Dynamic Routing to Multiple Destinations in IP Networks using Hybrid Genetic Algorithm (DRHGA)

K. Vijayalakshmi and S. Radhakrishnan

Abstract—In this paper we have proposed a novel dynamic least cost multicast routing protocol using hybrid genetic algorithm for IP networks. Our protocol finds the multicast tree with minimum cost subject to delay, degree, and bandwidth constraints. The proposed protocol has the following features: i. Heuristic local search function has been devised and embedded with normal genetic operation to increase the speed and to get the optimized tree, ii. It is efficient to handle the dynamic situation arises due to either change in the multicast group membership or node / link failure, iii. Two different crossover and mutation probabilities have been used for maintaining the diversity of solution and quick convergence. The simulation results have shown that our proposed protocol generates dynamic multicast tree with lower cost. Results have also shown that the proposed algorithm has better convergence rate, better dynamic request success rate and less execution time than other existing algorithms. Effects of degree and delay constraints have also been analyzed for the multicast tree interns of search success rate.

Keywords—Dynamic Group membership change, Hybrid Genetic Algorithm, Link / node failure, QoS Parameters.

I. INTRODUCTION

ALL the IP networks require Dynamic QoS (Quality of Service) based multicasting. Multicasting is the ability of finding the best path to transmit the data from a source node to multiple destination nodes in a particular network with QoS parameters [1], [2] and is shown to be NP complete (Non Deterministic Polynomial) problem [3], [4]. Growing demand of various real time group applications such as distance learning, video conferencing, online video/audio for news and multi-person communication require effective way of communications among multiple hosts with strict QoS parameters.

Many real time applications in IP networks require the

K.Vijayalakshmi is with the Mepco Schlenk Engineering College, India, 626005 and Research Scholar, Anna University, India. (phone: +91-4563-236214; e-mail : vijaya@mepcoeng.ac.in, & vijaya73mepco@gmail.com).

S. Radhakrishnan is with Arulmigu Kalasalingam College of Engineering, Krishnankoil, India. He is the project director for Network Technology group in CORE(TIFAC), India (e-mail: srk@akce.ac.in).

dynamic group membership of nodes. That means any node (person) could leave or join into the multicast group during run time. Also link / node failure occurs in real time applications. So we need an algorithm such that either leaving/entering a node into the multicast group or link/node failure does not require reconstruction of tree, and needs just modification in the current transmission (tree). This type of dynamism saves time and resources. These applications need delay, bandwidth constraint and least cost multicast tree construction [2], [3], [4]. Further such applications require two important criteria: first one is to obtain the dynamism and the second is QoS routing to multiple destinations.

Several heuristic algorithms are available for finding delay-constrained, least cost multicast tree [3], [5]. All these heuristic algorithms exhibit unacceptable high computational complexity for real time communication and leads to local optimum values. Also the average execution time grows rapidly as the network size grows.

Genetic Algorithm (GA) is a guided random search technique to solve large-scale optimization problems and combinatorial problems. It works based on the principle of evolution and uses payoff function to guide the random search [6], [7]. Various researchers proposed the construction of static least cost multicast tree which satisfy either single or multiple constraints using simple GA [8] - [13]. But all the above algorithms produce the static multicast tree and some times converged to local optimum.

Researchers have also proposed several traditional heuristic dynamic algorithms for handling the group membership [14] - [17]. These dynamic algorithms produced the least cost multicast tree and they have considered only group member ship. Some times these dynamic heuristic algorithms produced sub optimal values. Mostly these algorithms searched the entire network in sequential manner (try all the paths) and not in stochastic manner. Xu and Chen [18], [19] have proposed a fixed delay constrained multicasting with dynamic membership change. In their paper they considered only the group membership changes and delay constrained multicasting. But in all IP real time application, users are located in different distances and hence they require the bandwidth and delay constraints for transmitting messages to particular users. Also here dynamic situation arises due to either change in the multicast group membership or node /

link failure.

