International Journal of Soft Computing 2 (3): 471-481, 2007

Improved Genetic Algorithm for Dynamic QoS Routing to Multiple Destinations for Multimedia Applications (IGADQoS)

[1]K. Vijayalakshmi and [2]S. Radhakrishnan
[1]Department of IT, Mepco Schlenk Engineering College, Sivakasi, Tamilnadu, India
[2]Department of HOD/CSE, Kalasalingam University, Krishnankoil, Tamilnadu, India

Abstract: This study presents a new genetic method for solving dynamic multicast routing problem, which is found in multimedia applications. Multicast services in multimedia applications require the optimization of QoS parameters namely cost, end-to-end delay and each link must meet delay and bandwidth constraints. This study proposes an improved genetic algorithm for the construction of QoS multicast tree which has the following features: Multicast tree adopts for dynamic changes; all the links in the tree must meet delay constraint and bandwidth constraint; cost and end-to-end delay is better than other heuristic algorithms; the encoding method helps to perform dynamism; improved genetic operators and heuristic local search operation; Multicast routing over unicast. We have also performed a comparative study of selection mechanisms in GA using simulation and listed the best one for our problem. Experimental results show that our improved genetic algorithm has higher search success rate, convergence rate, dynamic request success rate and least cost than existing methods.

Key words: Dynamism, multicast QoS routing, improve genetic algorithm, elitism, QoS parameters, m point crossover

INTRODUCTION

Need for dynamic QoS routing: Multicast service is a key requirement for computer networks supporting multimedia applications. Most of the multimedia applications such as video conferencing, distance learning, concurrent editing system, multi-person communication etc require transmission of message from single source to multiple destinations with strict QoS parameters. Finding such a multicast tree with optimizing of one parameter (cost) is NP complete problem (Striegal and Manimaran, 2002) (Wang and Hou, 2000). In real time multimedia communication, a connection from source to multiple destinations needs to be found before any data transmission occurs.

Multimedia applications require optimization of cost and end-to-end delay simultaneously (Kompella *et al*., 1993). There are 2 steps in real time multimedia multicasting: Routing and Dynamic changes. The routing is to find the routing tree which is rooted from single source to multiple destinations under the required QoS constraints. And that tree must have a minimum cost and end-to-end delay (Kompella *et al*., 1993). In addition to minimum cost and end-to-end delay, the links present in the multicast tree, which is used, for multimedia real time services must meet bandwidth, delay bound constraints. But finding a multicast tree with optimization of two parameters: Cost and end-to-end delay is NP hard. The dynamic change is nothing but both the group membership change and node/link failure. Most of the multimedia applications require dynamic membership change in multicast group that is a destination node may leave or join the multicast group with out affecting the existing traffic in the connection (with out restructuring the existing tree and needs only small modification in that tree).

Related works and objectives: In this study we have given a brief review of several heuristic algorithms which is available for construction of multicast tree. This review includes both heuristic algorithms and stochastic algorithms proposed for multicasting. Also these algorithms are classified into various categories: Static, dynamic, constrained,without constrained, single solution and Multiple solution algorithms. Brief review of static and dynamic algorithm used for multicasting is given below. Various conventional heuristic algorithms were proposed by many researchers (Charikar *et al*., 2004; Jia *et al*., 1997; Kuipers and Van, 2002). They also proposed several traditional heuristic dynamic algorithms for handling the group membership (Debasissh *et al*., 2003; Imase and Waxman, 1991; Naryaez *et al*., 2000). These

Corresponding Author: K. Vijayalakshmi, Department of IT, Mepco, Schlenk Engineering College, Sivakasi, Tamilnadu, India

dynamic algorithms produced the least cost multicast tree. Some times these dynamic and heuristic algorithms produced good results: However, these are having the following weaknesses: Average execution time grows exponentially when size of network grows, no guarantee for optimal convergence solution, sometimes lead to local optimum and give optimum solution to the problem only for sparse networks. Mostly these algorithms searched the entire network in sequential manner (try all the paths) and not in stochastic manner (Gelenbe *et al*., 1997; Imase and Waxman, 1991; Jia *et al*., 1997). This takes more time for finding optimal results. So these algorithms are mostly efficient in small sized networks. Researchers used neural networks to improve the quality of multicast tree (Chotipat *et al*., 1995; Gelenbe *et al*., 1997). But the method proposed by them is too complex for solving the multicast routing. Neural network approaches are suitable for small sized networks. All iterative polynomial time algorithms did not lead to optimal solution. Stochastic heuristic algorithms are used for solving this type of NP problem.

Researches have proposed several heuristic algorithms for finding the least cost multicast tree under various constraints (Chotipat *et al*., 1998; Lee and Attiquazzaman, 2005). GA is a guided random search stochastic technique to solve large-scale optimization problems and combinatorial problems (Bui and Moon, 1996; Davis, 1991). GA has been used for solving various NP complete problems viz Network capacity assignment (Atzori and Raccis, 2002), Cellular Call Admission (Rose and Yener, 1997), Multicast routing (Shi *et al*., 2000), QoS routing (Shimamoto *et al*., 1993) etc. All researchers were concentrating on construction of static least cost multicast tree, which satisfy delay constraints (Barolli *et al*., 2003; Haghighat *et al*., 2004). A bandwidth delay constrained least cost multicast routing based heuristic genetic algorithm was proposed by Zhengying *et al*. (2001). They have used the penalty function to penalize the individual based on delay tolerance. Debasissh *et al*. (2003) have proposed a near optimal dynamic, delay, jitter, bandwidth constrained non-stochastic algorithm without concentrating on the end-to-end delay. Sun and Li (2004) have proposed a static least cost multiple constrained multicast routing algorithm based on genetic algorithm. His encoding method can not be able to adapt the dynamic changes. Leung *et al.* (1998) have proposed a genetic algorithm approach for sparse and dense networks with out constraints. But they also produced the static tree only. Bao *et al*. (2006) have also proposed a genetic approach for static least cost routing.

