A Lexicographically Fair Allocation of Discrete Bandwidth for Multirate Multicast Traffics

Chae Y. Lee, Young P. Moon and Young Joo Cho

Dept. of Industrial Engineering, KAIST,

373-1 Kusung Dong, Taejon, Korea

Abstract

Fair bandwidth allocation is an important issue in the multicast network to serve each multicast traffic at a fair rate commensurate with the receiver's capabilities and the capacity of the path of the traffic. Lexicographically fair bandwidth layer allocation problem is considered and formulated as a nonlinear integer programming problem. A nonincreasing convex function of the bandwidth layers of the virtual sessions is employed to maximize the bandwidth of each virtual session from the smallest.

To solve the fairness problem a genetic algorithm (GA) is developed based on the fitness function, ranking selection and the shift crossover. Outstanding performance is obtained by the proposed GA in various multicast networks. The effectiveness of the GA becomes more powerful as the network size increases.

1. Introduction

With the explosive increase of Internet service new traffics such as video conference, video on demand, online education, etc are preferred for real time applications. A solution to these one-source multiple-receiver traffics that share the limited network resource is multicasting. Thus multicasting poses some specific fairness issues in the IP network.

In multicast network a fair resource allocation among sessions is essential to adapt to the different QoS requirement by each application. This is the issue of inter-session fairness, i.e., fairness of members across multiple sessions. In addition to the inter-session fairness intra-session fairness is another issue that has to be solved. In each multicast session receivers are allowed to receive information at different rates. This is mainly due to the network bandwidth heterogeneity of the receivers that belong to the same multicast group. The requirement of different rates even in the same multicast session can be delivered by multirate layered transmission [1, 2, 3]. The source signal is encoded and presented to the network as a set of bit streams, called layers. The layers are so organized that the quality of reception is proportional to the number of layers received. The first layer provides the basic information, and every other layer improves the data quality. Therefore, both the inter-session and intra-session fairness problem has to be solved by the multirate layered transmission to satisfy the receivers in multiple multicast groups.

In the literature several types of fairness are defined that include the max-min fairness [4, 5, 6], lexicographically optimal [7] and maximal fairness [7, 8, 9]. Sarkar and Tassiulas [7] prove that the computation of lexicographically optimal layer allocation is NP-hard. They thus provide a polynomial algorithm for maximally fair allocation of discrete layers [7, 8]. However, note that the lexicographically optimal allocation is the stronger concept of fairness in the sense that if a layer allocation is lexicographically optimal, then it is maximally fair. Thus, we in this paper are interested in the lexicographically optimal fair allocation of bandwidth.

Other fairness objectives include the utility maximization [10, 11]. The utility function of a receiver represents the value associated with the bandwidth assigned to the receivers. Rate control algorithms [10, 11] are proposed to maximize the sum of utilities over all receivers, subject to the link capacity constraints.

In this paper we consider lexicographically fair allocation of the bandwidth layers among multirate multicast sessions. The lexicographic fairness is the best solution to find fair allocation in discrete case in view of the nonexistence of max-min fair allocation. The fairness problem is formulated as a nonlinear integer programming model. For our fairness objective, we introduce a nonincreasing convex function of the bandwidth that maximizes the bandwidth of the smallest session first. Subject to this it maximizes the second smallest, and so on. The nonlinear model is restricted by the link capacity constraint and the minimum bandwidth requirement by each receiver. A genetic algorithm (GA) is proposed to solve the nonlinear integer programming problem. Several selection methods are considered for the strings that survive generation by generation. Shift crossover and mutation operator are employed for the alteration of genes in each string. Lexicographically fairer solutions are obtained by the proposed GA compared to the best known existing algorithm.

This paper is organized as follows. In Section 2, we briefly discuss the fairness; max-min fairness, lexicographically optimal and maximal fairness. A nonlinear integer programming model is provided for the lexicographically optimal fair bandwidth allocation problem in Section 3. A genetic algorithm is developed to solve the fairness problem in Section 4.

Computational result and conclusion are presented in Section 5 and Section 6 respectively.

2. Fairness in Bandwidth Allocation

When a network has profound heterogeneity, the fairness must include characteristics of multi-rate multicast network. Each multicast session transmit data to all of its receivers at different rate. One of the frequently used definitions of fairness in multi-rate multicast networks is max-min fairness [5, 12]. Informally speaking, a rate allocation is max-min fair, if no receiver can be allocated a higher rate without hurting another receiver having equal or lower rate.

As an example, consider the network in Figure 1. It has one source and three destination nodes with two links. The bandwidth of each link capacity is 6 and 5 units respectively. The max-min fair allocated rate vector in this network is (6, 2.5, 2.5). If we increase the bandwidth allocated to the second destination, we decrease the bandwidth allocated to the third destination. When continuous allocation of the bandwidth is allowed, the max-min fairness always exists and the allocation procedure is studied by Sarkar and Tassiulas [4, 6]. However, in layered transmission scheme, bandwidth is allocated in discrete fashion. In this case, the max-min fair allocation may not exist.

