ISSN 2229-5518

GA based optimal design of network architecture for desired connectivity and traffic demand

S.V.Uma, K.S.Gurumurthy, Manoj Kumar Singh

Abstract— Designing of a network which could fulfill most of the requirements is always a challenging task for a researcher. Often this happens either with manual approach or by applying some kind of conventional methods. In both cases results do not have high level of optimality. Evolutionary computation is a beacon of hope in handling complex design problems in an efficient manner. To reduce the effect of congestion, various approaches can be applied at different levels namely over Architecture level and Protocol level. The performance of a protocol completely depends upon the facilities available with existing architectures which are stagnant. Hence our research aims at improving the connectivity according to demand in the network having minimum cost of architecture, which is a better alternative and a very efficient way to handle network congestion and reliability. Such kind of network design is a very tedious task. Hence involment of intelligence incorporated into the network design for automatic synthesis is a must. Automation of this design in this paper is done using genetic algorithm. This paper proposes a technique based on Genetic algorithm, which uses a new method of two point crossover, a different and efficient technique of fitness evaluation and tournament selection. The simulations yielded an automatic network architecture which satisfied the requirement of connectivity and minimum cost, fulfilling the traffic demand all along for varying number of nodes and connectivity constraints.

.

Index Terms— Network design, Congestion control, Connectivity, Dynamic objective function, Traffic matrix, Evolutionary computation, Genetic algorithm.

1 INTRODUCTION

Modeling and design of large communication and computer networks have always been an important area to both researchers and practitioners. The interest in developing efficient design models and optimization methods have been stimulated by high deployment and maintenance cost of networks, which makes good network design potentially capable of securing considerable savings. In the past decade networks have undergone a substantial change, caused by the emergence and rapid development of new technologies and services, an enormous growth of traffic, demand for service availability and continuity, and attempts to integrate new networking system techniques and different types of services in one network. As a consequence, today's network designers face new problems associated with diverse technologies, complicated network architectures, and advanced resource and service protection mechanisms. Network architecture and planning to minimize the cost and also build congestion free optimal connectivity is certainly a great challenge today. The motivation for our work arises from the great diversity of questions and problems that originate from today's network planning tasks, which require a large number of algorithms, each of which specializes in a specific problem with specific constraints.For most problems, there is no known algorithm that could guarantee to find the global optimum in a polynomial amount of time. Hence the use of Evolutionary computation is a better choice. It is seen that Evolutionary computation, offers practical advantages to the researcher facing difficult optimization problems. These advantages are multi-fold, including the simplicity of the approach, its robust response to changing circumstances, its flexibility, and many other facets. The evolutionary approach can be applied to problems where heuristic solutions are not available or generally lead to unsatisfactory results. In many cases the mathematical function, which describes the problem is not known and the values at certain parameters are obtained from simulations. In contrast to many other optimization techniques, an important advantage of evolutionary algorithms is they can cope with multi-modal functions.

To handle the complexity associated with network design, various models and related solution techniques have been proposed, including mathematical programming models [1]. Standard approaches for dealing with uncertainty include numerically-intensive scenario techniques that basically solve a large number of problem instances for randomly generated parameter patterns [2]. While useful, scenario techniques often lead to computationally demanding problems that may not be easily solved already for medium scale networks. In many cases, sophisticated heuristics have to be developed to achieve satisfying results. Mostafa Abd-El-Barr' et al [3] proposed the use of three iterative techniques, namely Tabu Search, Simulated Annealing, and Genetic Algorithm, in solving the multi-objective topological optimization network design problem for fault tolerance and reliability of networks. This design could not take inputs such as traffic demand and cost and adhere to good reliability.Mitsuo GEN, Kenichi IDA & Jongryul KIM [4] proposed an evolutionary algorithm by employing spanning

- *S.V.Uma is presently working as an assistant professor in R.N.S.I.T, Bangalore under the department of E&C and also she is currently pursuing her Ph.D in the Electronics and Communication department of Bangalore University in communication network. E-mail: umakeshav2000@gmail.com.*
- *K.S.Gurumurthy is Presently working as a professor in the DOS in E & CE, UVCE, BU, Bangalore He obtained his Ph.D degree in the area of 'Microelectronics' He is a member of IEEE and ISTE. E-mail: drksgurumurthy@gmail.com*
- *M.K.Singh is presently holding the post of director in Manuro Tech. Research.He is also a technology consultant for advanced research in the area of soft computing. He is a member of IEEE. E-mail: mksingh@manuroresearch.com*

tree-based genetic algorithm to solve bicriteria LAN topology design problems and also developed the method to search for the Pareto solutions. Sun-Jin Kim and Munkee Choi [5] proposed a genetic algorithm (GA) to design a non-hierarchical and decentralized Multimedia on-Demand (MOD) network architecture to optimize the MOD network resource based on cost analysis. Anton Riedl [6] has proposed a genetic algorithm which guides the search towards an optimal point by applying genetic operators to specific solutions of a given problem and applied his concept to two very different fields in network planning: To minimize the costs of fiber ducts when building a new passive optical network and in the context of packet switched network planning.