In realistic communication network, each node is a switch or router and its multicasting ability is different from each other. If more number of information is given to a particular router then that router consumes more time and also other resources. Limiting the count of information to a router saves time in real time applications. Ability of a Router in supporting the multicast is referred as degree constraint. This constraint helps to keep the network in balanced condition by balancing the work of router.

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. Hybrid 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 [7]: they are local search, use of special genetic operators and problem specific encoding.

So we propose an algorithm for developing a degree, delay and bandwidth constrained least cost dynamic multicast tree using hybrid GA. Our protocol's convergence speed is very high due to the following reasons:

i. Random initial population.
ii. GA based Routing table construction.
iii. Chromosome representation depends on number of destinations.
iv. Dynamic algorithm for
   i. Updating node or link failure.
   ii. Joining or leaving the member in the multicast group.
v. Heuristic Local search function embedded with Simple GA.
vi. 'm' point cross over.
vii. Two different mutation and cross over probabilities for maintaining diversity in solution and quick convergence.
viii. Removal of overlapping vectors during the first 50% of generation – helps to search the space quickly.

II. PROBLEM DESCRIPTION

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∈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∈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∈V is the source node, T=$\{v_1,v_2,v_3,…v_k\}$ 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, $v_i∈T$. The goal is to find a multicast tree between a single source and set of destinations, which will optimize the monitory cost and adapts for dynamic change (multicast group membership change and node/link failure).

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:

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

Node degree function of any router (r) is defined as follows:

$$D_e(r) = \sum_{j \in V} \gamma(r, v_j)$$

$$\gamma(r, v_j) = \begin{cases} 1 & if\ (r, v_j) \in E \\ 0 & otherwise \end{cases}$$

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

Our protocol finds the multicast tree with minimum cost, subject to delay, degree and bandwidth constraint, which is formulated as follows:

$$Min(C_t(X))$$

Subject to

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

$$D_e(r) < D_{gm}, \qquad \forall r \in T$$

$$D_y(X) < \Delta$$

Each node in the multicast tree will not exceed the degree constraint; D_{gm}. B_d is the bandwidth constraint of the destination node d in T and Δ is the delay constraint.

In our protocol which is named as 'Dynamic routing to multiple destinations using Hybrid Genetic Algorithm' (DRHGA), first we have constructed the routing table (optimized routes) using GA based algorithm. Then we have constructed the multicast tree using these routes.

III. ROUTING TABLE CONSTRUCTION

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 [20]. Here a HGA based QoS routing method is used to find the optimal routes and these are stored in routing table. Optimal routes are placed in sorted order based on fitness value in the routing table. Network is configured as chromosome based on the dynamic multi-parameter encoding mechanism [21]. Depth first search is used for population initialization. Every

chromosome after the initialization phase is different from each other. This is because of removal of repeated chromosomes in the initialization phase. This helps to search the space at different places (randomly), which increases the convergence rate. Objective function is the minimization of cost under the constraints of delay, bandwidth and degree.

High mutation probability (P_m) in the first half of generation helps to maintain the diversity of solution. Convergence rate is increased due to high crossover probability. Low cross over probability (P_c) avoids the solution to fall into local minimum. We have used the following GA operators, P_m, and P_c parameters for our GA method.

Selection: Tournament selection with out replacement.
Mutation: For the first 50 % of total generation, P_m is 0.54 and for the next half, P_m is 0.05
Crossover: For the First 50 % of total generation, cross over probability (P_c) is 0.15(to get optimal solution) and for the next half, P_c is 0.44.

IV. HYBRID GENETIC ALGORITHM FOR DYNAMIC ROUTING TO MULTIPLE DESTINATIONS

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

A. Genetic Operators Used in DRHGA

Genetic algorithm is a generalized search and optimization technique. 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. Three basic operations namely: reproduction, crossover, and mutation are adopted in the evolution to generate new offspring.

