ISSN 2047-3338

Multi-objective Design of Communication Networks Using Memetic Algorithms

R. O. Oladele and H. A. Sanusi
Department of Computer Science, University of Ilorin, P.M.B. 1515, Ilorin, Nigeria
roladele@yahoo.com, ahmedsanusi@gmail.com

Abstract– **Evolutionary Algorithms (EAs) represent an elegant class of solution paradigms that can efficiently tackle NP-hard problems such as network design problems. The most widely used of these EAs is genetic algorithm (GA). However, GA is prone to premature convergence making it unable to search numerous solutions of the problem domain. A memetic algorithm (MA) which is a symbiosis of GA and local search technique is an effective option for reducing the likelihood of premature convergence. This paper proposes a MA-based approach for multi-objective design of communication networks. To be able to estimate the quality and cost (in computation time) of obtained MA solutions, we design a GA and use it to equally solve the problem. Our computational experiments show that MA is superior to GA in solution quality but inferior to GA in computational time. However, for 36-node network (large network), MA is able to find solutions (though not best) that are better both in quality and cost than the best GA solution.**

Index Terms– **Memetic Algorithm, Network Design, NP-Hard Problems, Local Search and Genetic Algorithm**

I. INTRODUCTION

A network can be described as a finite collection of nodes (switches) and links whereby all nodes are connected by links. Typically, a large scale network has hierarchical structure consisting of a backbone network and many local access networks (LANs). The backbone network is a distributed system dedicated for routing data from source to destination using switching nodes. The LAN is usually a centralized system which grants users access to hosts and local servers. The backbone network design is the focus of this paper.

Topological design of communication networks such as mesh/wide area networks is a typical multi-objective problem involving simultaneous optimization of cost of the network and various performance criteria such as average delay of the network, throughput and reliability [1]. Optimizing one or more of such criteria to make the network efficient and cost-effective is often the main objective of design. The two criteria that are often considered are costs and average packet delay. The problem can be stated as: given a set of nodes, design the links layout among the nodes while optimizing certain factors such as overall cost, average packet delay, reliability and provision for expansion. This implies optimizing conflicting factors, subject to several constraints. For instance, reducing the links' cost could mean reducing the links' capacities, which will ultimately lead to increase in packet delay. Searching the entire solution space for a design problem of this nature is an NP hard problem. Real-life applications will therefore benefit from efficient optimization of these conflicting factors.

Since this problem is a well-known NP-hard problem, enumerative-based and heuristic techniques have been widely used for such design problem. Single objective versions of the problem are well studied in the literature. Nevertheless, a multiobjective version of the problem has received very little attention from researchers. Some work on multiobjective network design problem include the following: Kumar et al [1] applied Pareto Converging Genetic Algorithm (PCGA) and discovered that the convergence properties of PCGA are better than those of branch exchange heuristics. In addition PCGA was found to be scalable to large networks. Banerjee and Kumar [2] studied multiobjective network design using an EA heuristic and empirically showed that the EA heuristic generally provides better solution than its deterministic counterparts. Papagianni et al [3] used particle swarm optimization (PSO) to solve multi criteria network design problem. It is claimed that the approach is more effective than the GA used in [2] but the algorithm was only tested for 16-node network and nothing was said about its efficiency. Duarte and Bar´an [4] proposed a parallel EA for solving network design problems with cost and reliability as objectives. The proposed EA was found to be capable of obtaining a broader set of solutions than the sequential variant in addition to its better efficiency. Among these techniques, GA is the most successful method that has aroused the interest of many researchers.

Nevertheless, little or no attention has been given to MA (hybrid GA) regarding its suitability for network design problem. This paper therefore proposes a MA for multi-objective design of network topology with a view to investigating and estimating his performance relative to the widely used GA, vis-à-vis solution quality and computation time.

The remainder of this paper is organized as follows: in section 2, the main contributions of the paper are presented. Mathematical formulation of the problem is given in section 3. Section 4 contains a description of the solution methodology – MA and GA. Section 5 presents results of numerical experiments using both MA and GA. Discussion of results is presented in section 6 while the paper is finally concluded in section 7.