A lexicographically fair optimal allocation [7], however, exists even in discrete case. A bandwidth allocation vector is lexicographically optimal if its smallest component is the largest among the smallest components of all feasible bandwidth allocation vectors. Subject to this, it has largest second smallest component, and so on. In the network of Figure 1, if the bandwidth is allocated in discrete layer, the max-min fair allocation vector does not exist. However, a lexicographically fair optimal allocation exists and given by (6, 3, 2) or (6, 2, 3).

Note that max-min fairness and lexicographic optimality are not equivalent. The max-min fairness is stronger than lexicographic optimality. If a max-min fair vector exists, it is lexicographically optimal. However, a max-min fair bandwidth allocation may not exist in a discrete case. In view of the nonexistence of max-min fair bandwidth allocation vector, lexicographically optimal bandwidth allocation is the best solution to find fair allocation in discrete case. However, it is known that the lexicographically optimal bandwidth allocation is NP-hard in case of discrete layer allocation [7].

3. A Lexicographically Optimal Fair Bandwidth Allocation Model

Consider a network with N multicast sessions. The traffic of each session is transmitted from a source to a set of destination nodes across a predetermined multicast tree. We call a source and destination pair of a session a virtual session.

For a virtual session i, let x_i be the bandwidth allocated to the virtual session i and u_i be the minimum bandwidth requirement, then we have

$$x_i \geq u_i \qquad i = 1, \ldots, I$$

Now, consider a link l in the network where a set of virtual sessions of session j is passing through. Let $v(j,l)$ be the set of virtual sessions belonging to session j and traversing link l. Note in the multicast tree that the actual bandwidth assigned to the session j is determined by the maximum bandwidth among the virtual sessions. Let the link capacity y_{il} be the maximum, then

$$y_{jl} = \max_{i \in v(j,l)} x_i \qquad j = 1, \ldots, J, \quad l = 1, \ldots, L.$$

Note, the total bandwidth assigned to sessions traversing link l cannot exceed the capacity of link l. By letting $s(l)$ be the set of sessions passing through link l, and c_l be the link

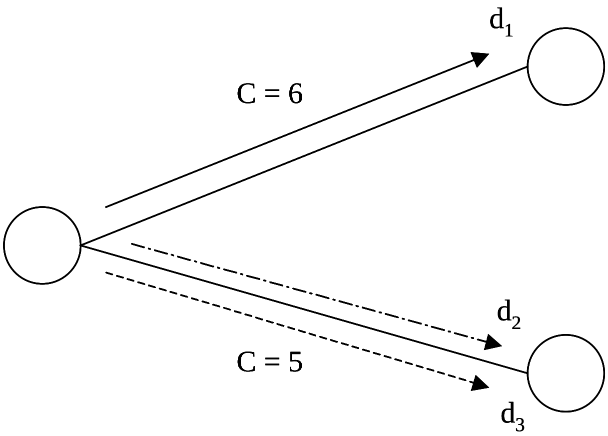

Figure 1. Network of three multicast sessions

capacity constraint, we have

$$\sum_{j \in s(l)} y_{jl} \le c_l \quad l = 1, \ldots, L.$$

Now, our objective is to allocate the bandwidth for each virtual session such that the solution satisfies the lexicographic optimal fairness. Note in the lexicographic optimal solution that the minimum component is maximized among all feasible solutions. Subject to the maximization, the second minimum is maximized, etc.

Thus, we consider a nonincreasing convex function $1/x^p$ where p is a large integer. Clearly the function gives more credit to a virtual session x_i with smaller value, when we minimize the sum of $1/x^p$. Thus, we are interested in the objective function given below.

$$\text{Min} \quad \sum_{i=1}^{I} 1/x_i^{\,p}$$

The above objective function is consistent with the definition of the lexicographic optimal fairness in the following sense. For the unit increase of the bandwidth of a virtual session x_i the improvement of objective function becomes

$$\frac{(x_i+1)^p - x_i^{\,p}}{x_i^{\,p}(x_i+1)^p}$$

Clearly, better improvement is obtained with the smaller x_i. If the minimum virtual session is maximized, then the second minimum is supposed to be maximized when p is sufficiently large. Thus, our bandwidth allocation problem is formulated as follows.