All the previous algorithms construct the least cost multicast tree under various constraints. That is these algorithms optimize only one parameter. But most of the real time multimedia applications require optimization of residual cost and end-to-end delay. Dynamism and single scalar optimized solution is required for these applications. And these two parameters (cost and end-to-end delay) are consistent with each other (minimization of both parameters). So we propose a novel protocol for developing a single solution that satisfies all QoS constraints using improved GA.

Genetic Algorithms (GA) are adaptive search techniques that derive the models from the genetic processes of biological organisms based on evolution theory. GA provides a robust and powerful adaptive search mechanism. The most important advantage of GA is that they use only the pay off (objective function) and hence independent of nature of the search space such as smoothness, convexity or unimodality. For increasing the convergence of GA, heuristic local search functions are embedded into Simple GA (SGA). Modified GA operators are also used for this purpose.

Improved genetic algorithm is used to incorporate the domain specific knowledge. That is some modification is added to simple GA in order to take into account the discrete nature of multicast search space. Three ways to incorporate the domain specific knowledge are suggested in (Davis, 1991): Local search, use of special genetic operators and problem specific encoding. These methods were followed in various applications and successful performance has been obtained (Jog *et al*., 1989; Zheng *et al*., 1997).

This study proposes an improved genetic algorithm for the construction of bandwidth constrained multicast tree which optimizes cost and end-to-end delay simultaneously. Each link in the multicast tree must satisfy the delay constraint Δ. We have embedded the new local heuristic function into the SGA. Our proposed method utilizes the m point crossover operator for quick convergence.

Objective of our research is to construct the least cost and minimum end-to-end delay multicast tree.

This algorithm has following features:

- Dynamic Change: Group Membership change: A Destination can leave or enter into the multicast group with out restructuring the multicast tree (Not affecting the existing traffic). It also updates the node or link failure.
- Improved Genetic Operators: Modified ‘m’ point crossover, Tournament selection with size is 2 and mutation with heuristic algorithm for final replacement.
- GA based Routing table: Bandwidth and delay constraint routes are found using GA and stored in

the routing table on fitness sorted order. The routes in routing table are cost and end-to-end delay optimized routes.

- Chromosome representation: Number of bits in the chromosome representation depends on number of destined nodes. Routes in the routing table are bandwidth and delay constrained optimized routes.
- Initial Population: Random Initial Population method is used and the entire chromosome in the initial stage must different from each other.
- GA is used for routing table and Multicast tree construction.
- Elitist model: For avoiding the removal of best feasible solution at every generation.
- Geno-Pheno Algorithm: Gene to Tree Conversion.
- Local heuristic function: Local heuristic function is used to get the best chromosomes for next generation.

PROBLEM SPECIFICATION

The network can be represented as the undirected and connected graph G(V,E) where V is set of network nodes, E is set links (edges) and n = |V| be the number of nodes in G. A link $e \in E$ connecting nodes v_1 and v_2 will be denoted by (v_1, v_2). Each edge is associated with edge cost C_{ij}, delay D_{ij} and bandwidth B_{ij} where $i, j \in V$. Delay includes transmission, propagation and queuing delay and edge cost could be a measure of buffer space or monetary cost. A non- empty set $U = \{s, v_1, v_2, v_{3,..} v_k\}$ in V is called the multicast tree (X), where $s \in V$ is the source node, $T = \{v_1, v_2, v_3, \ldots v_k\}$ is the set of destination nodes and k is the number of destination nodes. The Multicast tree X = (s, T) is a tree rooted at s and routes information to all members in T. P (s, v_i) is the unique path in a tree X(s, T) from the source node to a destination node v_i, $v_i \in T$. The goal is to find a multicast tree between a single source and set of destinations, which will simultaneously optimize the monitory cost and end-to-end delay.

Cost of the multicast tree X is the sum of the cost of all links in that tree and can be given as follows:

$$C_t(x)=\sum_{i,j\in X} C_{ij}$$

End-to-end delay of the multicast tree X is the sum of the delay of all links in that tree and can be given as follows:

$$D_y(x)=\sum_{i,j\in X} D_{ij}$$

Bandwidth of the path P(s,d) is defined as the minimum bandwidth at any link along the path is B(P(s, d)) = minimum ($B_{i,j}$, where link i to j is in P(s, d)).

Our algorithm has two objectives: Minimization of cost (Ct(X)) and minimization of end-to-end delay (Dy(X)). Aim of our work is to find least cost, minimum end-to-end delay multicast tree which is subjected to bandwidth constraints. Problem is defined as follows:

$$\text{Min } (Ct(X), Dy(X))$$

Subject to

$$B\,(P(s,d)) = B_d,\ \forall d \in T$$

IGADQoS contains two phases

- Construction QoS routing table.
- Construction of multicast tree.

In both phases, the basic idea is to optimize cost, end-to-end delay with degree and bandwidth constraints using GA. At run time, use dynamic algorithm for adapting the dynamic changes.

GENETIC ALGORITHM IN DYNAMIC QOS ROUTING

There are two steps in our algorithm: Routing and Dynamic changes. Routing consists of two stages: One is construction of optimized routing table using improved GA and second is the construction of QoS multicast tree using routes in routing table and improved GA.

Description of genetic algorithm: The genetic algorithms are part of the evolutionary algorithms family, which are computational models, inspired in the Nature. Genetic algorithms are powerful stochastic search algorithms based on the mechanism of natural selection and natural genetics (Davis, 1991). Genetic algorithms are able to evolve solutions to real world problems. It works with populations (chromosomes) of individuals, each representing a possible solution to a given problem. Each individual is evaluated to give some measure of its fitness to the problem from the objective functions. GA works with a population of binary string, searching many peaks in parallel. By employing genetic operators, they exchange information between the peaks, hence reducing the possibility of ending at a local optimum. GAs are more flexible than most search methods because they require only information concerning the quality of the solution produced by each parameter set (objective function values) and not like many optimization methods which require derivative information, or worse yet, complete knowledge of the problem structure and parameters.