Reproduction operator is used to improve the quality of the population by selecting the high-quality (fittest) individuals, which is copied to next generations. Pair wise tournament selection without replacement (tournament size=2) is used in the proposed model. In tournament selection method, 'n' ('n' is equal to tournament size) individuals are selected randomly from the population and best one among 'n' is entered into new generation, which serves as a parent to next generation. The same individual should not be selected twice as a parent. This process is repeated for population size (P_S). Convergence of GA 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 is used to cross breed the individuals. Using crossover operator, information between two chromosomes are exchanged which mimic the mating process. Here we use the 'm' point crossover. A pair of high fitted parents selected from the population randomly. Cross over operator chooses 'm' cutting points randomly from the parents and alternatively interchanges each segment between two parents. This operation is depicted in Fig. 1.
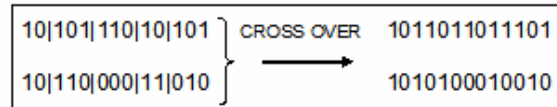

Fig. 1 '4' Point Cross over operation

Mutation operator changes 1 to 0 and vice versa with small probability P_m. 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. Mutation operation at the 9th bit is shown in Fig. 2.
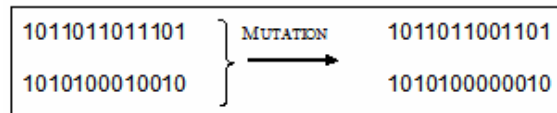

Fig. 2 Mutation operation

B. Encoding

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 Table 1.

TABLE I
CHROMOSOME REPRESENTATION

G_1	G_2	…	G_j	…	G_k

C. Geno_Pheno Function

Gene $G_j = 1$, then Geno_Pheno function gives the integer value in {0,1,2,…,R-1} which is the routing table's (s → v_j) route number. From the routing table the path route from s to v_j is selected. This coding method is first proposed by N.Shimamoto et al [21] for the point-to-point routing problem. Major advantages of the method is that chromosome representation requires only k bits (k = number of destinations nodes), and this method has the ability to solve routing problem very quickly even for large size networks. For example,

Gene: 100101

Geno_Pheno function gives the correct row entry numbers of routing tables s → v_4, s→v_1 and s → v_6

The corresponding path route is taken from routing tables s→v_1, s→v_4 and s→v_6 and constructs the multicast tree.

D. Population Initialization and Fitness Function

During initialization of chromosomes we assign the population size (P_S) is equal to 33 and select the path routes among 0 to R/3 rather than among 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. Repeated chromosomes are removed in the initialization phase (all chromosomes are different from

each other). This helps to search the solution space at different places (randomly) which increases the convergence rate. At each step, we use the repair function to check the loop formation and to avoid duplication. Algorithm for population initialization is shown below:

```
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=0;
    }
    If (Check _ feasibility ( i^th chromosome ))
        n++;
        }
    }
```

Fitness function is given as follows:

$$f(x) = (1/(C_t(x))) * A * B * C \qquad (1)$$

$$A = (\prod_{d \in T} \varphi(B(P(s,d)) - B_d))$$

$$B = \prod_{d \in T} (\varphi(D_y(x) - \Delta))$$

$$C = \prod_{d \in T} (\varphi(De(r) - D_{gm}))$$

$$\varphi(h) = \begin{cases} 1 & h > 0 \\ \lambda & h \le 0 \end{cases}$$

Penalty functions are used to handle the constraints [22]. These methods transform the constrained problem to unconstrained problem. These methods 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. φ(h) is the penalty functions[22] and λ is the penalty factor. In our protocol λ value is taken as 0.56.