Minimize $\sum_{i=1}^{I} 1/x_i^{\,p}$

subject to:

$$x_i \geq u_i \qquad i = 1, \ldots, I \tag{1}$$

$$y_{jl} = \max_{i \in v(j,l)} x_i \qquad j = 1, \ldots, J, \quad l = 1, \ldots, L \tag{2}$$

$$\sum_{j \in s(l)} y_{jl} \leq c_l \qquad l = 1, \ldots, L \tag{3}$$

$x_i \geq 0$ and integers

As proved by Sarkar and Tassiulas [7], the computation of the lexicographically optimal fair allocation problem is NP-hard. The proposed nonlinear integer programming problem may not be effectively solved by any conventional optimization techniques. Thus, we examine a genetic algorithm as a promising solution procedure for the fair bandwidth allocation problem. Note that the genetic algorithm starts with a population of strings and generates successive populations of strings by using probabilistic transition rules. Therefore the parallel flavor with the population of well-adapted diversity will contribute to the lexicographically fair allocation of the bandwidth among multiple multicast sessions.

4. Genetic Algorithm for Lexicographically Fair Bandwidth Allocation

Genetic algorithms are adaptive procedures that find solutions to problems by an evolutionary process based on natural selection. Motivated by the biological adaptation, they generate a new set of strings from parent chromosomes via stochastic operation. Strings with low fitness values survive and those with high fitness values die off generation to generation.

While randomized, genetic algorithms efficiently exploit historical information to speculate on new search points with expected improved performance.

For this propose we examine various selection methods in the literature. Crossover and mutation operators are employed for the alteration of real-valued genes within a specific range.

4.1. String and Initial Population

Each gene of a string represents the bandwidth of a virtual session. Since a solution has to satisfy the minimum required bandwidth constraint and the link capacity constraints, initial population is generated by feasible strings such that each gene x_i of the string satisfies the three constraints.

a string : $x_1, x_2, x_3, \ldots, x_I$, where x_i is the bandwidth of the *i*-th virtual session

4.2. Fitness function

The objective function employed in the formulation of Section 3 is used to evaluate the fitness of each string. Note that our evaluation function is to minimize the sum of $1/x^p$ that is consistent with the lexicographic optimal fairness. However, there may be cases when no resource can be allocated to a virtual session due to the limited link capacity. For such a case the fitness function needs to be modified. For the case when $x_i = 0$, the fitness function can be modified to $\sum_{i=1}^{I} \left(\frac{1}{x_i + 1} \right)^p$.

4.3. Selection method

We consider four selection schemes commonly used in genetic algorithms [13]. The following four methods are considered and compared.

1) Remainder stochastic without replacement

Get expected count by the strings fitness divided by population fitness average, and put strings as the integer part of expected count into population. Sort the decimal fraction from maximum to minimum, and put strings into the population until the population size is filled.

2) Ranking selection

Sort the population from the best to the worst, assign the number of copies that each individual should receive according to a non-increasing assignment function. Then select two individuals according to that assignment and alternate the worst individual with the new individual that is created after crossover.

3) Tournament selection

Choose two sets of individuals with two individuals in each set. Select the better in each set. Two children are created from crossing over the two better individuals. Finally, exchange the worst two individuals in the population with the two new individuals.

4) Stochastic tournament selection

Select the mating pool of the next generation by spinning the weighted roulette wheel and execute the tournament selection.

4.4. Crossover

The crossover that is usually employed in real-coded genetic algorithms is to crossover two collocated genes each from different parent. Each pair of genes in the same position of two parents is crossed over to generate a new string. In this paper we consider BLX $-\alpha$ crossover [14] and suggest the shift crossover operator for the fair bandwidth allocation.

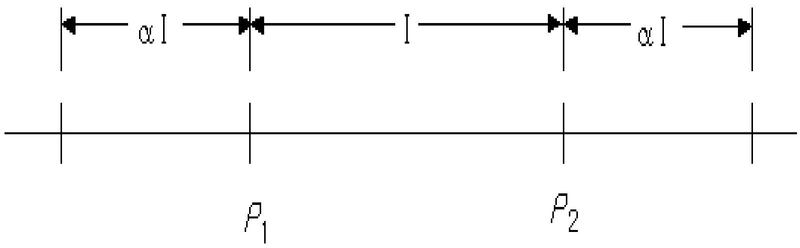

Figure 2. BLX-α crossover

1) BLX -α crossover

It uniformly picks values that lie between two genes that contain the two parents, but may extend equally on either side determined by a user specified GA-parameter α. For example, BLX-0.5 picks parameter values from points that lie on an interval that extends $I/2$ on either side of the interval I of between parents P_1 and P_2.

2) Shift crossover

Since the fairness we are interested in has the tendency to increase the bandwidth of the relatively smaller virtual sessions first, it is advantageous to shift the crossover range to the increased interval between P_1' and P_2' as shown in Figure 3. The shift crossover that is applied to relatively small virtual sessions will result in improved performance and better survival of the virtual sessions in the selection process.