Genetic operators used: The detailed description of genetic operators used in our proposed algorithm is given below:

Selection: Reproduction operator is used to improve the quality of the population by selecting the high-quality (fittest) individuals, which is copied to next generations. It is an operator that obtains a fixed number of copies of solutions according to their fitness value. If the score increases, then the number of copies increases too. A score value is of associated to a given solution according to its distance from the optimal solution (closer distances to the optimal solution mean higher scores). In this study we have taken into consideration the Truncation selection, roulette wheel selection and tournament selection for simulation. We have analyzed the results and finally use the pair wise tournament selection without replacement is used in the proposed method. In tournament selection method, n (n is equal to tournament size) individuals are selected randomly and the best one among n is entered into new generation, which serve as a parent to next generation. The same individual should not be selected twice as a parent. This process is repeated for Population Size (PS). Convergence of GA's may be fast when we increase the tournament size that leads to increase in the probability of selecting the wrong individuals exponentially. So we have selected the tournament size as 2.

Crossover, mutation and replacement: Using crossover operator, information between two chromosomes are exchanged which mimic the mating process. An m-point cross over operator with crossover probability Pc (0.52) is used. A pair of high fitted parents selected from the population randomly and cross over operator chooses 'm' cutting points randomly and alternatively copies each segment out of two parents. This operation is depicted in Fig. 1. Value of 'm' is greater than 2 but less than q-1, where 'q' is number of bits in the chromosome. This crossover may result in offspring that break the subset size requirements, because the exchanged gene segments may have different number of occurrences of 1.

Mutation is applied to offspring after cross over. Mutation involves switching a single bit in a random position in a string and thus introducing fresh genetic material. Mutation operator changes 1 to 0 and vice versa with small probability P_m (0.1) which indicates the frequency at which mutation occurs. The mutation operator introduces new genetic structures in the population by randomly modifying some of the genes, helping the search algorithm to escape from local loop. It replaces inferior parents and similar parents. Figure 2 shows the mutation at the 9th bit position. If the mutated chromosome is superior to both the parents, it replaces the similar parent; if it is between the two parents, it replaces the inferior parent; otherwise, the most inferior chromosome in the population is replaced. GA stops when the number of generations reaches the preset maximum generations Ns.

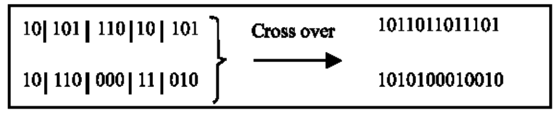

Fig. 1: '4' Point Cross Over

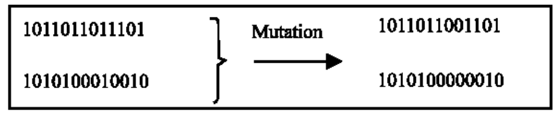

Fig. 2: Mutation operation

Simple GA (SGA): Simple Genetic algorithm is nothing but it has simple GA operators with out any heuristic search function. SGA is depicted as follows:

```
Generational_SGA( )
{
Initialize population P
{
for(i=1 to Ns)
{
    Calculate fitness values.
    Select 2 parents p1 and p2 from P
    Offspring i = single point cross over(p1, p2);
    Mutation(offspring i);
}
Replace P with offspring1, ... offspring n
}
```

Routing table construction (QoS routes): QoS unicast routing is to find the optimal path from source node to destination node, subject to many constraints and optimized resources and is NP complete (Cheng and Ramakrishnan, 2002; Korkmaz and Krunz, 2001). A GA based constraint QoS routing method is used to find the optimal routes and these are placed in sorted order based on fitness value in routing table. Variable length chromosomes representation has been used (Chang and Ramakrishnan, 2002). Each locus in the chromosome represents the node that included in a path between source and destination. First locus in the chromosome is always source node and last locus number is for destination. Maximum number of loci in the chromosome should not exceed the number of nodes in the network. During the construction of routing table, we have not considered the edge whose D_{ij} is greater than constant delay Δ and whose bandwidth bij is greater than B. This procedure helps the algorithm for not selecting the edge with unacceptable (maximum) delay and minimum cost. We have used the following parameters for the construction of routing table using GA.

Population Initialization: Depth first search technique is used for population initialization. Repeated chromosomes are removed in the initialization phase. So all the chromosomes after the initialization are different with each other, which helps to search the space quickly.

Fitness function: Weighted sum method is used for the minimization of cost and end to end delay under the constraints of bandwidth. Here equal weightage is given to weights associated with fitness function.

Chromosome representation (Multicast tree): For a given source node and set of destination nodes $\{v_1, v_2, v_3,.., v_k\}$, individual solution (chromosome) can be represented by a bit string of length k. Each bit in the chromosome corresponds to path in the routing table from source to the specific vertex. For a given multicast routing problem, Chromosome representation is shown in Fig. 3.

Geno_ pheno function: This function returns the correct row entry number from the routing table. If Gene Gj = 1, then Geno_Pheno function gives the integer value in {0, 1, 2, …, R-1} which is the routing table's ($s \rightarrow v_j$) route number. From the routing table the path route from s to v_j is taken and have constructed the multicast tree. This coding method is first proposed by Shimamoto *et al.* (1993) for the point-to-point routing problem. Major advantage of the method is that chromosome representation requires only k bits (k = number of destinations nodes). This method has the ability to solve routing problem very quickly even for large size networks. For example Gene: 100101 Geno_Pheno Algorithm gives the correct row entry numbers of routing tables $s \rightarrow v_4$, $s \rightarrow v_1$ and $s \rightarrow v_6$ The corresponding path route is taken from routing table $s \rightarrow v_1$, $s \rightarrow v_4$ and $s \rightarrow v_6$ and constructs the multicast tree.

