selected Areas in Communications, vol. 6, no. 9, pp. 1617-1622.

Woldesenbet Y.G., Yen G.G. and Tessema B.G. (2009), "Constraint Handling in Multiobjective Evolutionary Optimization", IEEE Transactions on Evolutionary Computation, vol. 13, no. 3, pp. 514-525.

Xiao X. and Ni L.M., (1999), "Internet QoS: a big picture", IEEE Network, vol.13, pp. 8-18.

Xu Y. and Qu R. (2011), "Solving multi-objective multicast routing problems by evolutionary multi-objective simulated annealing algorithms with variable neighbourhoods", Journal of the Operational Research Society, vol. 62, no.2, pp. 313-325.

Yen Y.S., Chan Y.K., Chao H.C. and Park J.H. (2008), "A genetic algorithm for energy-efficient based multicast routing on MANETs", Computer Communications, vol. 31, no. 4, pp.2632–2641.

Youssef H., Al-Mulhem A., Sait S.M. and Tahir M.A. (2002), "QoS-driven multicast tree generation using Tabu search", Computer Communications, vol. 25, no. 11-12, pp. 1140-1149.

Youssef H., Sait S.M. and Adiche H. (2001), "Evolutionary algorithms, simulated annealing and tabu search: a comparative study", Engineering Applications of Artificial Intelligence, vol. 14, no. 2, pp. 167-181.

Zahrani M.S., Loomes M.J., Malcolm J.A., Ullah A.Z.M., Steinhöfel K. and Albrecht A.A. (2008), "Genetic local search for multicast routing with pre-processing by logarithmic simulated annealing", Computers and Operations Research, vol. 35, no. 6, pp. 2049-2070.

Zhang K., Wang H. and Liu F.Y. (2005), "Distributed multicast routing for delay variation-bounded Steiner tree using simulated annealing", Computer Communications, vol. 28, pp. 1356-1370.

Zhang L., Cai L., Li M. and Wang F. (2009), "A method for least-cost QoS multicast routing based on genetic simulated annealing algorithm", Computer Communications, vol. 32, no. 1, pp.105–110.

Zhang Q. and Leung Y.W. (1999), "An orthogonal genetic algorithm for multimedia multicast routing", IEEE Transactions on Evolutionary Computation, vol. 3, no. 1, pp. .

Zhengying W., Bingxin S. and Erdun Z. (2001), "Bandwidth delay-constrained least cost multicast routing based on heuristic genetic algorithm", Computer Communications, vol. 24, no. 7, pp.685-692.

Zhou A., Qu B.Y., Li. H., Zhao S.Z., Suganthan P.N. and Zhang Q. (2011), "Multiobjective evolutionary algorithms : A survey of the state of art", Swarm and Evolutionary Computation, vol. 1, no. 1, pp. 32-49.

Zitzler E. and Thiele L. (1999), "Multiobjective evolutionary algorithms: A comparative case study and the strength Pareto approach", IEEE Transactions on Evolutionary Computation, vol. 3, no. 4, pp. 257–271.

NETWORK DATA

(a) 24-Node USIP Backbone Network

The 24-Node USIP Backbone Network is shown in Fig. A1 and the link data is given in Table A1. The link cost *C(e)* is the monetary cost of the communication, link delay *D(e)* is time delay in *ms*, available bandwidth $\alpha(e)$ is specified in *Mbps*. The link capacity $\phi(e)$ is uniform for all the links and is taken as 200 *Mbps*. The link cost is selected randomly between 2-10, delay between 1-5 ms, Available bandwidth between 70-170 *Mbps.* The packet loss probability is selected randomly between 0.01 and 0.1.