Various optimization techniques have been used for network designing till today. Fan Li Yu Wang [7] proposed a novel grid-based gateway deployment method using a cross-layer throughput optimization scheme giving better performance than random deployment and fixed deployment methods. But this technique is efficient only with wireless networks and cannot be extended to work with other networks. For effective traffic allocation on the links, when uncertain demands and capacities are modeled as unknown-but-bounded quantities restricted in intervals, the resulting robust decision problem can still be formulated as a linear program and solved at the same computational cost as its nominal counterpart[8].Chun-Yen Hsu* et al. [9] proposed the Pre-defined Gateway Set Algorithm (PGSA) using enhanced Dijkstra's algorithm and genetic algorithm to arrange the network configuration, including the gateways and the topology, subject to degree and delay constraints, such that the construction cost of the backbone wireless mesh networks is minimized.

Network design optimization problems which can also guarantee QoS to a certain extent become equally challenging and important [10]. Cem Ersoy et al. [11] proposed two approaches based on simulated annealing (SA) and genetic algorithms (GA) for the link capacity assignment in multiservice networks in which different classes of packets with different packet lengths, priorities and performance requirements exist. While they compared their algorithm performance with that of a standard heuristic by Maruyama and tang they have not considered cost structure supporting the charge on flow basis utilized for billing today, routing and nodal propagation delays in their design. Cost effective routing with emphasis on network congestion have also been discussed [12]. Genetic algorithms can also be improved using various techniques so as to yield better optimization and network design. [13], [14]. Usually grouped under the term evolutionary computation or evolutionary algorithms, we find the domains of genetic algorithms, evolution Strategies, evolutionary programming and genetic programming. They all share a common conceptual base of simulating the evolution of individual structures via processes of crossover, mutation, reproduction and selection. The processes depend on the perceived performance of the individual structures as defined by the problem. The genetic algorithm as an evolutionary approach is the best suited solution platform. In our approach for optimal network design, a population of candidate solutions (for the optimization task to be solved) is initialized as the first step. New solutions are created by applying reproduction operators (mutation and/or crossover). The fitness (how good the solutions are) of the resulting solutions are evaluated and a suitable selection strategy is then applied to determine which solutions will be maintained into the next generation.

The rest of the paper is organized as follows: We discuss the advantages of evolutionary algorithms in Section 2, the general genetic algorithm is introduced in section 3, and design of our own robust genetic algorithm fulfilling the traffic demand and cost requirements is explained in sections 4 and 5. The experimental setup, simulation and result analysis are given in section 6 and finally section 7 concludes the paper.

2 ADVANTAGES OF EVOLUTIONARY ALGORITHMS

A primary advantage of evolutionary computation is that it is conceptually simple. The procedure may be written as a difference equation:

$$x[t + 1] = s(v(x[t])) \quad (1)$$

Where $x[t]$the population at time t is , v is a random variation operator, and s is the selection operator. Other advantages can be listed as follows:

- Evolutionary algorithm performance is representation independent, in contrast with other numerical techniques, which might be applicable for only continuous values or other constrained sets.
- Evolutionary algorithms offer a framework such that it is comparably easy to incorporate prior knowledge about the problem. Incorporating such information focuses the evolutionary search, yielding a more efficient exploration of the state space of possible solutions.
- Evolutionary algorithms can also be combined with more traditional optimization techniques. This may be as simple as the use of a gradient minimization used after primary search with an evolutionary algorithm (for example fine tuning of weights of a evolutionary neural network), or it may involve simultaneous application of other algorithms (e.g., hybridizing with simulated annealing or Tabu search to improve the efficiency of basic evolutionary search).
- The evaluation of each solution can be handled in parallel and only selection (which requires at least pair wise competition) requires some serial processing. Implicit parallelism is not possible in many global optimization algorithms like simulated annealing and Tabu search. [1]
- Traditional methods of optimization are not robust to dynamic changes in problem environment and often require a complete restart in order to provide a solution (e.g., dynamic programming). In contrast, evolutionary algorithms can be used to adapt solutions to changing circumstances.
- Perhaps the greatest advantage of evolutionary algorithms comes from the ability to address problems for which there are no human experts. Although human expertise should be used when it is available, it often proves less than adequate for

automating problem-solving routines.Evolutionary algorithms are ubiquitous nowadays, having been success- fully applied to numerous problems from different domains, including optimization, automatic programming, machine learning, operations research, bioinformatics, and social systems.

3 GENETIC ALGORITHM

A typical flowchart of a Genetic Algorithm (GA) is depicted in Fig.1. One iteration of the algorithm is referred to as a generation. The basic GA is very generic and there are many aspects that can be implemented differently according to the problem (For instance, representation of solution or chromosomes, type of encoding, selection strategy, type of crossover and mutation operators, etc.) In practice, GA is implemented by having arrays of bits or characters to represent the chromosomes. The individuals in the population then go through a process of simulated evolution. Simple bit manipulation operations allow the implementation of crossover, mutation and other operations. The number of bits for every gene (parameter) and the decimal range in which they decode are usually the same but nothing precludes the utilization of a different number of bits or range for every gene.

