Computing, Information Systems, Development Informatics & Allied Research Journal
Vol. 6 No. 3, September, 2015 – www.cisdijournal.net

# Optimal Design of Communication Networks with Memetic Algorithm

**R. O. Oladele & H. A. Sanusi**
Department of Computer Science
University of Ilorin
P.M.B. 1515, Ilorin, Nigeria

**ABSTRACT**

Network design problem has been widely investigated using genetic algorithms. It is known that memetic algorithm solutions are better than genetic algorithm solutions. However, studies that investigate the behavior of memetic algorithm for various instances of multi-objective optimization problem such as network design are rare. In this paper, a memetic algorithm (MA) for communication network design problem is presented. The behavior of the MA is investigated for a set of problem instances ranging between 10 nodes and 36 nodes. The results show the most probable pareto-optimal result is not the same as the best result obtainable for 10-node network problem unlike in 36-node network problem where the most probable optimal result happened to be the best result. For 21-node network problem, the most probable optimal result (1167.4, 0.03) is better than the least probable optimal result (1167.4, 0.04) but ranks the same in quality with other likely result (1158.6, 0.04).

**Keywords: M**emetic algorithm, network design, NP-hard problems, local search, genetic algorithm

## 1. INTRODUCTION

The increasing complexity of the network design problems calls for advanced optimization techniques. Network design problems where even a single cost function is optimized are often NP-hard (Kumar & Banerjee, 2003). In addition, communication network design problems are not time critical. Therefore approaches based on meta-heuristics such as simulated annealing, tabu search, evolutionary computing, nature inspired algorithms (Pioro & Medhi, 2004), (Resende & Pardalos, 2006) have been designed to tackle these problems. Concerning evolutionary computing in telecommunication network design, a comprehensive study is presented in Kampstra (2005) up to 2005 containing relevant research study references, where network design problems are classified into node location problems, topology design, tree design, routing, restoration, network dimensioning, admission control and frequency assignment/wavelength allocation. The optimization techniques employed are mainly variations of Genetic Algorithms (GA). Additional work on telecommunication network optimization has followed in the last six years.

Real world network design problem normally involves simultaneous optimization of multiple and usually partially contradicting objectives. Therefore more often than not, there is not a single optimal solution, given the diversity of the set of objectives, but a set of congruent solutions, known as Pareto-optimal. The topological design of communication networks is usually a multi-objective problem involving simultaneous optimization of the cost concerning network deployment as well as various performance criteria (for example, average delay, throughput) subject to additional constraints such as reliability and bandwidth constraints. These problem-specific objectives are often opposing; for example over-provisioning is a strategy for reducing average delay in a network; that is to increase available link capacities which will consequently result in the increase of the total network deployment cost

Memetic Algorithm, recognized as a hybrid Genetic Algorithm (Moscato, 1989), (Moscato, 1999) has differentiated itself from the genetic algorithm classification. Unlike conventional genetic algorithms which emulate biological evolution, Memetic Algorithms imitate cultural evolution. Their conceptual difference is such that GA forbids individuals to choose, modify and improve their own genes in its natural process whereas MA allows individuals to intentionally acquire, modify, and improve their memes (Moscato, 1999), (Krasnogor & Smith, 2000). In other words, though GA is capable of finding good regions in the search space, the exploitation of these good regions needs more attention which GA is not designed for. Alternatively, MA has the local search procedure employed every time some new solutions are generated and emphasizes on local optimal solutions. Our literature survey shows that MAs are very effective and efficient in many hard combinatorial optimization problems (Krasnogor & Smith, 2000), (Herroelen, Demeulemeester, & Van Dommelen, 1997), (Merz & Freisleben, 2001), (Yin, 2004) but have been rarely applied in multi-objective network design problems. MA is considered and applied to solve multi-objective topology design problem in this paper.