4.5. Mutation

The mutation operator that allows random movement in the search space plays a secondary role in the alteration of strings. In this study we employ the mutation as an

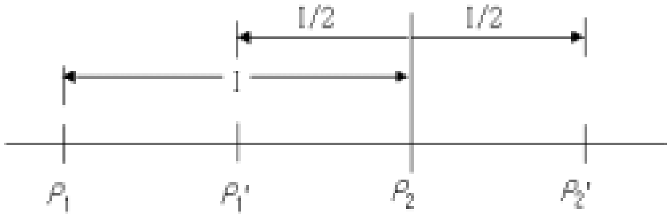

Figure 3. Shift crossover

operator that changes the bandwidth of a virtual session of a string. To maximize the effect of the mutation, the operator is allowed to move the bandwidth of a virtual session either to its maximum range or to its minimum required bandwidth. More specifically, when the capacity of each link the virtual session i traverses is fully occupied by other virtual sessions, then the bandwidth is reduced to its minimum, i.e., $x_i = u_i$. Otherwise, x_i is increased to its full extent that satisfies the capacity limit of links the virtual session traverses.

4.6. Repair procedure

When two strings are crossed over, the resultant string may be infeasible due to the increased bandwidth of a virtual session i. The total bandwidth required by all sessions traversing a link l may exceed the link capacity constraint. To have a feasible solution, a virtual session $i' \neq i$ traversing the link l is randomly selected and its bandwidth is reduced by one unit. This random selection process is repeated as far as the minimum bandwidth requirement of each virtual session is satisfied. Even with the minimum bandwidths of all other virtual sessions that traverse the link l, the feasibility of the string may not be satisfied. In this case the bandwidth of the virtual session i is adjusted to its maximum that satisfies the capacity constraint of link l.

4.7. Genetic algorithm procedure

The GA procedure employed in this paper is summarized in Figure 4. The initial population that satisfies the three constraints in Section 3 is generated with population size 100. Then the selection method is applied as in Section 4.3. The crossover in Section 4.4 is applied with crossover probability 0.7. Then the repair procedure is performed to recover the feasibility of a string as in Section 4.6. Finally, the mutation operator in Section 4.5 is applied with probability 0.01. As the termination criterion of the proposed GA, we consider

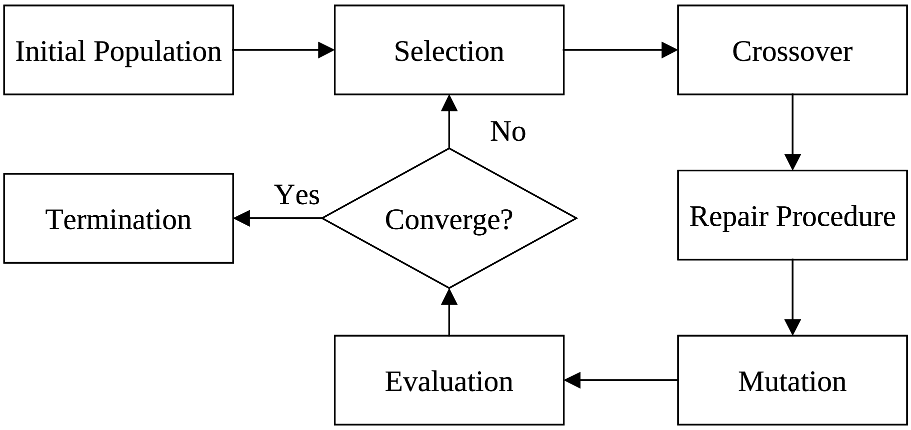

Figure 4. The Genetic Algorithm procedures

the maximum number of strings generated. When the string counter exceeds 10,000, the procedure is terminated.

5. Computational results

In this section, we discuss the computational results of the GA for the lexicographically fair bandwidth allocation. First, the four selection methods as well as the two crossover operators are compared in a network with 30 virtual sessions. Secondly, the performance of the proposed GA is investigated by comparison with the existing algorithm [7] for various problems. The algorithm presented in the previous section is implemented in Java (Jdk 1.1.3) and run on an PENTIUMⅢ-500 Intel personal computer with 256Mbytes of memory under Windows 98.

To compare the strategies for population selection and crossover methods experiments are performed with population size 100 by generating a test problem with 30 virtual sessions as in Table 1.

Table 1

Specification of test problem

Number of links	Number of sessions	Number of virtual sessions	Minimum requirement of each virtual session
17	25	30	0

In table 2 the four selection strategies are compared. The shift crossover is applied with crossover probability 0.7. Table 2 shows the fitness of string in each problem. The ranking selection demonstrates the best performance followed by the tournament, the stochastic tournament and remainder stochastic without replacement. Also the ranking selection is easier to implement than other selection procedures. Thus we apply the ranking selection in the following experiments.