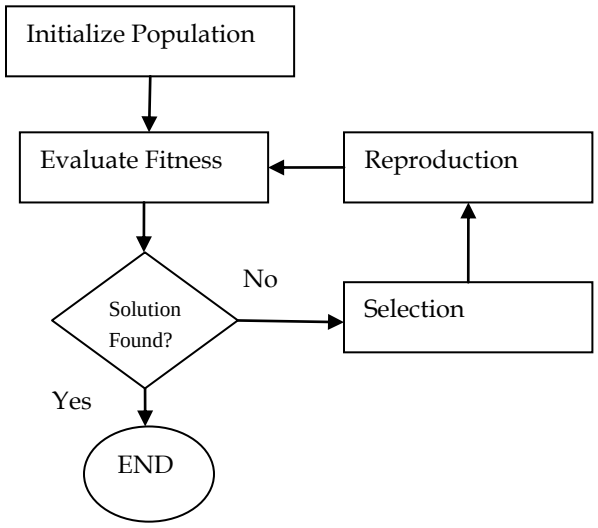

Fig .1 Flow chart of basic genetic algorithm iteration

4 DESIGN OF NETWORK WITH GA

The concept of GA has applied to find the automated design of network with desired objectives. Success of genetic algorithm completely depends upon how the genetic operations have been applied to get the result. Various genetic operators have modeled as shown in following section to implement the genetic algorithm.

4.1 Chromosome representation

A binary coding matrix developed to represent the architecture of network is as shown below. Each row and column represents one node in the network. If there is connection between ith to jth nodes in network, element in S_{ij} will have the value equal to 1 otherwise it will be 0.

$$S = \begin{matrix} \\ R \\ \\ \\ \\ \\ \\ \\ \\ \\ \end{matrix} \begin{matrix} \rightarrow \\ \begin{bmatrix} 1&0&1&1&0&1&1&0&0&1 \\ 0&1&1&0&0&0&1&1&0&0 \\ 0&0&0&1&1&0&1&0&0&1 \\ 1&0&0&0&1&1&1&0&0&0 \\ 0&1&1&0&1&1&1&0&0&0 \\ 1&1&1&1&0&0&0&1&1&1 \\ 1&0&1&1&0&0&0&1&1&1 \\ 0&1&1&0&0&1&1&0&0&0 \\ 1&1&0&0&1&1&1&0&0&0 \\ 0&0&0&1&1&0&1&0&1&1 \end{bmatrix} \\ C \end{matrix}$$

4.2 Objective function

Consider an undirected graph G (n,m),in which n is the number of nodes and m is the number of connections. In the case of complete graph, the number of connections is given by $m = n(n-1)/2$. Assume that each connection can have integer values from 0(branch does not exist) to b(b possible types are available).Greater values should represent a " stronger" connection, i.e. branch types with higher capacity such as communication links with higher bandwidth. Let N be any sub graph which can be formed within G (n,m).The following expression is proposed here in order to represent the network N in the solution space:

$$N = \sum_{i=1}^{m} w_i^N . \left(P + \left(\phi(N_i)\right)\right) \quad (2)$$

Where

W_i^N Topological weight of connection i in network N. This weight is a positive integer if the connection i exists and is 0 otherwise;

N_i Branch type of connection i in network$N$, with Error! Bookmark not defined. $0 \leq N_i \leq b$ these values should be ordered such that branch type which corresponds to stronger connections receives larger values of N_I;

$\phi(.)$ Monotonically increasing function.

P Constant factor such that $P \geq \phi(b)$

If a connection i does not exist, W_i^N=0 then the ith space coordinate is zero. Otherwise, the topological weight W_i^N is assigned to be a value that can be interpreted as a measure of topological importance of connection i in the network N. For instance, in tree-structure network, a change in connections nearer to the root is expected to cause larger impacts in the network flow than changing connections nearer to the leaves. The position of a connection nearer to the root would therefore

be associated with a greater value of W_i^N. Possible formulations for defining W_i^N could be, for instance, measuring the distance to the root, or the distance to the farthest leaf, or even the number of nodes above the connection. The specific rules to define W_i^N should be stated taking into account the characteristics of each problem. Another option for defining W_i^N would be to make it equal to the length of connection I, when such a length has an important role in the specific system under study. In eq (2) the increasing function $\phi(N_i)$ which has as an argument- the integer N_i (which ranges from 1 to b) expresses the strength of connection I [stronger branch types are associated to larger value of N_i and, therefore of $\phi(N_i)$, too]. The inequality $P \geq \phi(b)$ means that the effect of putting or removing a branch in the network is greater than the effect of changing the type of a branch, in the evaluation of expression $(P + (\phi(N_i)))$.The resulting value of $w_i^N.(P + (\phi(N_i)))$, which is assigned to the ith connection has the following features.

(1) It presents small variations for change in branch type.
(2) It presents large variations when a connection is inserted or removed.
(3) It is larger for connections with larger topological weights, i.e., for connections which are "more important" in a given network topology.

4.3 Fitness function:

To define the fitness of chromosome, a combination of constraint violation (cv) with penalty factor (pf) and objective function is created as shown by eq (3). This function minimizes to get the fitness of the solution. A Solution having a minimum value of f is having a higher value of fitness. When constraints are satisfied or they reach up to an optimum level, solutions become feasible, otherwise evolution is used to keep searching for feasible solutions using the genetic operator.

$$f = cv * pf + \sum_{i=1}^{m} w_i^N.\left(P + \left(\phi(N_i)\right)\right) \quad (3)$$

4.4 Crossover

A new method to apply two point crossovers is applied. Two random numbers r1 and r2 generated between 1 to NN, where NN is the total number of nodes available in the network. All the elements from one parent say row r1 to r2 are exchanged with all elements of the other parent from row r1 to r2.The above defined process of crossover is shown in Fig (2).The probability of crossover P_c is considered equal to 1. A single point crossover is also experimented where a single random number r1 is generated between 1 to NN and elements from parents are exchanged only with respect to that particular row r1.The action of cross-over is shown in Fig (2).