## 2. LITERATURE REVIEW

Dengiz, Altiparmak & Smith (1997) proposed a GA for reliable network design. The comparison of the algorithm with the Branch and Bound method was made using 79 testing sets. These tests proved that the proposed GA is superior to the Branch and Bound method. The algorithm focuses only on the design of the network that should be reliable, neglecting cost and delay requirements. Therefore, the results obtained by the proposed GA can only be used as candidates for future investigation of its attributes. It cannot be used for real life network design.

In Cheng (1998), a comparison of different methods of network design optimization was done. An approach based on genetic algorithm was briefly compared with Branch-and-Bound method and simulated annealing technique. It was pointed out that the Branch-and-Bound method finds the global optimum but it is a very time consuming search technique. The simulated annealing is faster than the Branch-and-Bound method but often finds only a local optimum. On the other hand, the genetic algorithm-based approach is very fast but is not able to ensure global extreme finding. In opposite to simulated annealing, genetic algorithm provides parallel search and therefore global extreme finding is more probable than for simulated annealing. Aggarwal, Chopra & Bajwa (1982) employed greedy heuristic approach to maximize reliability given a cost constraint for networks with different reliability of links and nodes. Ventetsanopoulos & Singh(1986) used a two-step heuristic procedure for the problem of minimizing network's cost subject to reliability constraint.

This algorithm first used a heuristic to develop an initial feasible network configuration, and then a branch-and-bound approach was used to improve this configuration. These two algorithms are capable of designing cost-effective reliable network. Unfortunately, they cannot ascertain minimal average delay for data delivery. Again, the method cannot be as scalable as pure heuristic method. Kumar, Parida & Gupta (2002) 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 & Kumar (2007) studied multi-objective network design using an EA heuristic and empirically showed that the EA heuristic generally provides better solution compared to its deterministic counterparts.

Papagianni, Papadopoulos, Pappas, Tselikas, Kaklamani & Venieris (2008) 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 Banerjee & Kumar (2007) but the algorithm was only tested for 16-node network and nothing was said about its efficiency. Duarte & Baran (2001) 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 did in addition to its better efficiency. Watcharasitthiwat & Wardkein (2009) employed Improved Ant Colony Optimization (IACO) method to solve the topology network design problem considering both economics and reliability and discovered that IACO is superior to GA, Tabu Search algorithm, and ACO both in solution quality and computational time.

## 3. MATHEMATICAL FORMULATION

### 3.1 Network Design Parameters

For the purpose of this paper, the following network parameters are used.

**N** denotes the total number of nodes in the network

$\boldsymbol{D_{ij}}$ denotes the physical distance between every pair of nodes $\boldsymbol{i}$ and $\boldsymbol{j}$

$\boldsymbol{LFLOW_{ij}}$ is the amount of traffic (in Kbps) flowing along link ($\boldsymbol{i, j}$)

$\boldsymbol{C_{ij}}$ represents the cost of the link between nodes $\boldsymbol{i}$ and $\boldsymbol{j}$

$\boldsymbol{K_i}$ is the cost of network equipment at node $\boldsymbol{i}$

$\boldsymbol{CAP_{ij}}$ is the capacity of link ($\boldsymbol{i, j}$) i.e. the maximum amount of traffic the link can carry at any point in time

$\boldsymbol{P_{ij}}$ is selection status of link ($\boldsymbol{i, j}$) : $\boldsymbol{P_{ij}}$ = 1 if link ($\boldsymbol{i, j}$) is selected, else $\boldsymbol{P_{ij}}$ = 0

$\boldsymbol{R_0}$ is the minimum reliability required of the network ($\boldsymbol{R_0}$ = 0.95)

L is maximum distance for which the signal is sustained without amplification ( L is set to 15km)

A is the cost of each amplifier unit ( A is set to #6.00)

Poisson process was used to model the traffic delay

### 3.2 Objective Functions

Two objective functions; network cost and end-to-end message delivery delay were used (Banerjee & Kumar, 2007), each of which is approximated by the following formulation.