Initial population (Multicast tree): The straightforward initial population is shown below. Repair function is used

G_1	G_2	...	G_j	...	G_k

Fig. 3: Chromosome representation

at each bit to avoid the loop formation and to avoid duplication. We have selected the path routes among the rows 0 to R/3 rather than among rows 0 to R from the routing table. This operation gives better initial chromosomes and saves time since routing table is already in sorted order based on fitness function. Acceptable cost is the average cost in the given input cost matrix. Δ is the acceptable delay constraint for a link. If the chromosome has unacceptable cost and end-to-end delay then discard the chromosome during initialization phase itself. Due to above two steps, diversity increases and gives optimized results (increasing convergence rate). Pseudo code for Population initialization is described as follows:

```
Initial Population ()
 { n = 1;
   While(n<Ps)
    {
     for (each gene g in i^th chromosome)
         {k = rand_number ();
      if (feasible(k)) then g=k
      else
             g = 1;
         }
 If (Check _ feasibility (Tree)) {n + +; continue;}
 if ((Acceptable cost* number of 1s in the chromosome)
>= Σ C_ij ) and (Acceptable delay (Δ)* number of 1s in the
                 chromosome) > = Σ D_ij))
n = n-1; where i, j∈X
```

Fitness function (multicast tree):

Objective function is given as follows:

$$f(x) = (w_1 \frac{1}{C_t(x)} + w_2 \frac{1}{D_y(x)}) * A \qquad (1)$$
$$A = (\prod_{d \in T} \varphi(B(P(s,d) - B_d)))$$

$$\{\text{Where}\ \varphi(h) = \begin{Bmatrix} 1 h > 0 \\ \lambda h <= 0 \end{Bmatrix}$$

w_1 and w_2 are weight factors and these values are
if ($D_y(X) < \Delta$) then
$w_1 = 0.5$ and $w_2 = 0.5$
else
$w_1 = 0.75$ and $w_2 = 0.25*(D_y(X) - \Delta)$

ϕ (h) is the penalty functions (Ozgur, 2005) and λ is the penalty factor. In our protocol λ value is taken as 0.56. Penalty functions are used to handle the constraints (Ozgur, 2005). These methods transform the constrained problem to unconstrained problem and these are used to penalize the individuals based on their constraint violation. The penalty imposed on infeasible individuals can range from completely rejecting the individual to decreasing its fitness based on the degree of violation.

Heuristic local search function in multicast tree: This function takes three chromosomes as parent and then calculates the cost and delay for the two chromosomes from source node to all destinations. Chromosome CT_1, CT_2 and CT_3 are calculated as follows:
$CT_1 = \{p_1 (s, d_1), p_1 (s,d_2) \ldots. P_1 (s, d_n)\}$; $CT_2 = \{p_2 (s, d_1), p_2 (s, d_2) \ldots. P_2 (s, d_n)\}$;
$CT_3 = \{p_3 (s, d_1), p_3 (s, d_2) \ldots. P_3 (s, d_n)\}$

The sub-fitness value of the path is calculated as follows in the Eq. (2).

$$P_i (s, d_j) = W_1{}^* C (s, d_j) + W_2{}^* D (s, d_j) \qquad (2)$$

W_1 and W_2 are given the same values.

Next, this function compares the sub-fitness value in the same destination for the above three chromosomes. Higher fitness value of sub-fitness for every destination is added (taken) and creates the new offspring. Best three among the four is selected for next generation.
For example

$CT_1 = \{3\rightarrow4,\ 3\rightarrow5\rightarrow6\rightarrow2,\ 3\rightarrow5\rightarrow7\}$, $CT_2 = \{3\rightarrow8\rightarrow4,\ 3\rightarrow15\rightarrow2,\ 3\rightarrow1\rightarrow5\rightarrow7\}$,
$CT_3 = \{3\rightarrow1\rightarrow4,\ 3\rightarrow6\rightarrow2,\ 3\rightarrow6\rightarrow5\rightarrow7\}$

The sub-fitness value for the destination 4 in CT_1 is less than CT_2 and CT_3, so we take $3\rightarrow4$ the sub-fitness value for new offspring. Sub-fitness value of $3\rightarrow15\rightarrow2$ in CT_2 is better than other two. Similarly the sub-fitness value of $3\rightarrow5\rightarrow7$ gives better value than other two sub-fitness values. So this function generated a new offspring as $\{3\rightarrow4,\ 3\rightarrow15\rightarrow2,\ 3\rightarrow5\rightarrow7\}$. Best three chromosomes among four chromosomes are selected for next generation.

Improved genetic algorithm for dynamic QoS routing: The main purpose of our algorithm is to find dynamic Qos routing to multiple destinations (IGADQoS), which simultaneously optimize cost and end-to-end delay under the constraints of bandwidth and delay.

Multicast Tree Construction Algorithm (MTC)

Step1: Input () -- Number of nodes, C_{ij}, D_{ij}, B_{ij}, destination nodes $\{v_1, v2, v3,.,., v_k\}$, $\triangle$, B_d, number of generation Ns and source s.

Step 2: Initialization of Parameters: Pm=0.1, Pc=0.52, Ns =150.

Step 3: Routing table is constructed as in previous study.

Step 4: Chromosomes for multicast tree construction are initialized according to the procedure in previous study
For (i = 1 to Ns)
{

Step 5: Evaluate the fitness value of all chromosomes using the Eq. 1.

Step 6: Genetic operation

- Use tournament selection method for selecting the individual for next generation.
- Elitism – Elitism can be used for keeping the best feasible solution in the population. In each generation, 5% of best individuals are transferred from one generation to next with out modification. We have avoided the removal of highfitted individuals during GA operations.
- Crossover –Perform 'm' point cross over operation with a probability of P_c.
- Heuristic local search function- Select high fitted three individuals and create a new individual from three using local search method. Then selects best three among four(high fitted three individuals and newly created one) and adds to next generation.
- Generate new chromosomes based on mutation probability P_m.
- Heuristic Local search function: Perform local search function at a probability of 0.25.