II. CONTRIBUTIONS

Our contributions in this paper are summarized as follows:

i). We propose a MA for multi-objective design of communication networks.

ii). We also present a GA-based variant of the proposed algorithm

iii). We investigate the performance of the proposed MA in relation to its GA-based variant.

III. MATHEMATICAL FORMULATION

A multi-objective minimization problem subject to constraints can be stated as:

Minimize $f_m(X)$, m = 1,2,…….,M (1)

Subject to $g_k(x) \le C_k$, k = 1,2,……..,K (2)

Where,

X = $(X_1, X_2, \ldots\ldots\ldots\ldots, X_N)$ is an N-tuple vector of variables

F = $(f_1, f_2, \ldots\ldots\ldots\ldots, f_M)$ is an M-tuple vector of objectives.

The following mathematical model is used for the problem:

A. Nomenclature

Node: It is the source or sink of traffic in the network. It could be a machine or individual network which can exchange data with other nodes.

Link: Links are network devices along which data are transferred in the network. They are assumed to be bi-directional and completely reliable. For example, they could be made of any physical media, such as fiber optic cables. (This excludes communication via satellite links). Links have a cost per unit distance.

Network equipment (NE): Network equipment is the generic term used to refer to the class of devices present in the nodes that are used for processing data packets. The processing speed of each node is a function of the class of device present in it. Examples of these devices are FDDI adapters and network cards.

B. Design Parameters

The following given network parameters are used in the design:

N, the total number of nodes in the network

(i,j), a link between nodes i and j

Dij, length of the link between nodes i and j in km

K, the number of active (selected) links in the network

Pij, selection status of (i, j): Pij = 1 if (i, j) is selected, else Pij = 0

C. Objective Functions

Two objective functions; network cost and end-to-end message delivery delay, each approximated by the following formulations:

1) Network Cost

$$NetCost = NodeCost + LinkCost + AmpCost \quad (3)$$

Where

$$NodeCost = \sum_i C_i \quad (4)$$

$$LinkCost = \sum_i \sum_j C_{ij} \quad (5)$$

$$Amp = Cost \frac{\sum_i \sum_j D_{ij} \; X \; A}{L} \quad (6)$$

Where:

C_i = cost of the network equipment at node i

C_{ij} = cost of the link between node i and node j

L = maximum distance for which the signal is sustained without amplification

A = cost of each amplifier unit

2) Average Delay

$$AvDelay = \frac{\sum_i \sum_j [DELAY_{ij} \; X \; LFLOW_{ij}]}{\sum_i \sum_j LFLOW_{ij}} \quad (7)$$

Where:

$LFLOW_{ij}$ = traffic flowing along link (i,j)

From queuing theory,

$$DELAY_{ij} = \frac{1}{[CAP_{ij} - LFLOW_{ij}]} \quad (8)$$

$DELAY_{ij}$ = link delay for packets flowing along link (i,j)

CAP_{ij} = capacity of link (i,j)

D. Constraints

$Flow\ Constraints$: $LFLOW_{ij} \le CAP_{ij}$ (9)

Reliability Constraint: R(x) $\ge R_0$ (10)

Where:

x = architecture of network

R(x) = reliability of network

$R_0 =$ minimum network reliability

E. Routing Policy

Breadth-first Search is used for routing. The metric used for this purpose is the length of the link.

F. Reliability Estimation

Monte Carlo Simulation is used to estimate network reliability. The network is simulated t times, given the design and the links' reliabilities. The method is given below.

intialize i = 0, c= 0

Step C0: while i < t Repeat.

Step C1: Randomly generate network

(a) : i = i + 1.

Step C2 : Check to see if the network forms a spanning tree

(a) : if YES, increment c by 1 and go to Step C0

(b) : if NO, go to Step C0

Step C3 : R(x) = c / t.

G. Assumptions

The location of each network node is given:
Each Cij is fixed and known
Each link is bidirectional i.e. a path can be traversed in either direction
There is no redundant link in the network

IV. SOLUTION METHODOLOGY

A. Memetic Algorithm Approach

1 Initialization: randomly generate population of N chromosomes
2 Fitness: calculate the fitness of all chromosomes
3 Create a new population:
a. Selection: select 2 chromosomes from the population
b. Crossover: produce 2 off springs from the 2 selected chromosomes
c. Local Search: apply local search to each offspring
d. Mutation: perform mutation on each offspring.
e. Local search: apply local search to each offspring.
4 Replace: replace the current population with the new population
5 Termination: Test if the termination condition is satisfied. If so stop. If not, return the best solution in the current population and go to step 2.