**1. Network Cost:**

$$NetCost = NodeCost + LinkCost + AmpCost \tag{1}$$

Where;

$$NodeCost = \sum_i K_i \quad \text{(cost of nodes)} \tag{2}$$

$$LinkCost = \sum_i \sum_j C_{ij} \quad \text{(cost of links)} \tag{3}$$

$$AmpCost = \frac{\sum_i \sum_j D_{ij} \; X \; A}{L} \quad \text{(cost of amplification units on the links)} \tag{4}$$

**2. Average Delay**

The traffic between nodes is modeled as a Poisson Process. Poisson process has been vastly used in the past to model traffic in conventional networks. For delay function, Poisson traffic model is utilized (Resende & Pardalos, 2006), (Banerjee & Kumar, 2007) and is as given in equation (6)

Queueing Delay for Poisson Traffic : The delay in a network for the queueing model is largely due to the queueing of packets in intermediate nodes. The delay formalization can be stated as follows:

$$AvDelay = \frac{\sum_i \sum_j \left[DELAY_{ij} \; X \; LFLOW_{ij}\right]}{\sum_i \sum_j LFLOW_{ij}} \tag{5}$$

From queueing theory, the queueing delay $DELAY_{ij}$ using standard M/M/1 (Poisson) model is given by:

$$DELAY_{ij} = \frac{1}{\left[CAP_{ij} - LFLOW_{ij}\right]} \tag{6}$$

$DELAY_{ij} = 0$ if there is no link between nodes $i$ and $j$

$DELAY_{ij} = \infty$ if the network cannot handle the traffic load with the existing links' capacities and routing policy.

### 3.3 Constraints

Minimization of Network Cost and Average Delay is done subject to the following constraints:
Flow Constraints which can be expressed as:

$$LFLOW_{ij} \leq CAP_{ij} \tag{7}$$

and

Reliability Constraint which can be expressed as:

$$R(x) \geq R_0 \tag{8}$$

Monte Carlo Simulation is used to estimate network reliability. The network is simulated t times, given the design and the links' reliabilities. Depth First Search (DFS) is used to implement spanning tree in the algorithm.

### 3.4 Routing Policy

Breadth First Search (BFS) is used for routing. The metric used for this purpose is the length of the link. Given a graph (topology) G = (V, E) and a distinguished source vertex s, BFS computes the distance (smallest number of edges) from s to each reachable vertex.

### 3.5 Assumptions

The following assumptions were made in the problem formulation

1 The location of each network node is given
2 Nodes are perfectly reliable
3 Each $C_{ij}$ is fixed and known

4 Each link is bidirectional i.e. a path can be traversed in either direction
5 There is no redundant link in the network
6 Links are either operational or failed

## 4. MA DESIGN

### *A. Memetic Algorithm*

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 offsprings 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. Implementation Details*

<u>Encoding Scheme</u>

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.

<u>Initial Population</u>

The algorithm starts 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 a repair strategy is used to replace the infeasible solutions with new feasible solutions.

<u>Fitness Evaluation</u>

Fitness of a chromosome is evaluated based on principle of Pareto ranking. Pareto-rank of each individual is one more than the number of individuals dominating it. All non-dominated individuals are assigned rank one. Network cost and average delay are used to evaluate the rank of an individual chromosome using the principle of Pareto dominance. The fitness of an individual as defined in Banerjee & Kumar (2007) is given by

$$Fitness = \frac{1}{Rank^2} \tag{9}$$

<u>Selection</u>

Two individuals are selected by Roulette wheel selection in which the probability of an individual i being selected is proportional to its fitness. Let $f_1, f_2, \ldots \ldots \ldots f_n$ be fitness values of individuals 1, 2, ………n. Then the selection probability, $P_i$, for an individual $i$, is given as

$$P_i = \frac{f_i}{\sum_{j=1}^{n} f_j} \tag{10}$$

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.

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.