Step 7: Repair Function: Used to avoid the infeasible chromosomes which is formed during cross over and mutation. Also this function avoids the formation of loop.

Step 8: Check_feasible function-In each generation, we check the feasibility of chromosome using feasible function. After genetic operation, there is very less chance for occurrence of the chromosome, which is best in one parameter (cost) and worst in another parameter (end-to-end delay). This is avoided by comparing acceptable cost and delay values with cost and delay of the chromosomes.
}

Step 9: Use Geno_pheno algorithm and Routing table to construct the multicast tree. Above nine steps give the optimized static multicast tree. The dynamism of our algorithm is described as follows:

Dynamic algorithm (): In case of real time multimedia application the following situation are occurred

- Dynamic change in the multicast group.
- Destined node failure/Link failure.

For the given input, call the MTC algorithm to construct the multicast tree. Then our dynamic algorithm always solves the dynamic situation (destination group change at run time or node/link failure) with out reconstruction of tree. Our protocol automatically updates the information in O (1) time to O (n) time. In case of real time application there is a chance for the occurrence of following three situations: situation one is either any new destination v wants to enter into destination group or any already participated destined node wants to leave from the multicast group (X). Second situation is either destined node failure or intermediate node failure. The third situation is either connection failure between any two nodes or connection failure between any destination node and intermediate node.

In the first situation, if any new node v enters into the multicast group then checks whether v is already part of the multicast tree for transmitting messages (not for user, only mediator). If it is so, then initiate v to participate in multicasting and it is allowed to send information to its users. Otherwise find the shortest path from any node in X to v from routing table and add those links into the multicast tree such that loop is not formed. In case of any destined node leaves from the multicast group, then remove the connection between the destined node and its parent by assigning NULL to the corresponding child field.

In the second situation, if the destination node is failed, then remove the node from X and broadcast that information to all vertices (to run the routing table algorithm). If the intermediate node v_n is failed then find the next optimal route from v_n's parent (proper ancestor) to v_n's child (proper descendent) and connect it to X after removing the old link.

In the third situation, connection failure (link failure) between any destination node and intermediate node is replaced by finding the best route among the following two routes: one is next optimal route between destination node and intermediate node, second is optimal route between intermediate's parent and destination node. That particular route is added to multicast tree. Similarly connection failure between any two intermediate nodes is handled and that route is added to multicast tree. If that alternate route is not optimal then it runs the MTC algorithm. If the failed link is not present in X, there is no change in the multicast group but that information is

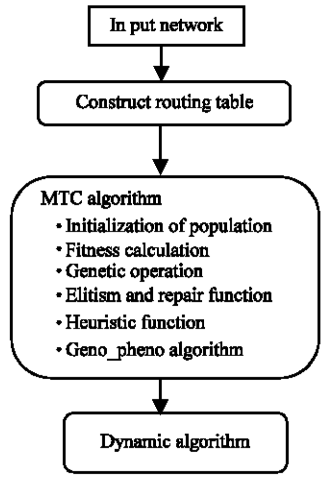

Fig. 4: IGADQoS algorithm

broadcast to all routers that help the associated nodes to run their routing algorithm.

IGADQoS easily finds the alternate optimized path and updates the information in O (1) time to O (n) time, since our protocol takes the routes from routing table, which was constructed during first stage of our algorithm. And this table always contains the optimized paths. Flow of our complete algorithm is depicted in Fig. 4.

Time complexity: During the first time construction, the time complexity is O (PS* generation* nklogn). Dynamic group change in the multicast group needs only constant time. Link/node failures requires O (n) at the worst case.

SIMULATION RESULTS

In this study we have used the simulation experiments to compare the proposed algorithm and other existing algorithms. Simulation was done on Pentium IV 512 MB PC in C + +. The following parameters have been used for this study. Size or Routing table = R/3; Number of generation = 150; Tournament size = 2; Group size = 35% of Network size.

Population size = 33; Crossover Probability = 0.52; Mutation Probability = 0.1.

Delay associated with link is randomly chosen with in range (0. 55 m), the cost of each link is generated within range (0. 350), bandwidth is within range (50-100 kbps) and Δ is 45 m. We have randomly selected the network topology to perform the simulation. For simulation random network of size 20 to large sized network of up to 450 was generated. Performance of our protocol is compared with NGA (Bao *et al.*, 2006), GAMDR (Leung *et al.*, 1998), NDA (Naryaez *et al.*, 2000), SGA (Vijyalakshmi and

Radhakrishnan, 2005) (our previous algorithm) heuristics. Our performance metric measures include convergence rate, cost, end-to-end delay, search success rate and dynamic request success rate. In all the simulation experiments, the average connection degree of the node considered is 6.

Figure 5 and 6 show the convergence rate of our protocol. Cantu (2000) suggests that the number of fitness function evaluations directly measure the excellence of convergence only if all GAs converge to identical of quality solutions. In Fig. 5 it is observed that the proposed algorithm exhibits fastest convergence rate than other algorithms for various group sizes. Other algorithms require more number of fitness function evaluations than our method for obtaining the similar solution. IGADQoS quickly converges to optimal solution than existing algorithms. Increasing the number of generation leads to better fitness values. From Fig. 6 it is observed that our proposed algorithm gives optimal values and did not converge to same constant value prematurely that is with in few generations. After 60th generation our protocol gives optimal converged solution. Because Local heuristic function increases the convergence rate. For Fig. 6 the network size is 200 (Group size is 70).