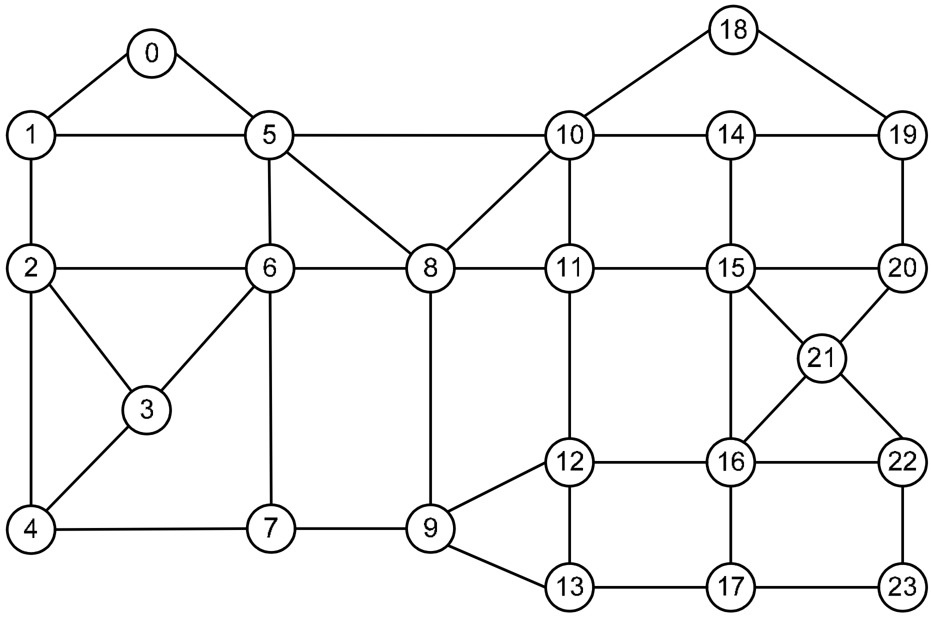


Fig. A1 Representation of 24-Node USIP Backbone Network

Table A1 Link data for 24-Node USIP Backbone Network

Link e(u,v)	Node (u)	Node (v)	Cost C(e)	Delay D(e) (ms)	Available Bandwidth α (e) (Mbps)	Packet loss probability pl(e)
0	0	1	5.16	4.36	148.31	0.08
1	0	5	3.58	4.65	103.52	0.08
2	1	2	6.43	2.11	117.74	0.06
3	1	5	6.11	2.46	165.22	0.09
4	2	3	7.74	3.54	84.16	0.06
5	2	4	3.94	1.07	83.72	0.08
6	2	6	5.21	1.63	82.98	0.01
7	3	4	3.75	5.00	121.29	0.08
8	3	6	4.37	3.45	133.76	0.05
9	4	7	9.78	2.97	99.25	0.08
10	5	6	8.16	3.11	110.02	0.09
11	5	8	4.82	2.13	150.77	0.09
12	5	10	9.59	1.28	122.60	0.01
13	6	7	7.31	1.77	159.02	0.03
14	6	8	2.16	1.26	115.77	0.01
15	7	9	9.77	1.95	160.22	0.09
16	8	9	6.32	2.07	107.52	0.08
17	8	10	7.34	3.05	123.16	0.00
18	8	11	9.45	2.75	163.08	0.07
19	9	12	7.91	2.14	134.00	0.04
20	9	13	3.33	3.75	114.01	0.09
21	10	11	4.64	4.32	92.90	0.09
22	10	14	7.49	2.40	165.65	0.06
23	10	18	8.87	3.63	113.96	0.09
24	11	12	8.52	2.59	138.42	0.09
25	11	15	3.73	2.93	165.03	0.09
26	12	13	9.05	1.59	134.11	0.04
27	12	16	4.25	3.48	148.60	0.03
28	13	17	3.81	2.79	88.75	0.03
29	14	15	5.33	3.23	86.96	0.09
30	14	19	3.01	1.41	119.54	0.08
31	15	16	9.48	4.94	138.44	0.04
32	15	20	4.95	4.00	99.42	0.02
33	15	21	3.96	3.34	85.24	0.07
34	16	17	8.35	1.50	86.41	0.07
35	16	21	9.60	1.30	75.25	0.05
36	16	22	3.92	1.70	149.78	0.07
37	17	23	9.74	3.63	133.95	0.08
38	18	19	3.08	1.37	122.02	0.01

39	19	20	3.64	1.28	116.14	0.08
40	20	21	8.04	3.29	75.19	0.02
41	21	22	3.63	5.00	159.00	0.01
42	22	23	2.43	4.99	157.05	0.01