Table 2

Comparison of selection strategies

Problem	Remainder	Ranking	Tournament	S-tournament
1	8.29511	7.41913	7.53073	7.42241
2	12.86472	11.55222	11.56444	11.56444
3	10.00875	7.94937	7.94937	8.67359
4	10.45594	9.60214	9.60214	10.30353
5	11.07090	10.39201	10.39764	10.39436
6	11.34844	9.28464	9.29977	9.28464
7	11.30690	10.70566	10.70801	11.43551
8	9.57221	8.10972	8.12361	8.10972
9	9.71074	9.08684	9.08450	9.08450
10	8.85235	7.69424	7.18797	7.58498

Table 3

Comparison of crossover methods

Problem	BLX-0.0	BLX-0.5	Shift crossover
1	8.02806	7.61485	7.42947
2	12.33333	12.07472	11.56444
3	8.28306	8.00485	7.94458
4	10.46194	9.78146	10.33992
5	10.55874	9.93757	10.39201
6	9.92361	9.51222	9.28464
7	11.15763	10.87291	10.70064
8	9.06250	8.38722	8.10972
9	9.65285	9.30019	9.08684
10	7.62167	7.11944	7.07636

Table 3 compares the crossover methods. The proposed shift crossover that transforms the solution range to the increased gene value performs better than the two blend crossovers. It seems mainly due to that by the shift crossover the genes with relatively smaller virtual sessions are assigned increased bandwidth and survive generation by generation with good performance.

Based on the preliminary test for the proposed GA we employ the ranking selection and the shift crossover to investigate the performance in various other problems.

Table 4

Multicast networks

Number of links	Number of sessions	Number of virtual sessions	Minimum requirement of each virtual session
7	8	10	1
15	16	20	1
17	25	30	1
31	43	50	1

Four different sizes of multicast networks are generated as in the Table 4. In each multicast network ten problems are tested with different link capacities. Each problem is run with the proposed GA and the algorithm by Sarkar and Tassiulas [7]. Since both procedures have randomness in the selection of virtual sessions to improve, each problem is run 100 times and the best solution is compared.

Note that the algorithm in [7] starts with a feasible bandwidth allocation to each virtual session. The initial feasible allocation to a virtual session is determined by the minimum requirement or the “fare share” computed by dividing each link capacity with the number of sessions. If the bandwidth of a virtual session cannot be increased due to the link capacity constraint, then the virtual session is saturated. Otherwise, to improve the fairness the procedure continues by randomly choosing a minimum virtual session and increasing the bandwidth by one unit.

Tables 5-8 show the best solution for 10, 20, 30 and 50 virtual sessions respectively. In each table the solution vector represents the nondecreasing order of bandwidth layers allocated to the virtual sessions. The corresponding fitness value is computed with $p = 2$. As illustrated in the tables the solution by GA gives better or at least equal solution compared to the algorithm by [7]. As an instance compare the solution vectors of the problem number 10 with 20 virtual sessions in Table 6. Clearly, the bandwidth assigned to the last virtual session is larger in the solution by Sarkar than by the GA. However, the solution by GA is lexicographically fairer than that by Sarkar. The same bandwidth layers are assigned to the first five smallest virtual sessions in the two procedures. But the bandwidth allocated to the sixth smallest virtual session is larger in the GA than in the Sarkar. As discussed in Section 2 the lexicographic fairness has the preference to the smallest component. If the smallest component is the same, then the second smallest is compared and the one with the larger bandwidth wins.

From Tables 5-8 notice that the proposed GA performs better in problems with large complex networks. Better solutions are obtained by the proposed GA in five problems out of ten in a network with 10 virtual sessions. The proposed GA provides lexicographically fairer solutions in problems 2, 3, 4, 5 and 9 of Table 5. In problems with 20 virtual sessions better performance is obtained by the GA in six problems: problems 3, 4, 5, 7, 8 and 10. The difference in solution quality by the two procedures becomes clear in problems with 30 and 50 virtual sessions. The proposed GA exceeds the existing algorithm in all ten problems with 30 and 50 virtual sessions.

Figure 5 shows the number of success experiments out of 100 trials to have the best solution. In the case of 50 virtual sessions, the best solution obtained by the GA is never searched even with the 100 runs of the existing algorithm in all problems. Clearly, the success ratio by the GA is higher than the algorithm by [7] in most problems. The ratio, however, is reduced as the multicast network becomes complex.