Figure 7 shows the comparison of tree cost performance of our algorithm with existing algorithms. As shown in Fig. 6 and. 7, IGADQoS produced better tree cost performance for large sized networks also. For Fig. 7 and. 8, multicast group size is 35% of network size. As shown in Fig. 8, IGADQoS has produced better end-to-end delay than SGA. This is because many routes which

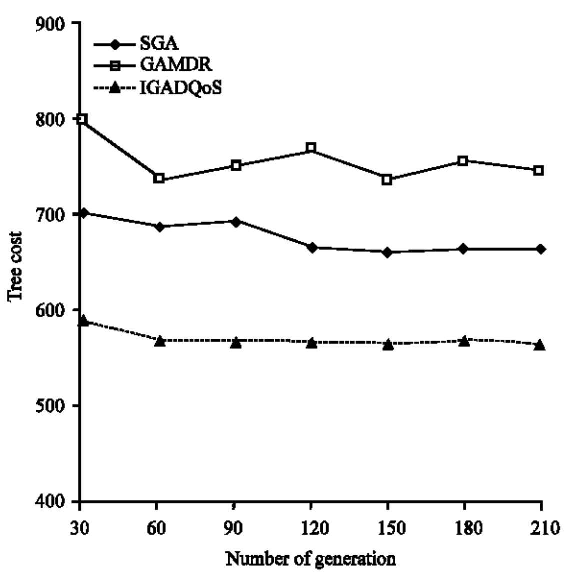

Fig. 5: Convergence rate in terms of Fitness function

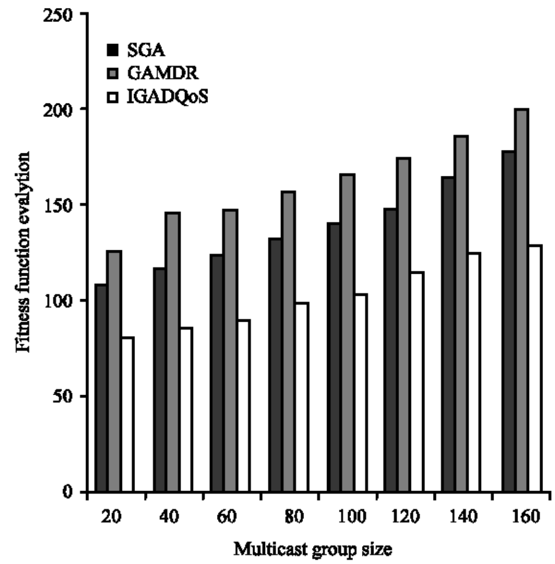

Fig. 6: Convergence rate in terms of Tree cost evaluations

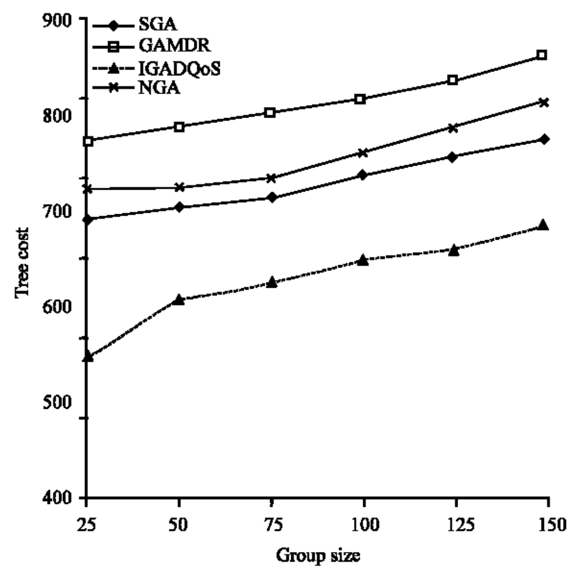

Fig. 7: Comparative analysis of Tree cost

satisfy the required delay constraint exist in the routing table and that optimized routes are used for multicast tree construction.

Search success rate is nothing but how often an algorithm could find a route that can satisfy all the QoS constraints. Figure 9 shows the search success rate of our method for different multicast group sizes. Our algorithm produced an average search success rate of 99.5% for both small sized and large sized networks. Next we have analyzed the dynamicity of our protocol by measuring the dynamic request success rate. Dynamic request success rate (D_{req}) of our protocol for different delay constraints is