(b) 100-Node Random Network

A random network of 100 nodes is generated by the BRITE network topology generator using the Waxman model, detailed in Appendix-B. The network spread is assumed in the area of $500\,Km \times 500\,Km$. The parameters that are controlling the edge density β and the density of short edges with respect to longer ones α are specified as 0.9 and 0.7 respectively. The connection of each new node has been considered with two other nodes. The link cost *C(e)* is the monetary cost of the communication, link delay *D(e)* is time delay in *ms*, available bandwidth $\alpha(e)$ is specified in *Mbps*. The link capacity $\phi(e)$ is uniform for all the links and is taken as 200 *Mbps*. For 100-Node random network, the link cost is selected randomly between 2-10, delay between 1-5 ms, Available bandwidth between 70-170 *Mbps.* The packet loss probability is selected randomly between 0.01 and 0.1. The link data for 100-Node random network is specified in Table A2.

Table A2 Link data for 100-Node Random Network

Link e(u,v)	*Node (u)*	*Node (v)*	*Cost C(e)*	*Delay D(e) (ms)*	*Available Bandwidth* α *(e) (Mbps)*	*Packet loss probability pl(e)*
0	31	30	5.16	4.36	148.31	0.08
1	31	62	3.58	4.65	103.52	0.08
2	95	31	6.43	2.11	117.74	0.07
3	95	30	6.11	2.46	165.22	0.09
4	86	95	7.74	3.54	84.16	0.06
5	86	31	3.94	1.07	83.72	0.08
6	59	95	5.21	1.63	82.98	0.02
7	59	30	3.75	5	121.29	0.09
8	32	31	4.37	3.45	133.76	0.06
9	32	95	9.78	2.97	99.25	0.08
10	34	32	8.16	3.11	110.02	0.09
11	34	31	4.82	2.13	150.77	0.09
12	96	34	9.59	1.28	122.6	0.02
13	96	62	7.31	1.77	159.02	0.04
14	60	86	2.16	1.26	115.77	0.02

15	60	32	9.77	1.95	160.22	0.09
16	85	96	6.32	2.07	107.52	0.08
17	85	34	7.34	3.05	123.16	0.01
18	1	31	9.45	2.75	163.08	0.07
19	1	34	7.91	2.14	134	0.04
20	61	86	3.33	3.75	114.01	0.09
21	61	62	4.64	4.32	92.9	0.09
22	29	62	7.49	2.4	165.65	0.06
23	29	86	8.87	3.63	113.96	0.09
24	68	31	8.52	2.59	138.42	0.09
25	68	59	3.73	2.93	165.03	0.09
26	58	1	9.05	1.59	134.11	0.05
27	58	85	4.25	3.48	148.6	0.04
28	82	86	3.81	2.79	88.75	0.03
29	82	32	5.33	3.23	86.96	0.09
30	63	31	3.01	1.41	119.54	0.08
31	63	1	9.48	4.94	138.44	0.04
32	3	82	4.95	4	99.42	0.03
33	3	68	3.96	3.34	85.24	0.08
34	81	61	8.35	1.5	86.41	0.08
35	81	96	9.6	1.3	75.25	0.06
36	27	82	3.92	1.7	149.78	0.08
37	27	85	9.74	3.63	133.95	0.08
38	64	31	3.08	1.37	122.02	0.02
39	64	34	3.64	1.28	116.14	0.08
40	2	82	8.04	3.29	75.19	0.02
41	2	27	3.63	5	159	0.02
42	28	68	2.43	4.99	157.05	0.02
43	28	58	9.38	1.02	129.39	0.03
44	38	58	5.13	1.65	161.3	0.08
45	38	85	6.42	2.44	127.94	0.05
46	57	96	2.8	3.75	123.08	0.08
47	57	29	9.94	2.22	127.7	0.09
48	94	82	7.03	3.99	73.54	0.08
49	94	64	9.4	4.33	157.33	0.08
50	23	94	7.95	4.92	160.34	0.1
51	23	64	5.98	3.67	86.4	0.08
52	87	95	2.62	4.56	134.97	0.03
53	87	61	3.83	3.52	140.06	0.04
54	35	95	3.85	2.32	77.42	0.07
55	35	32	7.21	1.89	121.07	0.1
56	7	34	6.37	2.12	141.93	0.02
57	7	1	6.74	2.89	164.43	0.05
58	54	32	8.78	2.35	113.45	0.01