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 as 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
```

## 5. 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 algorithm was implemented in Java. The results in bold (objective vectors) are the best pareto-ranked vectors for each run (that is, part of the pareto optimal front). Other algorithm design parameters used are:
Population size - 100 (250 for Problem 3)
Mutation probability – 0.02
Two- point crossover was used
Number of Node Type – 4
Test Problems:
Problem 1: 10-node network design
Problem 2: 21-node network design
Problem 3: 36-node network design.

**Tables of Results for Problem 1**

**Run 1**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 619.6 | 0.06 | 145 |
| 10 | 100 | **619.6** | **0.05** | 410 |
| 15 | 100 | 619.6 | 0.05 | 548 |
| 20 | 100 | 619.6 | 0.05 | 643 |

**Run2**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 619.6 | 0.06 | 145 |
| 10 | 100 | **619.6** | **0.05** | 410 |
| 15 | 100 | 619.6 | 0.05 | 548 |
| 20 | 100 | 619.6 | 0.05 | 643 |

**Run3**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 619.6 | 0.06 | 145 |
| 10 | 100 | **619.6** | **0.05** | 410 |
| 15 | 100 | 619.6 | 0.05 | 548 |
| 20 | 100 | 619.6 | 0.05 | 643 |

**Run4**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 619.6 | 0.06 | 145 |
| 10 | 100 | **619.6** | **0.05** | 410 |
| 15 | 100 | 619.6 | 0.05 | 548 |
| 20 | 100 | 619.6 | 0.05 | 643 |

**Run 5**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 619.6 | 0.06 | 145 |
| 10 | 100 | **619.6** | **0.05** | 410 |
| 15 | 100 | 619.6 | 0.05 | 548 |
| 20 | 100 | 619.6 | 0.05 | 643 |

**Run 6**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 534.0 | 0.06 | 166 |
| 10 | 100 | **534.0** | **0.05** | 472 |
| 15 | 100 | 534.0 | 0.06 | 537 |
| 20 | 100 | 534.0 | 0.07 | 615 |

**Run7**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 534.0 | 0.06 | 166 |
| 10 | 100 | **534.0** | **0.05** | 472 |
| 15 | 100 | 534.0 | 0.06 | 537 |
| 20 | 100 | 534.0 | 0.07 | 615 |

**Tables of Results for Problem 2**

**Run1**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 954 |
| 10 | 100 | 1167.4 | 0.05 | 1892 |
| 15 | 100 | 1167.4 | 0.04 | 4467 |
| 20 | 100 | **1158.6** | **0.04** | 5665 |

**Run 2**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 910 |
| 10 | 100 | 1167.4 | 0.05 | 1989 |
| 15 | 100 | 1167.4 | 0.04 | 4320 |
| 20 | 100 | **1167.4** | **0.03** | 5766 |

**Run 3**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 954 |
| 10 | 100 | 1167.4 | 0.05 | 1892 |
| 15 | 100 | 1167.4 | 0.04 | 4467 |
| 20 | 100 | **1158.6** | **0.04** | 5665 |

**Run 4**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 910 |
| 10 | 100 | 1167.4 | 0.05 | 1989 |
| 15 | 100 | 1167.4 | 0.04 | 4320 |
| 20 | 100 | **1167.4** | **0.03** | 5766 |

**Run 5**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 910 |
| 10 | 100 | 1167.4 | 0.05 | 1989 |
| 15 | 100 | 1167.4 | 0.04 | 4370 |
| 20 | 100 | **1167.4** | **0.03** | 5766 |

**Run 6**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 98 | 1167.4 | 0.06 | 910 |
| 10 | 100 | 1167.4 | 0.05 | 1989 |
| 15 | 100 | **1167.4** | **0.04** | 4322 |
| 20 | 100 | 1167.4 | 0.09 | 5756 |

**Run 7**

| | | | | | |
|---|---|---|---|---|---|
| 5 | MA | 1259 | 0.06 | 910 | |