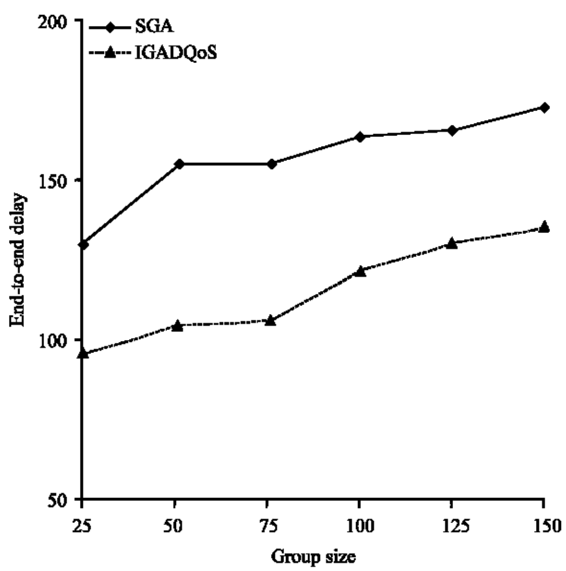

Fig. 8: End-to-end Delay versus group size

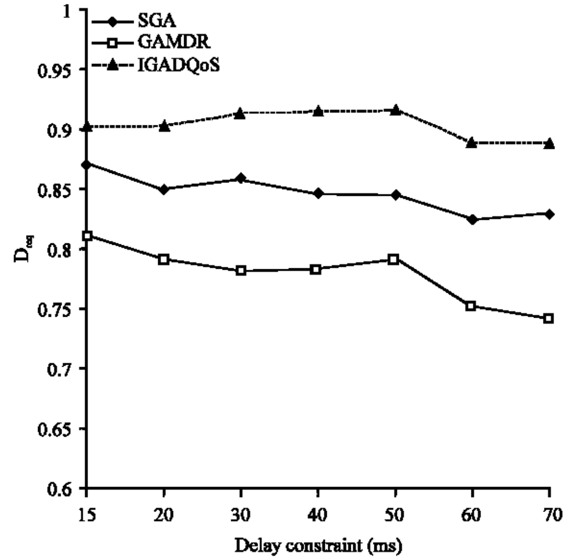

Fig. 10: D_{req} Versus delay constraint

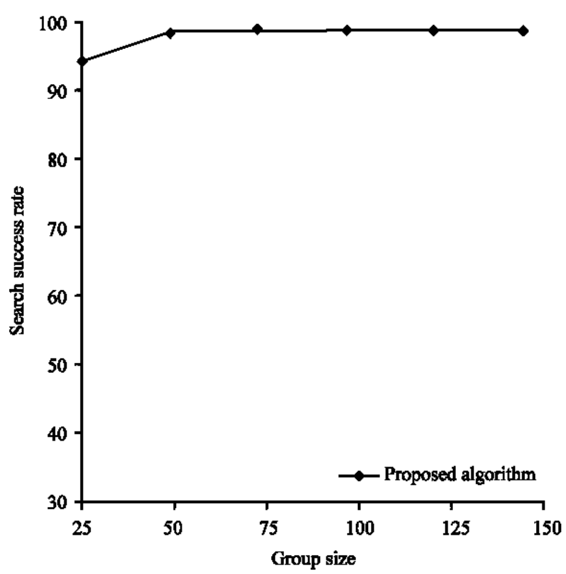

Fig. 9: Search success rate versus multicast group size

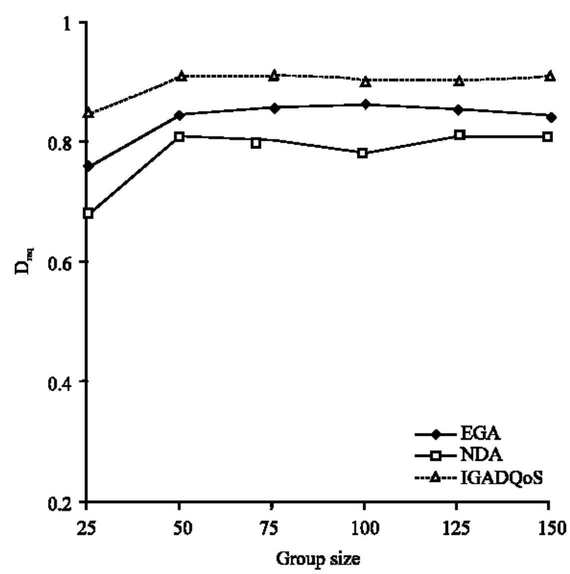

Fig. 11: Dynamic property of our algorithm

shown in the Fig. 10 Dynamic request Success rate is the ratio of total number of dynamic requests success to the total number of dynamic requests. In Fig. 10 the multicast group size considered is 60 and bandwidth constraint is 80 Kbps. When the delay constraint (Δ) is around 50 m then the D_{req} is about 91%

Figure 11 shows the dynamic property of our protocol for different sized networks. In Fig. 11 the degree constraint is 45 m. It was observed that 91% of dynamic request success rate is achieved. This is because it uses the routes from already optimized routing table, which was created during the first stage of our protocol. Since improved GA is used in both unicast and multicast many routes which satisfy the required delay constraint exist in the routing table.

From Fig. 12, it is observed that tournament selection method gives much better result than the roulette wheel and truncation selection methods. The experimental results are shown in the graph. In case of truncation selection method, the objective function values are quite high through out the generations and do not lead to convergence (high variations in the objective function values thorough out the generations). In case of roulette wheel method, the average objective function

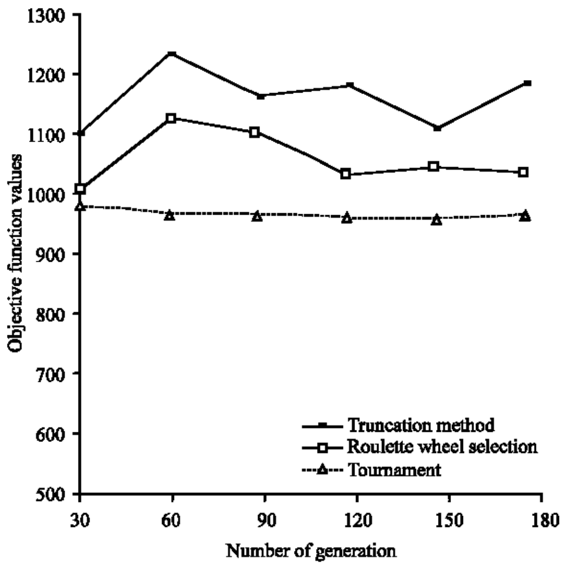

Fig. 12: Comparison of objective function values using different selection methods

values are quite high but leads to convergence. But tournament selection method converges during 60th generation and average objective function values are better than truncation and roulette wheel selection methods.

CONCLUSION

This study has proposed an improved genetic algorithm to solve the Dynamic QoS multicast routing in Multimedia applications. Our proposed algorithm optimizes the cost and end-to-end delay simultaneously. The results obtained from the simulation prove that it has minimal cost, minimal end-to-end delay and better convergence rate than the conventional algorithms. It also has high search success rate and dynamic request success rate. Heuristic local search function embedded in our protocol helps to increase the convergence speed and to get the optimized results. Our proposed algorithm also adapts to dynamic changes (both dynamic group membership and node/link failure). Our experimental result shows that tournament selection method is the best choice for dynamic QoS routing problem. Simulations also prove that our IGADQoS is relevant and efficient even in large size networks. In the forthcoming months, we plan to apply this technique to wireless and MANETS.

ACKNOWLEDGMENT

This research was supported by Center of Relevance and Excellence (CORE) in Network Engineering at Arulmigu Kalasalingam College of Engineering. CORE was sanctioned by Technology Information Forecasting and Assessment Council (TIFAC), Department of Science and Technology, Government of India.

REFERENCES

Atzori, L. and A. Raccis, 2002. Network capacity assignment for multicast services using genetic algorithm, IEEE. Commun. Lett., 8: 403-405.