Table 5. Bandwidth allocation for 10 virtual sessions

Problem	Procedure	Solution vector	Fitness value
1	GA	(6,6,6,6,6,7,7,7,8,8)	0.2314
	Sarkar	(6,6,6,6,6,7,7,7,8,8)	0.2314
2	GA	(5,5,6,6,7,7,7,8,8,10)	0.2380
	Sarkar	(5,5,6,6,6,7,7,7,8,10)	0.2502
3	GA	(6,6,7,7,8,8,8,8,8,9)	0.1868
	Sarkar	(6,6,7,7,7,8,8,8,8,9)	0.1916
4	GA	(6,7,7,7,7,7,7,8,8,11)	0.1897
	Sarkar	(6,7,7,7,7,7,7,7,8,11)	0.1945
5	GA	(5,5,6,6,6,6,7,8,8,9)	0.2551
	Sarkar	(5,5,6,6,6,6,7,8,8,8)	0.2584
6	GA	(6,7,7,7,8,8,8,8,9,12)	0.1708
	Sarkar	(6,7,7,7,8,8,8,8,9,12)	0.1708
7	GA	(5,5,5,5,5,6,6,7,9,9)	0.3007
	Sarkar	(5,5,5,5,5,6,6,7,9,9)	0.3007
8	GA	(5,5,6,7,7,7,7,7,8,8)	0.2411
	Sarkar	(5,5,6,7,7,7,7,7,8,8)	0.2411
9	GA	(5,6,8,8,9,9,9,10,10,11)	0.1643
	Sarkar	(5,6,8,8,9,9,9,9,9,10)	0.1708
10	GA	(5,5,6,6,7,7,8,8,9,9)	0.2323
	Sarkar	(5,5,6,6,7,7,8,8,9,9)	0.2323

Table 6. Bandwidth allocation for 20 virtual sessions

Problem	Procedure	Solution vector	Fitness value
1	GA	(4,4,4,6,6,6,6,6,6,6,6,6,7,7,8,9,11,12,12)	0.5562
	Sarkar	(4,4,4,6,6,6,6,6,6,6,6,6,7,7,8,9,11,12,12)	0.5562
2	GA	(3,3,4,5,5,5,5,6,6,6,6,6,7,7,7,9,10,13,16,16)	0.6809
	Sarkar	(3,3,4,5,5,5,5,6,6,6,6,6,7,7,7,9,10,13,16,16)	0.6809
3	GA	(5,5,5,5,6,6,6,6,6,6,6,7,7,7,8,9,9,9,9,10)	0.4907
	Sarkar	(5,5,5,5,6,6,6,6,6,6,6,6,7,7,8,9,9,9,9,10)	0.4980
4	GA	(4,4,5,5,5,5,6,6,6,6,6,6,6,7,8,10,11,11,12,13)	0.5549
	Sarkar	(4,4,5,5,5,5,5,6,6,6,6,6,6,7,9,11,11,11,12,13)	0.5621
5	GA	(3,3,4,5,5,5,6,6,6,7,7,8,8,8,9,9,10,10,11,12)	0.6356
	Sarkar	(3,3,4,5,5,5,6,6,6,7,7,8,8,8,9,9,10,10,10,12)	0.6374
6	GA	(5,5,5,5,6,6,6,6,7,7,7,7,7,8,8,8,9,10,11,11)	0.4589
	Sarkar	(5,5,5,5,6,6,6,6,7,7,7,7,7,8,8,8,9,10,11,11)	0.4589
7	GA	(5,5,5,5,6,6,6,6,6,6,7,7,7,8,8,8,9,9,9,12)	0.4787
	Sarkar	(5,5,5,5,6,6,6,6,6,6,7,7,7,7,8,8,9,9,9,12)	0.4835
8	GA	(5,5,5,5,5,5,6,6,7,7,7,8,8,8,8,9,10,10,10,14)	0.4667
	Sarkar	(5,5,5,5,5,5,6,6,7,7,7,8,8,8,8,9,9,10,10,14)	0.4691
9	GA	(5,5,5,5,5,5,6,6,6,6,6,7,7,7,7,7,8,10,13,14)	0.5176
	Sarkar	(5,5,5,5,5,5,6,6,6,6,6,7,7,7,7,7,8,10,13,14)	0.5176
10	GA	(5,5,5,5,5,6,6,6,6,6,6,7,8,8,9,9,9,10,13,13)	0.4772
	Sarkar	(5,5,5,5,5,5,5,6,6,6,6,7,8,8,9,9,9,10,13,14)	0.5008