4.5 Mutation

The probability of mutation P_m, is taken as 0.1 for each feasible connection in network. To estimate the probability of mutation of a particular connection, a random number mr, in the range [0 1], is generated using uniform distribution. If $mr < P_m$ then that particular connection is flipped, i.e. if a connection exists then the connection is removed or if a connection does not exist then a connection is created. If $mr > P_m$ then that particular connection will not be mutated and the connection status of that particular position will remain unaltered.

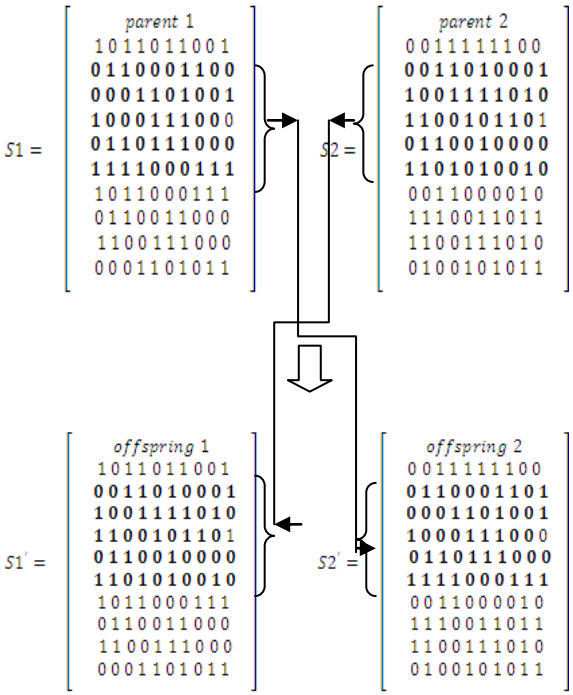

Fig 2.0: Two point all node crossover is shown with selected points [2 5]

4.6 Selection

Selection of suitable candidates for the next generation is one of the most important parts in evolutionary computation. This process defines the exploration and exploitation balance in generations. The Selection method applied in the present work is tournament selection.

Pseudo code of tournament selection

1. Create a combined population [parents & offsprings]
2. For i =1 to $2P$
   (Where P is the size of the population)
3. Chset =Select T number of challengers randomly between 1 to 2P
4. For j= 1to T
5. Pickup one challenger from Chset in sequence
6. Tourres=fitness(^{ith} solution)> fitness(challenger);
7. Winscore(i)=winscore(i)+Tourres;
8. End
9. End
10. Sort Winscore in increasing order,
11. Mxscore= Right half scores having maximum value,
12. Selected_chr= Chromosomes corresponding to Mxscore
13. Next generation population ,NXTGEN=selected_chr;

4.7 Termination:

A very important characteristic of any evolutionary algorithm is how termination of evolution is defined within the model. Practically two aspects are available for termination, namely (1) forced termination (2) self termination. Forced termination is applied where there is a constraint defined either in terms of available execution time, or when there is interest to see the performance of evolution after certain number of generations. In this kind of termination there is no guarantee of optimality at the end process. This type of termination is generally selected for design verification or when design is completed. In self termination, point of termination is decided by the program itself, usually when evolution does not find improvement even after a number of generations. This happens in two cases, either when optimality in the solution has been achieved by evolution or when the design has lost the diversity in population. If the difference in fitness of best chromosome for continuous 'Gn' number of generations is less than certain defined threshold, then it is terminated.

5 GENETIC ALGORITHM

Perform initialization:

1) Generate the initial population with procedure using method of chromosome representation.
2) Define $P_c = 1\ and\ P_m = 0.1$
3) Evaluate the objective function of each individual;
4) Assign the fitness value to each individual,

While not stop criterion

Perform binary operation:

(i)Arrange the population pair wise (randomly, with uniform probability distribution);
(ii)For each pair of individuals:
(a) Generate two random numbers $1 \leq r_1, r_2 \leq N$ with uniform probability distribution;
(b) Perform crossover operation;

Perform unary operations:

(iii) For each individual:
For each position
(a) Generate a random number $0 \leq mr \leq 1$ with uniform probability distribution
(b) if $mr \leq P_m$ then chose the mutation operation and perform the action.

Perform function evaluation and selection:

(iv) Evaluate the objective function for each newly generated individual;
(v) Assign a fitness value to each individual of the population;
(vi) Perform selection using tournament method

End while

5) Perform local search in the last generation for the best solution found by the genetic algorithm.

6. SIMULATION AND RESULTS

A network with 10 nodes is considered in our work to verify the proposed algorithm. The connection feasibility and link cost requirements are as shown in table 1.0[Appendix] A position containing '∞' indicates that there is no connectivity between the corresponding nodes. Table 2.0[Appendix] contains the traffic matrix that represents the demands between any two nodes. Initial population of 100 chromosomes is created randomly. Each solution has a matrix structure as shown in Fig.1. Probability of crossover is equal to 1 whereas mutation probability is equal to 0.1. In all experiments, self termination has been applied. Simulation and performance analysis is done for three different connectivity (k) constraints $(a)k \geq 2\ \ (b)k \geq 3\ \ (c)k \geq 4$.