E. Heuristic Local Search Function

Heuristic functions are used along with the genetic algorithm. This helps to increase the convergence speed and to get the optimal tree. This function takes three high fitted chromosomes as parents and then calculates the cost for the three chromosomes from source node to all destinations. Chromosomes cost CT_1, CT_2 and CT_3 are calculated as follows:

$$CT_1 = \{P_1(s,d_1), P_1(s,d_2),....P_1(s,d_n) \}$$
$$CT_2 = \{P_2(s,d_1), P_2(s,d_2),....P_2(s,d_n) \}$$
$$CT_3 = \{P_3(s,d_1), P_3(s,d_2),....P_3(s,d_n) \}$$

We calculated the sub-fitness value of the path using (2).

$$P_i(s,d_j) = C(s,d_j) \qquad (2)$$

Next, this function compares the sub-fitness values in the same destination for the above three chromosomes. Higher fitness value of sub-fitness value for every destination is taken to create the new offspring. Best three among the four is selected for next generation.

For example, consider

$CT_1 = \{ 3{\to}4, 3{\to}5{\to}6{\to}2, 3{\to}5{\to}7\}$,
$CT_2 = \{ 3{\to}8{\to}4, 3{\to}15{\to}2, 3{\to}1{\to}5{\to}7\}$
$CT_3 = \{ 3{\to}1{\to}4, 3{\to}6{\to}2, 3{\to}6{\to}5{\to}7\}$

If the sub-fitness value for the destination '4' in CT_1 is better than CT_2 and CT_3, then we take sub-fitness value of $3{\to}4$ for new offspring. Similarly if the sub-fitness value of '$3{\to}15{\to}2$' in CT_2 and the sub-fitness value of '$3{\to}5{\to}7$' in CT_1 give better value than the other two corresponding sub-fitness values then this function generates a new offspring as $\{3{\to}4, 3{\to}15{\to}2, 3{\to}5{\to}7\}$. Now best three chromosomes among four chromosomes (CT_1, CT_2, CT_3, new offspring) are selected for next generation.

F. Multicast Routing using Hybrid GA (MR)

This algorithm generates a static multicast tree, which has two parts: construction of optimized routing table and construction of multicast tree. In the following we explain the algorithm:

Step 1: Input() –We get the following Inputs : Number of nodes, C_{ij}, D_{ij}, B_{ij}, destination nodes $\{v_1, v2,v3,. , .,v_k\}$, Δ, B_d, number of generation(N_s) and source s.
Step 2: We assign P_m is 0.54, P_{m1} is 0.05, P_{c1} is 0.44 and P_c is 0.15
Step 3: We construct the routing table using GA based method as described in section III.
Step 4: We initialize the populations according to section IV.D
Step 5: The fitness value of all chromosomes are evaluated using (1).
Step 6: **Genetic operation:**
  i. We use tournament selection method for selecting the individual for next generation
  ii. Elitism – 5% of high fitted individual are directly selected for next generation.
  iii. Crossover – 'm' point cross over operator is used with a probability of P_c.
  iv. Heuristic function- We select high fitted three individuals and create a new individual from these three as described in section IV.E. Then we select best three among four and add to next generation.
  v. New chromosome is generated based on mutation probability P_m.

vi. After mutation we perform heuristic local search function at the probability of 0.15.

Step 7: The overlapped vectors (repeated vectors) are removed at each generation .

Step 8: We repeat Step5, Step 6 and Step 7 at $((N_s)/2)$ times.

Step 9: We repeat Step 5 and Step 6 at $((N_s)/2)$ times with the cross over probability P_{c1} and mutation probability P_{m1}.

Step 10: We construct the multicast tree using both Geno_Pheno function and routing table.

Step 11: Output— we obtain the optimized Multicast tree.

Elitism is used here for maintaining the best feasible solution in the population that provides an exhaustive search. Elitism helps to avoid the removal of high fitted individuals during GA operation. We have used repair function to avoid the infeasible chromosomes which is formed during cross over and mutation. This function also avoids the formation of loop.

G. Dynamic Algorithm

In case of real time multimedia application the following situations are occurred

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

For the given input, first we call the MR 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.