59	54	27	6.79	2.38	153.32	0.03
60	53	58	5.86	3.7	118.19	0.04
61	53	87	3.46	3.85	132.18	0.01
62	65	86	7.57	2.66	137.39	0.07
63	65	29	3.48	2.39	130.91	0.07
64	33	62	4.63	3.92	144.04	0.03
65	33	86	7.48	4.68	135.31	0.03
66	6	62	2.7	3.13	96.05	0.09
67	6	27	2.75	3.74	81.13	0.04
68	93	59	6.75	3.31	136.66	0.04
69	93	63	4.31	4.1	102.96	0.03
70	24	7	2.03	4.94	152.74	0.04
71	24	1	5.49	1.75	165.86	0.09
72	88	58	7.59	4.06	82.11	0.07
73	88	57	8.19	2.54	164.31	0.09
74	4	60	3.63	4.45	149.37	0.06
75	4	34	9.24	2.19	160.96	0.09
76	36	24	6.61	2.99	86.28	0.03
77	36	31	5.94	4.46	116.37	0.09
78	26	88	4.33	2.98	88.04	0.07
79	26	3	3.11	3.91	130.31	0.05
80	55	4	7.79	4.35	87.82	0.03
81	55	95	2.97	2.99	83.82	0.04
82	67	59	9.46	2.3	160.85	0.07
83	67	31	8.55	4.35	119.61	0.04
84	76	27	7.27	2.58	130.89	0.03
85	76	26	2.58	1.6	80.78	0.07
86	45	93	4.31	2.45	103.14	0.02
87	45	7	9.48	2.71	128.36	0.03
88	74	55	8.09	3.63	118.74	0.02
89	74	53	7.01	4.53	121.77	0.03
90	46	96	5.41	3.23	152.99	0.05
91	46	65	4.61	1.98	142.94	0.07
92	25	3	4.71	4.94	159.76	0.02
93	25	46	2.04	2.64	148.33	0.08
94	56	34	2.92	2.17	156.55	0.07
95	56	67	5.59	1.2	168.65	0.07
96	5	54	5.79	1.84	156.52	0.02
97	5	46	5.06	1.4	100.18	0.07
98	15	27	3.05	4.24	75.15	0.01
99	15	82	8.25	2.83	139.21	0.05
100	79	62	6.72	1.48	127.86	0.06
101	79	4	4.9	3.38	100.43	0.09
102	43	28	3.36	2.91	130.97	0.06

103	43	5	6.77	3.48	93.37	0.08
104	16	1	2.79	1.28	162.37	0.03
105	16	4	3.8	2.93	152.68	0.04
106	48	7	9.03	2.43	104.43	0.08
107	48	27	2.29	3.64	95.75	0.08
108	80	2	8.69	3.5	100.82	0.03
109	80	16	6.9	1.79	80.97	0.07
110	73	3	7.76	4.13	90.04	0.05
111	73	27	5.47	2.26	93.1	0.04
112	41	38	3.24	3.13	125.54	0.01
113	41	95	5.06	2.52	100.54	0.08
114	97	79	7.2	2.04	125.23	0.09
115	97	38	8.48	3.74	139.78	0.04
116	70	6	2.05	3.58	123.3	0.09
117	70	23	7.14	3.47	121.85	0.05
118	14	70	7.75	2.45	150.19	0.07
119	14	53	2.26	1.61	76.36	0.07
120	83	97	6.95	1.75	140.03	0.06
121	83	76	2.05	1	100.52	0.03
122	47	25	8.86	3.62	88.12	0.04
123	47	30	9.03	3.67	135.33	0.04
124	75	82	3.49	4.54	85.71	0.06
125	75	2	7.41	4.32	160.42	0.03
126	18	32	7.65	2.58	156.89	0.06
127	18	4	9.46	3.96	93.31	0.09
128	66	34	9.47	3.21	119.44	0.06
129	66	95	8.4	4.76	151.41	0.06
130	98	56	9.96	3.63	163.59	0.04
131	98	7	6.71	4.5	133.78	0.08
132	12	97	8.36	4.1	96.28	0.06
133	12	3	3.34	2.88	149.55	0.09
134	44	6	7.32	4.49	111.25	0.07
135	44	68	7.16	3.39	123.86	0.02
136	17	28	2.26	3.32	140.09	0.06
137	17	82	6.12	4.33	81.26	0.05
138	13	48	2.39	3.04	151.44	0.04
139	13	33	5.62	3.55	84.4	0.05
140	77	95	5.25	1.99	71.75	0.07
141	77	44	8.5	3.29	128.27	0.05
142	84	15	9.96	2.91	75.87	0.02
143	84	80	6.78	3.56	92.26	0.03
144	37	27	9.39	3.52	143.79	0.05
145	37	68	8.8	2.75	165.27	0.1
146	51	98	8.14	4.6	103.36	0.06