B. Genetic Algorithm (GA) Variant

This is just MA (shown above) without local search.

C. Implementation Details

1) Encoding Scheme

The chosen encoding scheme is such that every chromosome codes a possible network, which corresponds to an individual in a set of feasible solutions of the problem. This set of feasible solutions constitutes a population. The chromosome is represented by a constant length integer string representation. The chromosome consists of two parts, the first part contains details of NE's at the nodes and the second part consists of details of the links. For example, if there are H types of nodes, then $\log_2 H$ bits are required to encode a node. Therefore the first part of the chromosome consists of $N .\log_2 H$ bits. If a link exists between nodes 1 and 2 then the first bit position in the link part is set to 1. Hence the second part of the chromosome consists of (N(N-1))/2 bits.

2) Initial Population

The two algorithms start by creating an initial population. There are two ways of generating initial population namely heuristic process and random process. A random process of generating initial populations is adopted. The random initialization procedure does not guarantee the feasibility of each solution in the initial population. As such a *checking* process is involved. A checking process checks if each solution in the population is feasible (i.e. satisfies the constraints). For all infeasible solutions an *update* function is used to replace the infeasible solutions with new feasible solutions.

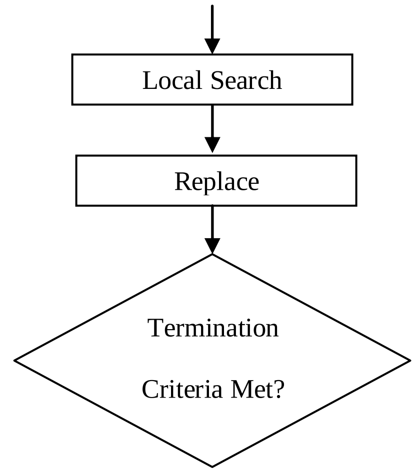

Fig. 1: Flow Diagram

3) Fitness Evaluation

The fitness of the individual chromosome is a function of the NodeCost, NetCost as well the number of links in the network. Mathematically, the fitness F of each solution can be expressed as follows:

$$F(x) \; = \sum_i C_i \,/\left(\sum_i C_i + \sum_i \sum_j C_{ij} + \frac{\sum_i \sum_j D_{ij} \; X \; A}{L}\right) X \; K \qquad (11)$$

4) Selection

Two individuals are selected by Roulette wheel selection [15] in which the probability of an individual i being selected is proportional to $\text{Fitness}_i / (\sum \text{Fitness}_i)$. A higher probability is given for selection of fitter individuals. This is to ensure that fitter individuals stand a better chance of being parents.

5) Crossover

This operation operates on two chromosomes. The chromosomes are randomly selected based on the probability of crossover which is a randomly generated number ranging between 0 and 10. In this work, the two point crossover technique was implemented. The *crossover probability* (denoted by *p*C) is the probability of the number of offspring produced in each generation to the population size (denoted by *popSize*). This probability controls the expected number *p*C × *popSize* of chromosomes to undergo the crossover operation. A high crossover probability is used here to allow exploration of more of the solution space, and reduces the chances of settling for a false optimum; but if this probability is too high, it results in the wastage of a lot of computation time in exploring unpromising regions of the solution space.

6) Mutation

This is the operation of randomly changing some of the bits of the chromosome representing an individual with a view to increasing the exploration of the solution space.

7) Local Search

The local search technique used in MA is the hill climbing search algorithm. It is essentially an iteration that continuously proceeds in the direction of increasing quality value. The algorithm is shown below:

```
While (termination condition is not satisfied) do
        New solution ← neighbours(best solution);
        If new solution is better than actual solution then
                Best solution ← actual solution
        End if
End while
```

V. NUMERICAL EXPERIMENTS

In this section, results of numerical experiments using 3 test problems are reported. All experiments were performed on a HP 630 NOTEBOOK PC with the following configuration: 2.13GHz Processor Speed, 3.0GB RAM and 64 BIT OS
The two algorithms were implemented in Java. For each algorithm 2 runs were made for each test problem. Table 1, Table 2 and Table 3 report the simulation run that produced better optimum values with the best solution in bold. The algorithms were run with the following parameters:
Population size - 100 (250 for test problem 3)
Mutation probability – 0.02
Two- point crossover was used
Number of Node Type - 4