The performance in all cases in terms of cost of architecture, connectivity constraint, and required number of generations is measured. Graphs corresponding to cost optimization and constraint violation are also shown. To understand the internal behavior of GA various aspects have been defined, like:

(a) Contribution of parent and offspring population in creating the next generation, i.e in the next generation population how many belong to the parents' population and how many belong to offspring' population.
(b) On termination, how many times the parent's population has given the best chromosome and how many times offspring's population contributed.

Performances for three different connectivity constraints are shown in Fig. (4), (5) and (6). Fig.(x)(a) represents the performance of GA to minimize the cost for given connectivity constraint. In all cases it is very clear that the design procedures of the GA converge very quickly and stabilize. Some small fluctuation seen is the result of mutation operation. Fig.(x) (b) represents the total violation appearing with each generation progressively. In case of $k \geq 2$, there is no violation, whereas for $k \geq 3\ and\ k \geq 4$, violation appears, because it is impossible to define the connectivity in the network, for the given link matrix. This type of data set is purposely chosen so as to verify both sides of the possibility. For the $k \geq 3\ and\ k \geq 4$, it is very clear that the proposed GA maintains the best possibilities within very few generations. Fig.(x) (c) shows the network architecture which has the minimum cost and satisfies the constraint of connectivity.

Fig.(x)(d) represents the performance of parent population and offspring population with respect to generations, in the contribution of number of members they provided for next generation. From all the graphs it is clear that at the beginning parents contribution is more compared to the offspring population, but in later generations both populations contribute more or less in the same manner. Figure (e) is the cumulative plot, upper one for parent population and lower for offspring population, about who provides the larger number of best solutions in the complete process. The plot very clearly shows that the parent population gives the larger number of best solutions as compared to the offspring population, emulating the natural behavior in real life, where experience of the parents offers a better solution more often than offsprings. The connection matrix for each case is shown in Figure (f). A position value of 1 represents the presence of a connection, whereas a 0 represents no connection. Finally the cost value for all cases is shown in Fig 3.0.

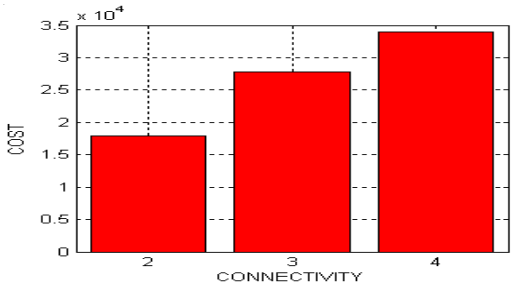

Fig 3.0 Cost of network for different connectivity

TABLE 1
Network cost for various K

k	Network Cost
2	1.7912e+004
3	2.7775e+004
4	33978

6 PERFORMANCE RESULT

6.1 Network design performance for k=2

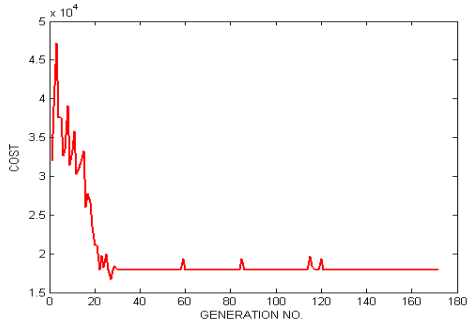

Fig. 4(a)

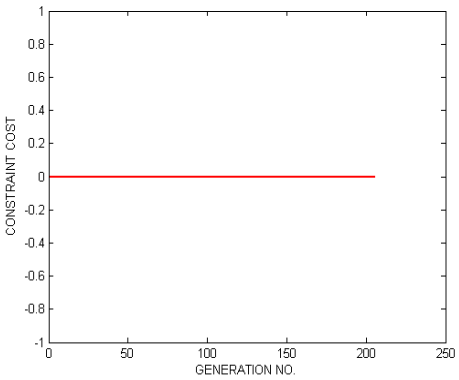

Fig. 4(b)

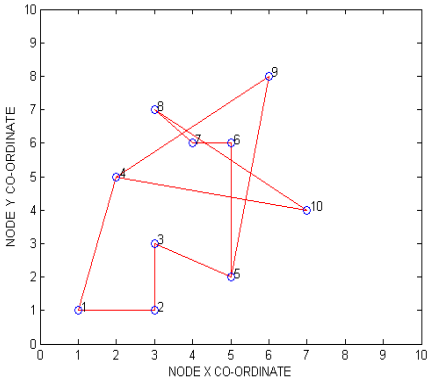

Figure 4(c)

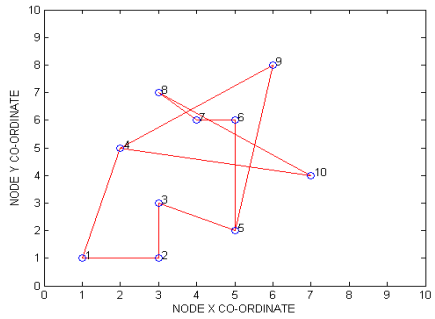

Fig. 4(d)