| 10 | No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 15 | 100 | 1167.4 | 0.04 | 4320 | |
| 20 | 100 | **1167.4** | **0.03** | 5766 | |

**<u>Tables of Results for Problem 3</u>**

**Run 1**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 3987 |
| 10 | 100 | **1167.4** | **0.05** | 5673 |
| 15 | 100 | 1167.4 | 0.05 | 9210 |
| 20 | 100 | 1167.4 | 0.05 | 14900 |
| 25 | 100 | 1167.4 | 0.05 | 18634 |

**Run 2**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1167.4 | 0.06 | 3598 |
| 10 | 100 | 1167.4 | 0.05 | 5890 |
| 15 | 100 | 1167.4 | 0.05 | 9673 |
| 20 | 100 | 1167.4 | 0.05 | 15897 |
| 25 | 100 | **1167.4** | **0.04** | 18754 |

**Run 3**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1259 | 0.06 | 3987 |
| 10 | 100 | **1167.4** | **0.05** | 5673 |
| 15 | 100 | 1167.4 | 0.05 | 9210 |
| 20 | 100 | 1167.4 | 0.05 | 14900 |
| 25 | 100 | 1167.4 | 0.05 | 18634 |

**Run 4**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1167.4 | 0.06 | 3598 |
| 10 | 100 | 1167.4 | 0.05 | 5890 |
| 15 | 100 | 1167.4 | 0.05 | 9673 |
| 20 | 100 | 1167.4 | 0.05 | 15897 |
| 25 | 100 | **1167.4** | **0.04** | 18754 |

**Run 5**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1167.4 | 0.06 | 3598 |
| 10 | 100 | 1167.4 | 0.05 | 5890 |
| 15 | 100 | 1167.4 | 0.05 | 9673 |
| 20 | 100 | 1167.4 | 0.05 | 15897 |
| 25 | 100 | **1167.4** | **0.04** | 18754 |

**Run 6**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1167.4 | 0.06 | 3598 |
| 10 | 100 | 1167.4 | 0.05 | 5890 |
| 15 | 100 | 1167.4 | 0.05 | 9673 |
| 20 | 100 | 1167.4 | 0.05 | 15897 |
| 25 | 100 | **1167.4** | **0.04** | 18754 |

**Run 7**

| MA | | | | |
|---|---|---|---|---|
| No of Gen | C/Ratio | Cost | AvDelay | CPU Time |
| 5 | 97 | 1167.4 | 0.06 | 3598 |
| 10 | 100 | 1167.4 | 0.05 | 5890 |
| 15 | 100 | 1167.4 | 0.05 | 9673 |
| 20 | 100 | 1167.4 | 0.05 | 15897 |
| 25 | 100 | **1167.4** | **0.04** | 18754 |

## 6. DISCUSSION

From the tables of results, it is evident that the time taken by MA to return solution increases with the number of generation and with the network size. The results also show that: (1) for 10-node network problem, the most probable pareto-optimal result is **(619.6, 0.05)** which was obtained in 5 runs out of the 7 runs. However, a better result of **(534.0, 0.05)** is obtainable. (2) for 21-node network problem, the most probable pareto-optimal result is **(1167.4, 0.03)** which was obtained in 4 runs out of the 7 runs. (3) for the 36-node network problem, the most probable pareto-optimal result is **(1167.4, 0.04)**

## 7. CONCLUSION

A memetic algorithm is designed and implemented for multi-objective design of communication networks. The algorithm is effective for finding optimal results. Our future work will focus on investigating the impact of intelligent initialization and smarter local search mechanism on solution quality 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 also to be investigated.

## REFERENCES

1. Aggarwal, K. K., Chopra, Y. C., Bajwa, J. S.(1982): Reliability evaluation by network decomposition. IEEE Trans Reliab 1982;R-31:355–358
2. Banerjee, N., Kumar, R. (2007): Multiobjective Network Design for Realistic Traffic Models. In *Proc. Genetic and Evolutionary Computations Conference (GECCO-07)*, pp. 1904-1911.
3. Cheng, S. T.(1998): Topological optimization of a reliable communication network. IEEE Transactions on Reliability, vol. 47, pp. 225 –233.