Handling of above situations by our dynamic algorithm is explained below:

In the first situation, if any new node 'v' enters into the multicast group then the algorithm checks whether v is already part of the multicast tree for transmitting messages (not for user, only mediator). If it is so, then it initiates 'v' to participate in multicasting and it is allowed to send information to its users. Otherwise finds 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 the algorithm removes 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 our algorithm removes 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 it finds the next optimal route from v_n's parent (proper ancestor) to v_n's child

(proper descendent) and connects 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 MR algorithm.

If the failed link is not present in X, there is no change in the multicast group but that information is broadcasted to all the routers that help the associated nodes to run their routing algorithm.

H. 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.

V. SIMULATION AND PERFORMANCE STUDY

DRHGA algorithm was simulated on Pentium IV 512 MB PC in C language. In this simulation we have used the following parameters:

Size of Routing table = R/3; Number of generation =150.
Tournament size =2; Group size = 35 % of Network size.
Population size = 33.

Crossover probability and mutation probability have already been mentioned in MR algorithm. Delay of link is randomly chosen with in range [0, 60 ms], the cost of each link is generated within range [0,200], bandwidth is within range [50-100 kbps] and Delay constraint (Δ) is 45 ms. We have randomly selected the network topology to perform the simulation. High mutation rate(P_m) during the initial 50 % of total generations help to maintain the diversity of solution and low cross over probability(P_c) during the initial 50 % of total generations help to skip the local optimum values. These two parameters tend to slow down the converging speed and hence we have used P_{c1} and P_{m1} probability for the second half of generations. Performance of our protocol is compared with NGA[10], QoSMR[11], MRMLS[12] NDA[16], and EADM[18]. Our performance metric measures include search success rate, cost, convergence rate, execution time, and dynamic request success rate. The effect of degree constraint and delay constraint are also analyzed.

Fig. 3 and Fig. 4 show the search success rate of our algorithm. Search success rate is nothing but how often an algorithm could find a route that can satisfy all the QoS constraints. Due to hybridization and diversity of solution, our algorithm achieves the 100% of search success rate for large sized networks also. For Fig. 3, group size is 35 % of network size. The plot present in the Fig. 3 shows the effect of degree constraint with search success rate for different

degree constraints. When the node degree is increased, more number of alternatives occurred (increase) that helps to find the cheaper/optimized routes. Load of the router is balanced due to degree constraint and thus the search success rate is increased.
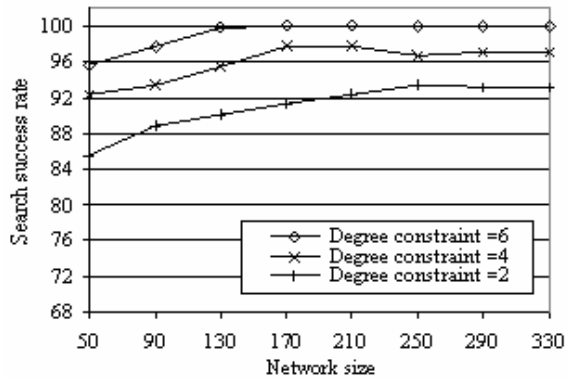

Fig. 3 Search success rate versus Network size for different Degree constraints

Comparison of search success rate with different delay constraints for networks with different sizes is shown in Fig.4. When the delay constraint is large, the search success rate of our algorithm increases. This is because many routes which satisfy the required delay exist in the routing table. When we strict the delay constraint, it decreases the overall performance. When we relaxed the constraints, at some point it produces the 100 % success rate. For a connection degree constraint 6, bandwidth constraint 80kbps and delay constraint 45ms, our algorithm produces the optimized result.