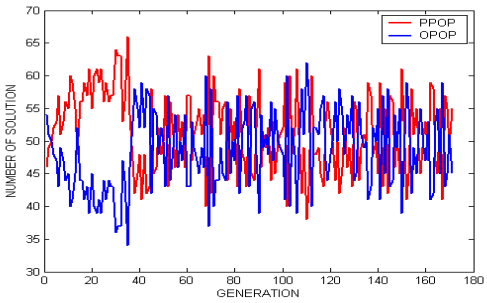

Fig. 4(e)

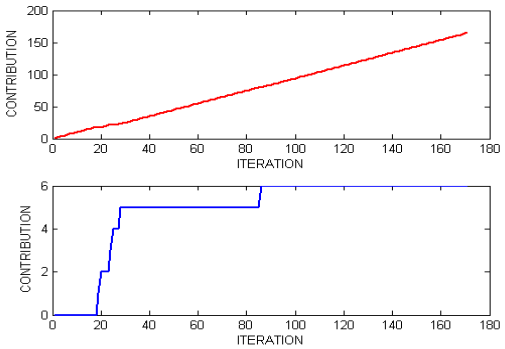

Figure 4(f)

TABLE 2
Generated architecture connection

		TO									
	N	1	2	3	4	5	6	7	8	9	10
F	1	0	1	0	1	0	0	0	0	0	0
R	2	0	0	1	0	0	0	0	0	0	0
O	3	0	0	0	0	1	0	0	0	0	0
M	4	0	0	0	0	0	0	0	0	1	0
	5	0	0	1	0	0	0	0	0	0	0
	6	0	0	0	0	1	0	1	0	0	0
	7	0	0	0	0	0	1	0	0	0	0
	8	0	0	0	0	0	0	1	0	0	0
	9	0	0	0	0	1	0	0	0	0	0
	10	0	0	0	1	0	0	0	1	1	0

6.2 Network design performance for k≥3

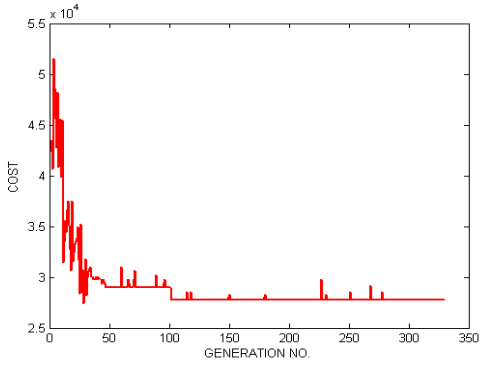

Fig. 5(a)

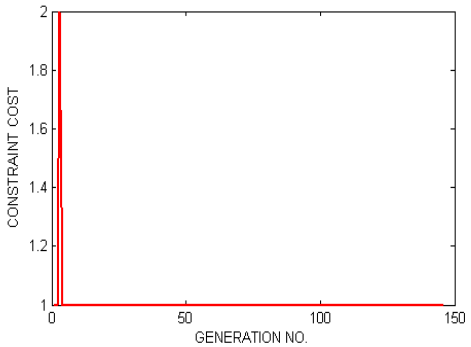

Fig. 5(b)

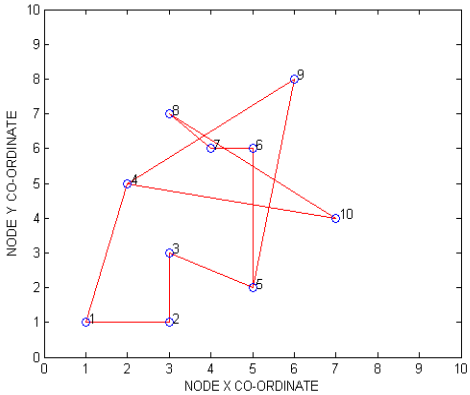

Fig. 5(c)

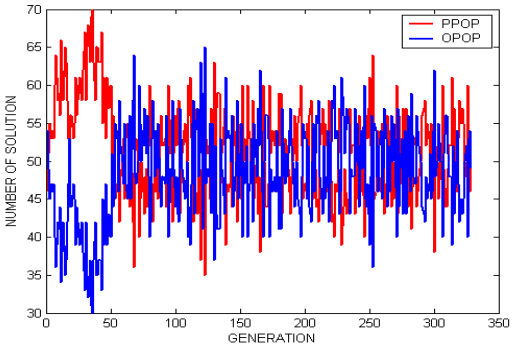

Fig. 5(d)

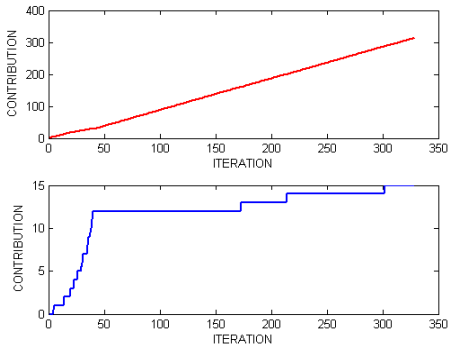

Fig. 5(e)