4. Dengiz, B. Altiparmak, F. and Smith, A. (1997): Local search genetic algorithm for optimal design of reliable networks. IEEE Transactions on Evolutionary Computation, vol. 1, pp. 179 –188.
5. Duarte, S., Baran, B. (2001): Multiobjective Network Design Optimization Using Parallel Evolutionary Algorithms. *XXVII Conferencia Latinoamericana de Inform´atica CLEI'2001, M´erida* - Venezuela.
6. Herroelen, W. S., Demeulemeester, E., P. Van Dommelen, P. (1997): Project network models with discounted cash flows: A guided tour through recent developments. *Eur. J. Oper. Res.*, vol. 100, pp. 97-121.
7. Kampstra, P (2005): Evolutionary Computing in telecommunications, A likely EC success story. M.Sc Thesis, VU University Amsterdam, The Netherlands.
8. Krasnogor, N., Smith, J. (2000): A memetic algorithm with self-adaptive local search: TSP as a case study, in: *Proceedings of the Genetic and Evolutionary Computation Conference* (GECCO-2000), Morgan Kaufman, pp. 987- 994
9. Kumar, R., Banerjee, N. (2003): Multicriteria network design using evolutionary algorithm, *In:* Proceedings of Genetic Computation Conference (GECCO), pp. 2179-2190.
10. Kumar, R., Parida, P. P., and Gupta, M.(2002): Topological design of communication networks using multiobjective genetic optimization. In Proc. Congress Evolutionary Computation (CEC-2002), pages 425 – 430. IEEE Press.
11. Merz, P., Freisleben, B.(2001): Memetic algorithms for the traveling salesman problem. *Complex Syst.*, vol. 14, pp.297-345.
12. Moscato, P. (1989): *On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms,* Caltech Concurrent Computation Program, Report No. 826.
13. Moscato, P. (1999): Memetic algorithms: a short introduction, in D. Corne, F. Glover, and M. Dorigo (eds.), *New Ideas in Optimization*, McGraw-Hill, pp. 219-234..
14. Papagianni, C., Papadopoulos, K., Pappas, C., Tselikas, N. D., Kaklamani, D. T., Venieris, I. S.(2008): Communication Network Design Using Particle Swarm Optimization. *In Proceedings of the International Multiconference on Computer Science and Information Technology*. Vol. 3, 915 – 920.
15. Pioro, M., Medhi, D. (2004): Routing, flow and capacity design in communication and computer networks, Morgan-Kaufmann.
16. Resende, M., Pardalos, P. (2006): Handbook of Optimization in Telecommunications, Springer
17. Ventetsanopoulos, A. N. and Singh, I.(1986): Topological Optimization of Communication Networks Subject to Reliability Constraints," *Problem of Control and Information Theory*, **15**, 63-78.
18. Watcharasitthiwat, K., Wardkein, P.(2009): Reliability Optimization of topology communication network design using an improved ant colony optimization". Computers and Electrical Engineering, vol. 35, pp. 730 – 747. Elsevier.
19. Yin, A (2004): A new neighborhood structure for the job shop scheduling problem, In *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics,* August 26-29, Shanghi, vol. 4, pp. 2098-2101.

**TEST DATA FOR 10-NODE NETWORK**

Node Details (NodeType, $K_i$) = { (01,42) ,(0, 78) ,(10,33) ,(00,53) ,(01,42) ,(00,13) ,(10,9) ,(11,23) ,(10,57) ,(10,25) }

Link Details ( $D_{ij}$ , $C_{ij}$, $CAP_{ij}$, $LFLOW_{ij}$ ) =

{ (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),(11,41,39,31) ,(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) }