MA						GA					
No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time	No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time
5	98	619.6	0.97	0.06	156	5	100	534.0	0.98	0.07	134
10	**100**	**534.0**	**0.99**	**0.05**	**432**	**10**	**100**	**534.0**	**0.98**	**0.06**	**344**
15	100	534.0	0.99	0.05	557	15	100	534.0	0.98	0.06	450
20	100	534.0	0.99	0.05	605	20	100	534.0	0.98	0.06	570

Table 1: Results for 10-node network problem

For GA at 10th generation the solution (i.e chromosome, network) generated from the experiment is shown below:
10100000010001001110||0000100100110100001101110000101111001000000001
For MA at 10th generation the solution (i.e chromosome, network) generated from the experiment is shown below:
00010010110000100010 || 000000001100100010110100010110101011111111100

MA						GA					
No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time	No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time
5	97	1259	0.97	0.06	954	5	98	1305	0.993	0.07	837
10	100	1167.4	0.97	0.05	1892	10	100	1167.4	0.998	0.06	1482
15	100	1167.4	0.98	0.04	4467	**15**	**100**	**1265.8**	**0.998**	**0.05**	**4143**
20	**100**	**1158.6**	**1.0**	**0.04**	**5665**	20	100	1265.8	0.998	0.06	5434

Table 2: Results for 21-node network problem

MA						GA					
No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time	No of Gen	C/Ratio	Cost	Reliability	Delay	Comp. Time
5	97	1167.4	0.97	0.06	3598	5	98	1305	0.993	0.05	2333
10	*100*	*1167.4*	*0.97*	*0.05*	5890	10	100	1259	0.998	0.07	3771
15	*100*	*1167.4*	*0.98*	*0.05*	9673	15	100	1259	0.92	0.08	7043
20	100	1167.4	0.985	0.05	15897	**20**	**100**	**1167.4**	**0.95**	**0.06**	**12980**
25	**100**	**1167.4**	**1**	**0.04**	**18754**	25	100	1259	0.95	0.09	13090

Table 3: Results for 36-node network problem

VI. DISCUSSION

From the tables of results, it is evident that MA returns better solutions than GA does for the 3 test problems. In fact MA solutions are better than GA solutions in both runs of the experiment for all the 3 test problems. The computation time(in seconds) of MA is however significantly more than that of GA. Also the best solution of MA is more reliable than the best solution of GA. Table 3 reveals that MA is able to find solutions (italicized and underlined) that are both better and cheaper (in computation time) than the best GA solution for 36-node network.

VII. CONCLUSION

A memetic algorithm is proposed for multi-objective design of communication networks. The performance of this algorithm was investigated and evaluated in relation to its genetic algorithm variant vis-à-vis solution quality and computation time. MA solutions are superior to those of GA. However MA is significantly more time consuming. Future work will focus on reducing the computational time. Besides, we will further investigate how an intelligent initialization and smarter local search mechanism will make impact on quality solutions and the overall performance of the algorithm. Currently the choice of mutation position in a chromosome is made randomly. A guided mutation based on the feature of the network topology is to be investigated.

REFERENCES

[1] R. Kumar, P. P. Parida, and M. Gupta. Topological Design of Communication Networks using Multiobjective Genetic Optimization. In *Proc. Congress Evolutionary Computation (CEC-2002)*, vol.1, pp. 425 – 430, May 2002.

[2] N. Banerjee, R. Kumar, "Multiobjective network design for realistic traffic models.", In *Proceedings of GECCO '07* , pp. 1904-1911, 2007.

[3] C. Papagianni, K. Papadopoulos, C. Pappas, N. D. Tselikas, D. T. Kaklamani, I. S Venieris. Communication Network Design Using Particle Swarm Optimization. *In Proceedings of the International Multiconference on Computer Science and Information Technology*. Vol. 3, 915 – 920, 2008.