Table 7. Bandwidth allocation for 30 virtual sessions

Problem	Procedure	Solution vector	Fitness value
1	GA	(3,3,3,3,3,3,3,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,7,7,8,8,8,12,12,16)	1.4926
	Sarkar	(3,3,3,3,3,3,3,3,3,4,4,5,5,5,5,5,6,6,6,6,7,7,7,7,8,8,9,12,12,16)	1.5791
2	GA	(2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,6,7,7,7,8,9,10,11)	2.0194
	Sarkar	(2,2,2,2,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,6,7,7,7,8,9,10,11)	2.2972
3	GA	(3,3,3,3,4,4,4,4,5,5,5,5,5,6,6,6,6,6,6,7,7,7,8,8,9,9,9,9,11,13)	1.2171
	Sarkar	(3,3,3,3,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,7,7,8,8,9,9,9,9,11,13)	1.2519
4	GA	(3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,6,6,6,6,8,11,12,12,13)	1.4368
	Sarkar	(3,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,5,5,5,5,6,6,6,6,6,8,11,12,12,14)	1.5209
5	GA	(2,2,2,3,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,6,6,6,6,6,7,8,9,10,11,15)	1.9744
	Sarkar	(2,2,2,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6,7,8,9,9,11,14)	2.0935
6	GA	(2,2,2,3,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,6,8,8,8,8,9,9,13,13,13,13)	2.0678
	Sarkar	(2,2,2,3,3,3,3,3,3,4,4,4,4,4,5,5,5,5,5,5,8,8,8,8,9,10,13,13,13,13)	2.0777
7	GA	(2,3,3,3,3,3,4,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,6,6,8,8,8,11,14,15)	1.5477
	Sarkar	(2,3,3,3,3,3,3,4,4,5,5,5,5,5,5,6,6,6,6,6,6,6,7,7,8,8,8,11,14,15)	1.5816
8	GA	(2,3,3,3,3,3,3,4,4,4,4,5,5,5,6,6,6,6,7,7,7,8,8,8,9,9,9,10,11,12)	1.5681
	Sarkar	(2,3,3,3,3,3,3,4,4,4,4,5,5,6,6,6,6,7,7,7,8,8,8,8,9,9,11,11,12)	1.5922
9	GA	(2,2,3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5,5,5,6,6,6,7,7,7,9,14,16)	2.0326
	Sarkar	(2,2,3,3,3,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,6,6,6,6,7,7,7,9,14,16)	2.0690
10	GA	(2,3,3,3,3,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,5,5,7,7,7,8,9,12,12)	1.8920
	Sarkar	(2,3,3,3,3,3,3,3,3,4,4,4,4,4,4,5,5,5,5,5,5,5,5,7,7,7,9,10,12,12)	1.9313

Table 8. Bandwidth allocation for 50 virtual sessions

Problem	Procedure	Solution vector	Fitness value
1	GA	(3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5, 5,5,5,5,6,6,6,6,6,6,6,6,6,7,8,9,9,11,11,11,11,13)	2.6333
	Sarkar	(3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5, 5,5,5,5,5,6,6,6,6,6,6,6,7,7,8,9,9,10,10,11,12,13)	2.6853
2	GA	(2,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5, 5,5,5,5,5,5,5,5,6,6,7,7,7,7,8,8,8,9,11,11,11,12)	2.9754
	Sarkar	(2,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5, 5,5,5,5,5,5,5,5,5,6,7,7,7,7,8,8,8,10,11,11,11,12)	2.9852
3	GA	(3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5, 5,5,5,6,6,6,7,7,7,7,7,7,8,8,8,8,9,9,9,10,11,15)	2.4258
	Sarkar	(3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5, 5,5,5,6,6,6,7,7,7,7,7,7,8,8,8,8,9,9,9,10,10,15)	2.4275
4	GA	(2,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5, 5,5,5,6,7,7,7,7,7,8,8,8,8,8,9,10,11,11,12,13,13,14)	2.6082
	Sarkar	(2,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5, 5,5,5,6,7,7,7,7,7,7,8,8,8,8,9,10,10,10,12,12,13,14)	2.6175
5	GA	(2,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5, 5,5,5,5,5,5,6,6,6,6,7,7,7,8,9,9,9,9,9,11,12,13)	3.2939
	Sarkar	(2,2,3,3,3,3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5, 5,5,5,5,5,5,6,6,6,6,7,7,7,8,8,9,9,9,10,12,12,12)	3.2945
6	GA	(2,2,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5,5,5,6,6,6,6,6, 6,6,6,6,6,7,7,7,7,7,7,8,8,8,8,9,9,10,11,12,13,15)	2.6119
	Sarkar	(2,2,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5,5,5,5,6,6,6,6, 6,6,6,6,6,7,7,7,7,7,8,8,8,8,8,9,9,10,11,12,13,14)	2.6200
7	GA	(3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5, 5,5,6,6,6,6,6,6,7,7,8,8,8,9,9,10,10,10,10,11,11,12)	2.3850
	Sarkar	(3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5,5,5, 5,6,6,6,6,6,6,6,7,7,8,8,8,9,9,10,10,10,11,11,11,12)	2.3972
8	GA	(3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5, 5,5,5,6,6,6,6,7,7,7,8,8,8,8,9,9,10,11,11,12,13,13)	2.3612
	Sarkar	(3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5, 5,5,5,6,6,6,6,7,7,7,7,8,8,9,9,9,10,10,11,12,12,13)	2.3655
9	GA	(3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5, 5,5,6,6,6,6,6,6,6,6,6,6,7,8,8,9,10,10,11,11,14,15)	2.4754
	Sarkar	(3,3,3,3,3,3,3,3,3,3,3,4,4,4,4,4,4,4,4,5,5,5,5,5,5,5,5,5, 5,6,6,6,6,6,6,6,6,6,7,7,7,8,8,9,10,10,11,12,14,15)	2.5218
10	GA	(3,3,3,3,3,3,3,3,3,3,4,4,4,4,5,5,5,5,5,5,5,5,6,6,6,6,6,6, 6,6,6,6,7,7,7,8,8,8,8,8,9,10,10,10,10,11,11,12,14,14)	2.1843
	Sarkar	(3,3,3,3,3,3,3,3,3,3,3,3,3,4,5,5,5,5,5,5,5,5,6,6,6,6,6,6, 6,6,6,7,7,7,7,8,8,8,8,8,10,10,10,10,11,11,11,12,14,14)	2.3186