TABLE 3
Generated architecture connection

FROM \ TO	N	1	2	3	4	5	6	7	8	9	10
	1	0	1	1	1	0	0	0	0	0	0
	2	0	0	0	0	1	0	0	0	0	0
	3	0	1	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	1	0
	5	0	0	1	0	0	0	0	0	0	0
	6	0	0	0	0	1	0	0	0	0	0
	7	0	0	0	0	0	1	0	0	0	0
	8	0	0	0	0	0	1	1	0	0	0
	9	0	0	0	0	1	0	0	0	0	0
	10	0	0	0	1	0	0	0	1	1	0

6.3 Network design performance for k≥4

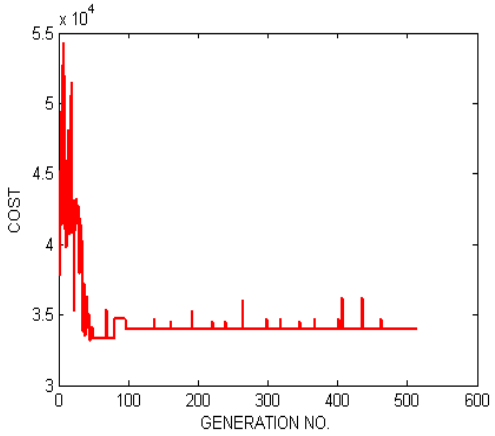

Fig.6(a)

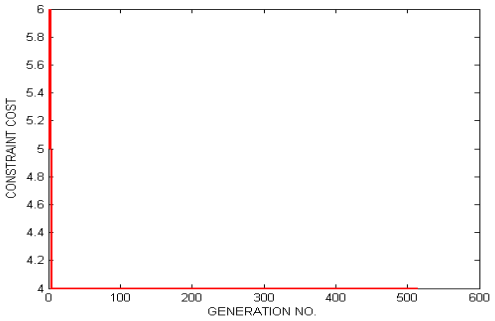

Fig.6(b)

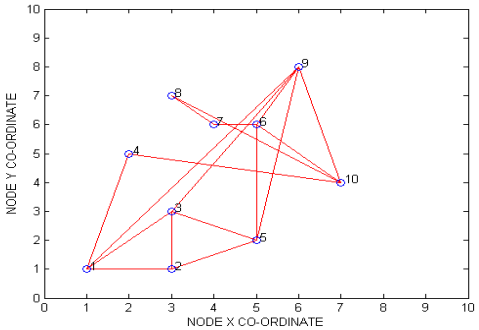

Fig.6(c)

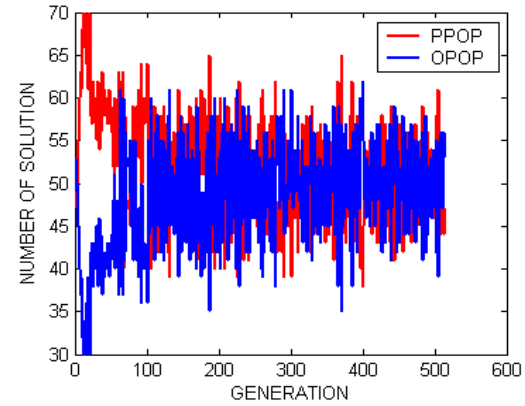

Fig.6(d)

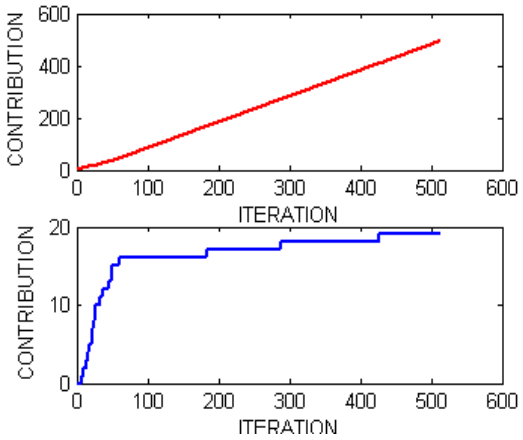

Fig.6(e)

TABLE 4
Generated architecture connection

FROM \ TO	N	1	2	3	4	5	6	7	8	9	10
	1	0	1	1	1	0	0	0	0	1	0
	2	0	0	0	0	1	0	0	0	0	0
	3	0	1	0	0	0	0	0	0	0	0
	4	0	0	0	0	0	0	0	0	0	1
	5	0	0	1	0	0	0	0	0	0	0
	6	0	0	0	0	1	0	0	0	0	0
	7	0	0	0	0	0	1	0	0	0	0
	8	0	0	0	0	0	0	1	0	0	0
	9	0	0	1	0	1	1	0	0	0	0
	10	0	0	0	0	0	1	0	1	1	0

7 APPENDIX

TABLE 1

CONNECTION AND COST DEFINITION MATRIX

FROM \ TO	N	1	2	3	4	5	6	7	8	9	10
	1	-	23.9	24.6	11.5	∞	∞	∞	∞	21.6	∞
	2	23.9	-	9.9	∞	17.1	∞	∞	∞	∞	∞
	3	24.6	9.9	-	∞	7.5	∞	∞	∞	9.4	∞
	4	11.5	∞	∞	-	∞	∞	∞	∞	15.6	14.4
	5	∞	17.1	7.5	∞	-	15.9	∞	∞	9.3	∞
	6	∞	∞	∞	∞	15.9	-	14.5	20.8	19.6	24.3
	7	∞	∞	∞	∞	∞	14.5	-	10.2	∞	∞
	8	∞	∞	∞	∞	∞	20.8	10.2	-	∞	14.0
	9	21.6	∞	9.4	15.6	9.3	19.6	∞	∞	-	17.7
	10	∞	∞	∞	14.4	∞	24.3	∞	14.0	17.7	-