Fig. 5 and Fig. 6 show that our protocol produces better tree cost performance than existing algorithms. DRHGA achieves better optimal tree cost in both small and large networks. From Fig. 6, it is seen that the proposed algorithm exhibits fastest convergence rate than other algorithms for various group sizes. For Fig. 6 the group size is 35 % of network size (100).
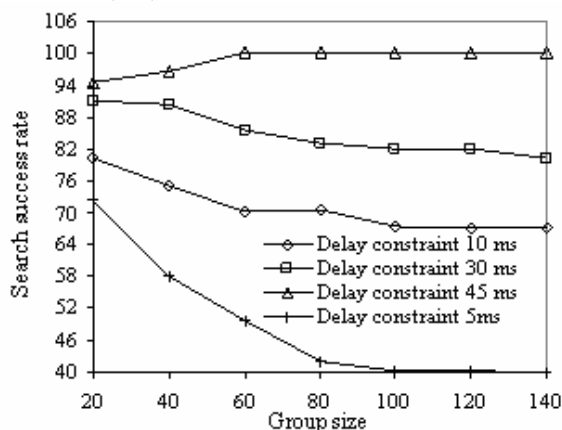

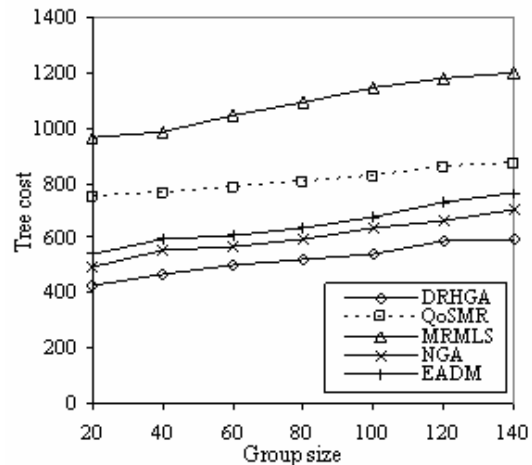Fig. 4 Search success rate versus Group size for different delay constraints.

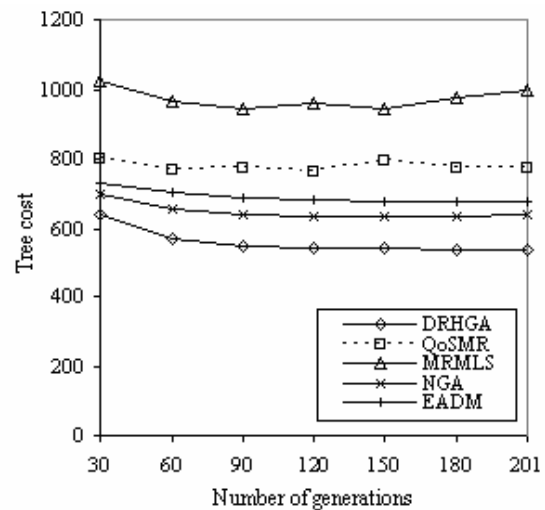Fig. 5   Tree cost versus Group size

Fig. 6 Convergence property of our algorithm

As seen in the Fig. 7, DRHGA required less execution time because of applying hybrid genetic algorithm to find multicast tree. Pc, Pc1, Pm, Pm1 is used to increase the convergence speed and to obtain the optimized results.

Dynamic property of our algorithm is measured by dynamic request success rate. Dynamic request Success rate (D_{req}) is the ratio of total number of dynamic request success to the total number of dynamic requests. Dynamic request contain both membership change (join/leave request into/from multicast group) request and link/node failure. Dynamic request success rate of our algorithm for various network sizes are shown in Fig. 8. Here the average connection degree is six, delay constraint is 45ms and bandwidth is 80 kbps. It is observed that 93% of dynamic request success rate is achieved in our algorithm.