[4] S. Duarte, B. Bar´an. Multiobjective Network Design Optimisation Using Parallel Evolutionary Algorithms. *XXVII Conferencia Latinoamericana de Inform´atica CLEI'2001, M´erida* - Venezuela. 2001

[5] B. Dengiz, F. Altipamark and A.E. Smith. Local Search Genetic Algorithm for Optimal Design of Reliable Networks. *IEEE Trans. Evolutionary Computation*, vol.1, no. 3, pp.179-188, Sept. 1997.

[6] F. Altiparmak, M. Gen, B. Dengiz, A.E. Smith. A Genetic Algorithm with Fuzzy Logic Controller for Design of Communication Networks. *IEEE Transactions on Electronics, Information and Systems*, vol.124-C, no. 10, pp. 1979-1985, Oct 2004.

[7] P. Kampstra. Evolutionary Computing in telecommunications: A likely EC success story. *Thesis*, VU university Amsterdam, The Netherlamds,2005.

[8] S.D. Flores, B.B. Cegla, D.B. C´aceres. Telecommunication Network Design with Parallel Multi-objective Evolutionary Algorithms. *In IFIP/ACM Latin America Networking Conference 2003 - Towards a Latin American Agenda for Network Research, IFIP - TC6 and ACM SIGCOMM*, La Paz, Bolivia October 3-5, 2003.

[9] M. Gen, L. Lin. Multiobjective hybrid genetic algorithm for bicriteria network design problem. *The 8th Asia Pacific Symposium on Intelligent and Evolutionary Systems*, December 2004.

[10] M. Gen, L. Lin, R. Cheng. Bicriteria Network Optimization Problem using Priority-based Genetic Algorithm. *IEEE Transactions on Electronics, Information and Systems*, vol. 124-C, no. 10, pp. 1972-1978, Oct 2004.

[11] J.R. Kim, M. Gen. Genetic algorithm for solving bicriteria network topology design problem. *In Proceedings of the 1999 IEEE Congress on Evolutionary Computation*, pp. 2272-2279, 1999

[12] R. Kumar, N. Banerjee. Multicriteria Network Design Using Evolutionary Algorithm. *In Lecture Notes in Computer Science*, vol. 2724, pp. 2179-2190, Jan 2003

[13] R. Kumar. Multicriteria Network Design Using Distributed Evolutionary Algorithm. *In Lecture Notes in Computer Science*, vol. 2913, pp. 343-352, Nov. 2003.

[14] R. Kumar, S. Prasanth, M.S. Sudarshan. Topological Design of Mesh Communication Networks using Multiobjective Genetic Optimisation. *In PPSN/SAB Workshop on Multiobjective Problem Solving from Nature (MPSN)*, Paris, France, September 2000.

[15] P, Moscato. On evolution, scorch, optimization. genetic algorithms and martial arts: toward memetic algorithms. *Technical Report*, California, 1989.

TEST DATA FOR 10-NODE NETWORK

NODE DETAILS
(code, cost)

01, 35
10, 49
01, 71
11, 51
01, 44
10, 30
00, 7
01, 73
10, 58
11, 40

LINK DETAILS
(-Dij, A, CAPACITY, DFLOW)

28 ,47, 60, 46
20, 43, 90, 72
28, 12, 54, 28
62, 39, 61, 46
42, 23, 24, 9
42, 30, 16, 14
36, 3, 44, 16
40, 18, 75, 54
10, 36, 29, 8
44, 30, 79, 53
44, 45, 54, 35
36,18, 66, 51
32,13, 78, 25
14,16, 96, 54
16,13,84, 74
21,28, 76, 17
22, 3, 80, 71
3,39, 55, 54
47,12, 66, 62
26,11, 89, 56
13,42,77, 47
46,22,45, 39
28,6,53, 16
5,38, 89, 57
28,40, 16, 9
48,49, 49, 40
18,34, 37, 9
34,35, 11, 8
11,4, 39, 31
46,20, 32, 9
11,3, 50, 35
70,1,54, 41
18,6, 8, 65
35,42, 91, 66
14,33, 10, 26
11,33, 60, 9
43,16, 79, 49
20, 43, 88, 56
16,13, 96, 68
6,30, 91, 67
34,49, 16, 7
37,21, 57, 49
20,12,79, 62
33,46, 81,70
48, 25, 8, 7