147	51	45	5.82	1.88	164.98	0.05
148	8	95	9.74	4.54	88.38	0.05
149	8	23	8.13	4.12	160.48	0.03
150	40	6	9.71	4.05	103.18	0.05
151	40	83	6.44	3.24	132.22	0.03
152	99	73	4.88	2.91	135.39	0.09
153	99	96	6.85	1.84	156.54	0.02
154	72	5	3.59	2.49	134.65	0.06
155	72	57	6.77	3.71	75.89	0.06
156	49	40	3.94	3.25	71.89	0.04
157	49	82	9.39	1.04	130.14	0.08
158	71	2	9.47	4.55	87.31	0.05
159	71	85	8.36	2.95	133.9	0.1
160	22	36	4.34	1.62	158.22	0.04
161	22	41	7.98	4.6	117.58	0.03
162	9	84	2.98	4.79	156.57	0.07
163	9	65	9.4	3.87	88.41	0.04
164	21	35	3.62	1.67	132.61	0.03
165	21	66	3.82	1.51	164.69	0.01
166	78	47	2.96	1.64	116.18	0.07
167	78	14	2.81	4.66	131.42	0.02
168	52	23	5.97	2.57	113.66	0.04
169	52	2	9.3	1.98	126.62	0.03
170	42	84	5.45	1.14	151.39	0.08
171	42	6	9.98	2.43	73.57	0.06
172	90	55	7.29	1.8	139.98	0.04
173	90	88	7.17	4.56	104.15	0.01
174	19	2	8.43	4.07	139.87	0.07
175	19	5	4.5	4.62	145.25	0.04
176	11	14	3.51	4.24	129.11	0.01
177	11	81	3.26	1.41	94.41	0.02
178	50	21	2.46	3.36	158.96	0.1
179	50	27	9.4	1.22	116.9	0.03
180	91	60	3.35	3.35	128.46	0.05
181	91	11	9.41	4.26	122.65	0.06
182	39	18	3.8	3.92	96.42	0.07
183	39	23	2.13	3.15	163.15	0.04
184	92	29	6.18	1.82	110.1	0.04
185	92	79	7.16	3.72	114.33	0.03
186	89	94	4.66	3.81	91.45	0.08
187	89	17	7.47	2.03	71.62	0.09
188	20	61	6.81	4.41	102.15	0.07
189	20	78	8.78	3.11	95.02	0.03
190	10	83	6.12	1.29	158.98	0.07

191	10	66	8.57	3.12	165.9	0.08
192	0	89	4.88	2.38	74.39	0.01
193	0	25	5.9	1.02	99.29	0.07
194	69	2	6.06	4.28	116.75	0.02
195	69	81	5.87	1.76	162.34	0.01
196	62	94	3.96	1.34	141.14	0.07
197	62	91	9.69	1.37	156.75	0.02
198	30	2	8.06	2.9	147.75	0.01
199	30	28	7.89	3.31	144.37	0.09

RANDOM NETWORK GENERATOR

A random network of 100 nodes is generated by the BRITE network topology generator using the Waxman model (Waxman, 1988). Waxman model used the following probability function to create links interconnecting to the nodes

$$P_e(u,v) = \beta e^{\frac{-l(u,v)}{L\alpha}} \quad \text{(b.1)}$$

Where, L is the maximum distance between any two nodes in the network and $l(u,v)$ is the Euclidean distance between u and v. The parameter α controls the ratio of short links to a long links, while the parameter β controls the average node degree of the network. A large value of α increases the number of long links, and a large value of β results in a large average node degree.

The network spread is assumed in the area of $500\,Km \times 500\,Km$. The connection of each new node has been considered with two other nodes. The values of α and β are taken as 0.15 and 0.2.