TABLE 2
TRAFFIC DEMAND MATRIX

FROM \ TO	N	1	2	3	4	5	6	7	8	9	10
	1	-	56	124	79	88	71	147	78	136	175
	2	165	-	199	67	191	130	85	55	74	79
	3	159	178	-	135	71	159	153	161	140	146
	4	148	102	130	-	129	110	150	131	100	151
	5	164	120	78	54	-	104	71	92	149	166
	6	150	188	126	181	192	-	54	106	180	107
	7	183	85	114	55	101	181	-	52	136	117
	8	91	180	150	128	116	144	68	-	198	123
	9	113	149	152	79	121	87	61	108	-	142
	10	82	184	194	158	73	197	178	89	73	-

7. CONCLUSION

The focus of this paper is to design an automatic solution for a network architecture which could meet the requirements of connectivity for reduction in congestion. Simply increasing the connectivity will make the network unacceptable from cost and traffic perspectives. Hence a unified approach is applied in this work to take care of all these parameters. To develop the network design, GA algorithm is applied which takes care of three very important issues, which heavily affect the design characteristics-(i) connectivity to improve the network congestion performance (ii) meet the demand of network traffic in an adaptive manner (iii) minimize the cost. A suitable genetic model, using a novel crossover operator, an efficient fitness evaluator and tournament selector is presented to design the network according to the given connectivity constraint, traffic requirement and to minimize the cost. The proposed method was successfully applied for varying connectivity constraints and optimal results were obtained. Also it was clearly seen that the contribution of the parents towards the optimal solution was very much greater than offspring's in accordance with nature where the maturity and experience of parents most often leads to better solutions. With the proposed solution, the concept of automated design in the field of network research will move one step forward.

ACKNOWLEDGEMENTS

S.V.Uma expresses sincere thanks to Dr.P.V.Rao, HOD, Department of ECE, Rajiv Gandhi Institute of technology, Bangalore for his valuable timely suggestions and help.

REFERENCES

[1] M.G.H. Bell and Y. Iida, Transportation Network Analysis. John Wiley & Sons, New York, 1997.

[2] S. Peeta and C. Zhou. Robustness of the off-line a-priori stochastic dynamic traffic assignment solution for on-line operations. Transportation Research, Part C, 7(5):281–303, 1999.

[3] Mostafa Abd-El-Barr', Ahmer Zakir**, Sadiq M. Sait', and Abdulaziz Almulhem, Reliability and Fault Tolerance based Topological Optimization of Computer Networks - Part 11: Iterative techniques, IEEE Communications magazine, 2003

[4] Mitsuo GEN, Kenichi IDA & Jongryul KIM, A Spanning Tree-Based Genetic Algorithm for Bicriteria Topological Network Design, IEEE Communications magazine, 1998

[5] Sun-Jin Kim and Munkee Choi, A Genetic Algorithm for Server Location and Storage Allocation in Multimedia-on-Demand Network, IEEE, 2003

[6] Anton Riedl, A Versatile Genetic Algorithm for Network Planning, Eunice 1998

[7] Fan Li Yu Wang and Xiang Yang Li, Gateway Placement for Throughput Optimization in Wireless Mesh Networks, IEEE Computer Society

[8] Giuseppe Calafiore and Laurent El Ghaou, Robust Dynamic Traffic Assignment under Demand and Capacity Uncertainty, IEEE

[9] Chun-Yen Hsu*, Jean-Lien C. Wu+, Shun-Te Wang+ and Chi-Yao Hong, A Time-Efficient Algorithm for Optimal Design of Backbone Wireless Mesh Networks, IEEE, 2006

[10] Salah Al-Sharhan, Fakhri Karray, and Wail Gueaieb, Learning-Based Resource Optimization in Asynchronous Transfer Mode (ATM) Networks, IEEE transactions on systems, man, and cybernetics—part b: cybernetics, vol. 33, no. 1, pp 122 - 132, February 2003

[11] Cem Ersoy, Albert Levi, and Okan Gumrah, Artificial Intelligence Search Techniques for Discrete Link Capacity Assignment in Prioritized Multiservice Networks,

[12] Ka-Cheong Leung and Victor O. K. Li, Flow Assignment and Packet Scheduling for Multipath Routing, KICS journal of communications and networks, vol. 5, no. 3, September 2003, pp 230 to 239

[13] Sancho Salcedo-Sanz and Xin Yao, A Hybrid Hopfield Network-Genetic Algorithm Approach for the Terminal Assignment Problem, IEEE transactions on systems, man, and cybernetics—Part b: Cybernetics, vol. 34, n0. 6, pp 2343 to 2353, December 2004

[14] Hongfeng Xiao and Guanzheng Tan, A Novel Simplex Hybrid Genetic Algorithm, The 9th International Conference for Young Computer Scientists, 2008, IEEE Computer Society, pp 1801 to 1806.