According to all the above analysis DRHGA produces better results than other protocols.
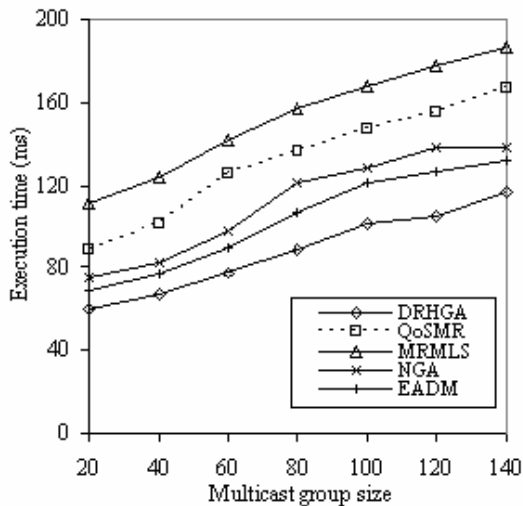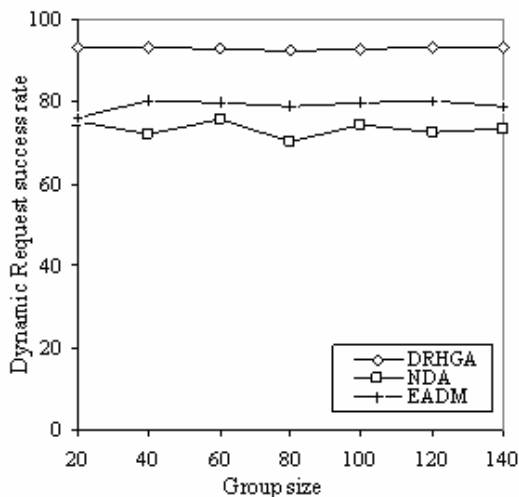
Fig. 7 Execution time versus Multicast group size

Fig. 8 Dynamic request success rate versus Group size

VI. CONCLUSION

In this work, we hybridize GA by incorporating the heuristic local search function and by using improved genetic operators. Hybrid GA is used in both unicast and multicast routing in the proposed algorithm. Our algorithm adapts for dynamic change (multicast group membership change and node/link failure). Our dynamic algorithm takes less time since it uses the routes from already optimized routing table, which was created during the first stage of our algorithm. Average Dynamic request success rate of our algorithm is found to be 93 % and is found to be better than the existing algorithms. Our algorithm produces the optimized result for degree constraint 6, and delay constraint 45ms. From the simulation results we conclude that DRHGA has minimal cost, less execution time, fast convergence rate than the existing algorithms. Future work will focus on other networks such as wireless networks, MANETS, MPLS networks and Active networks.