Bao, G., Z. Yuan, Q. Zheng and X. Chen, 2006. A Novel genetic algorithm to optimize QoS multicast routing, Lecture Notes in Control and Information Sci., pp: 150-157.

Barolli, L., A. Koyama, T. Suganuma and N. Shiratori, 2003. A Genetic Algorithm Based QoS Routing Method for Multimedia Communications over High-Speed Networks, IPSJ. J., .44: 544-552.

Bui, T.N. and B.R. Moon, 1996. Genetic algorithm and graph partitioning, IEEE Transaction on Computers, 45: 841-855.

Cantu-Paz, E., 2000. Efficient and Accurate parallel Genetic Algorithm. Norwell, MA: Kluwer.

Chang Wook Ahn. and R.S. Ramakrishnan, 2002. A Genetic Algorithm for Shortest Path Routing Problem and the Sizing of Populations, IEEE Trans. Evol. Comput., 6: 566-579.

Charikar, M., J.S. Naor and B. Schiebar, 2004. Resource optimization in QoS multicast routing of real time multimedia, IEEE/ACM. Trans. Networking, 12: 340-348.

Chotipat Pornavalai, Goutam Chakraborty and Norio Shiratori, 1998. Routing with multiple QoS requirements for supporting multimedia applications, J. Telecommu. Sys., 9: 357-373.

Chotipat, P., C. Goutam and S. Norio, 1995. Neural network approach to multicast routing in real time communication networks, IEEE. Int. Con. Network Protocols, pp: 332-339.

Davis, L., 1991. Handbook of Genetic Algorithms. Van Nostrand Reinhold.

Debasissh Chakraborty, Goutam Charaborty and Norio Shiratori, 2003. A Dynamic multicast routing satisfying multiple QoS constraints. Int. J. Network Manage., 13: 1-15.

Gelenbe, E., A. Ghanwani and V. Srinivasan, 1997. Improved neural heuristics for multicast routing, IEEE. J. Selected Areas in Commun., 15: 147-155.

Haghighat, A.T. K. Faez, M. Dehghan, A. Mowlaei and Y. Ghahremani, 2004. GA based heuristic algorithms for bandwidth-delay-constrained least-cost multicast routing. Computer Commu., 27: 111-127.

Imase, M. and B. Waxman, 1991. Dynamic Steiner tree problems, SIAM. J. Disc. Mathe., 4: 369-384.

Jia, X., N. Pissinou and K. Makki, 1997. A real time multicast routing algorithms for multimedia applications, Com. Commun., 20: 1098-1106.

Jog, P. J. Suh and D. Gucht, 1989. The effect of Population size, Heuristic Crossover and Local Improvement on a Genetic Algorithm for the Travelling Salesman Problem. Pro. Int. Conf. Genetic Algorithms, pp: 110-115.

Kompella, V.P., J.C. Pasquale and G.C. Polyzos, 1993. Multicast routing for multimedia Communication, IEEE /ACM Trans. On Networking, 1: 286-292.

Korkmaz, T. and M. Krunz, 2001. A randomized algorithm for finding a path subject to multiple QoS requirements, Computer Networks, 36: 251-268.

Kuipers, F. and P. Van Mieghem, 2002. MAMCRA: A Constrained-based multicast routing algorithm, Computer Commun., m25: 802-811.

Lee, Y. and M. Atiquazzaman, 2005. Least cost heuristic for the delay-constrained capacitated minimum spanning tree problem, Computer Commun., 28: 1371-1379.

Leung, Y., G. Li and Z.B. Xu, 1998. A genetic algorithm for the multiple destination routing problems, IEEE. Tran. Evol. Computation, 2: 50-161.

Naryaez, P., K.Y. Siu and H.Y. Tzeng, 2000. New Dynamic algorithms for shortest path tree computation, IEEE. Trans. Networking, 8: 734-746.

Özgür Yeniay, 2005. Penalty Function Methods for Constrained Optimization with Genetic Algorithms, J. Mathe. Compu. Applications, 10: 45-56.

Rose, C. and A. Yener, 1997. Genetic algorithm applied to Cellular Call Admission Control:Local Policies, IEEE Trans. Vehicular Tech., pp: 46.

Sheu, P.R. and S.T. Chen, 2002. A fast and efficient heuristic algorithm for the delay-and delay variation-bounded multicast tree problem, Computer Commu., 25: 825-833.

Shi, J., L. Zou and T.L. Dong, 2000. The application of genetic algorithm in multicasting, ACTA Electronic Sinica, 28: 88-91.

Shimamoto, N., A. Hiramatuand K. Yamasaki, 1993. A dynamic routing control based on a GA, IEEE Int. Con. Neural Network, pp: 1123-1128.

Striegal, A., G. Manimaran, 2002. A survey of QoS multicasting issues, IEEE Commun. Mag., 40: 82-87.

Sun, Q. and L. Li, 2004. Optimizing on multiple constrained QoS multicast routing algorithms based on GA, J. Sys. Engin. Elect., 15: 677-683.

H.Y. Tyan, J.C Hou and B. Wang, 2003. Many-to-many multicast routing with temporal quality of service quarantees, IEEE Transa. Computers", 52: 826-832.

Vijayalakshmi, K and S. Radhakrishnan, 2005. Dynamic Routing from one to group of nodes using Elitism based GA-Novel Multiparameter Approach, IEEE International conference on emerging trend in electrical and Information technology (INDICON), pp: 565-569.

B.Wang and J.C.Hou, 2000. Multicast routing and its QoS extension: Problems, algorithms and protocols, IEEE Network, 14: 22-36.

X.Zheng, B.A. Julstrom and W.Cheng 1997. Design of Vector Quantization Codebooks Using a Genetic Algorithm, Proc. IEEE. Int. Conf. Evolu. Computa., pp.525-529.

W.Zhengying, W.Bingxin and Z.Erdun, 2001. Bandwidth-delay-constrained least cost multicast routing based on heuristic genetic algorithm, Computer commun. 24: .685-692.