REFERENCES

[1]   Laxman H. Sahasrabuddheh Mukherjee, "Multicast Routing Algorithms and Protocols: A Tutorial," IEEE Network, pp. 90-102, Jan/Feb.2000.
[2]   L .Barolli, A. Koyama, T. Suganuma and N. Shiratori, "A Genetic Algorithm Based QoS Routing Method for Multimedia Communications over High-Speed Networks," IPSJ Journal, Vol.44, No. 2, pp. 544-552, February 2003.
[3]   V. P. Kompella, J .C Pasquale and G. C. Polyzos, "Multicast routing for multimedia Communication," IEEE / ACM Trans. On Networking, Vol. 1(3), pp. 286-292, 1993.
[4]   A. Striegal, G. Manimaran, "A survey of QoS multicasting issues," IEEE Communications Magazine, vol.40, pp. 82-87, 2002.
[5]   M. Parsa, Q. Zhu, and J. J. Garcia-Luna-Aceves, "An iterative Algorithm for Delay-Constrained Minimum-Cost Multicasting," IEEE/ACM Transactions on Networking, Vol. 6(4), pp.461-474, 1998.
[6]   T. N. Bui and B. R. Moon, "Genetic algorithm and graph partitioning," IEEE Transaction on Computers, Vol. 45(7), pp. 841-855, 1996.
[7]   L. Davis, Handbook of Genetic Algorithms. Van Nostrand Reinhold, 1991, Chapter 6 & 7.
[8]   W. Zhengying, S. Bingxin and Z. Erdun, "Bandwidth-delay-constrained least cost multicast routing based on heuristic genetic algorithm," Computer Communication, Vol. 24 (7-8), pp. 685-692, 2001.
[9]   Q. Sun and L. Li, "Optimizing on multiple constrained QoS multicast routing algorithms based on GA," Journal of Systems Engineering and Electronics, Vol, 15(4), pp. 677-683, 2004.
[10] G. Bao, Z. Yuan, Q. Zheng and X. Chen, "A Novel genetic algorithm to optimize QoS multicast routing," Lecture Notes in Control and Information Sciences, Vol. 344, pp. 150-157, 2006.
[11] A. T. Haghighat, K. Faez, M. Dehghan, A. Mowlaei, Y. Ghahremani, "GA-based heuristic algorithms for QoS based multicast routing," Knowledge-Based System, Vol. 16, pp. 305-312, 2003.
[12] C. F. Tsai, C. W. Tsai, C. P. Chen, "A novel algorithm for multimedia multicast routing in a large scale networks," The journal of Systems and Software 72, pp. 431-441, 2004.
[13] C. P. Ravikumar and R. Bajpai, "Source based delay-bounded multicasting in multimedia networks," Computer communications, Vol.21, pp. 111-127, 2004.
[14] Debasissh Chakraborty, Goutam Charaborty and Norio Shiratori, "A Dynamic multicast routing satisfying multiple QoS constraints," International journal of network management, vol. 13 (5), pp 1 – 25, 2003.
[15] M. Imase and B. Waxman, "Dynamic Steiner tree problems," SIAM Journal of Disc. Mathematics, Vol. 4(3), pp.369-384, 1991.
[16] P. Naryaez, K. Y. Siu and H. Y. Tzeng, "New Dynamic algorithms for shortest path tree computation," IEEE/Transaction on Networking, Vol. 8(6), pp. 734-746, 2000.
[17] N. Rammohan and C. Siva Ram Murthy, "On-line multicast routing with QoS constraints in WDM networks with no wavelength converters," Computer Networks, Vol. 50(18), pp. 3666-3685, December 2006.
[18] Z. Xu and L. Chen, "An effective algorithm for delay-constrained dynamic multicasting," Knowledge based systems, Vol 19(3) , pp. 172-179, 2006.
[19] L. Chen and Z. Xu, "Effective multicasting algorithm for dynamic membership with delay constraint," Journal of Zhejiang University, Vol. 7(2), pp. 156-163, 2006.
[20] Xin Yuan, "Heuristic Algorithms for multiconstrained QOS Routing," IEEE / ACM Transaction on Networking Vol.10 (2), April 2002.
[21] N. Shimamoto, A. Hiramatuand K. Yamasaki, "A dynamic routing control based on a GA," IEEE International Conference on Neural Network, 1993, pp. 1123-1128.

[22] Özgür Yeniay, "Penalty Function Methods for Constrained Optimization with Genetic Algorithms", Journal of Mathematical and Computation Applications, Vol. 10(1), pp. 45-56, 2005.

Mrs. K. Vijayalakshmi has received a degree and Master degree in Computer Science and Engineering from Madurai Kamarajar University. She is currently pursuing Ph.D degree in Faculty of Information and Communcation Engineering department, Anna University, India. Her fields of interest are Computer Networks, Evolutionary computing, Data Sturctures and Analysis of algorithms. She has more than 10 years of experience in teaching and Research. She has more than 10 publications to her credit in international journals and conferences.

Dr. S. Radhakrishnan is presently working as Senior Professor and Head, Department of Computer Science and Engineering at Arulmigu Kalasalingam College of Engineering, Krishnankoil, Tamilnadu. He is also serving as the Project Director (Network technologies) in TIFAC CORE in Network Engineering at A.K.College of Engineering. He has completed his M.Tech. and Ph.D. in Biomedical Engineering from Institute of Technology, Banaras Hindu University. He has more than 18 years of experience in teaching and research. He has guided more than 50 M.E./M.Tech. projects, and 10 M.Phil.Thesis. His fields of interests are Network Engineering, Computer Applications in medicine and evolutionary computing. He has more than 50 publications to his credit in international journals and conferences.