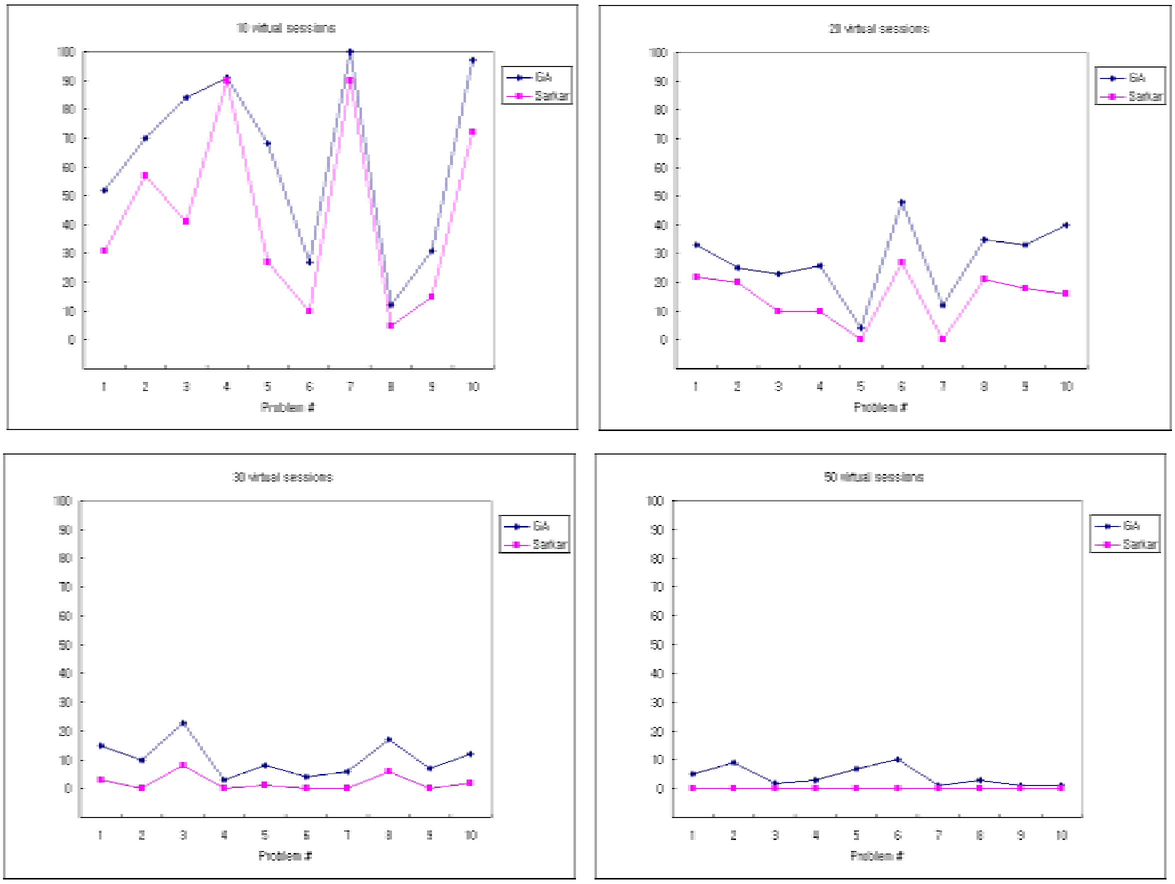

Figure 5. The number of the best solutions searched

6. Conclusion

A lexicographically fair discrete bandwidth allocation problem in multicast networks is examined. The fairness problem is formulated as a nonlinear integer programming problem which provides a lexicographically fair bandwidth allocation subject to the minimum bandwidth requirement by each virtual session, the actual bandwidth allocated to each session at each link and the link capacity constraint. As the objective function a nonincreasing convex function of the bandwidth is considered in which the sum of each component is minimized by first maximizing the smallest virtual session, then the second smallest, etc.

A genetic algorithm is developed with the fitness value based on the objective function of the nonlinear programming model. The ranking selection and the shift crossover are employed in the procedure. The proposed shift crossover transforms solution range of each gene to an increased value. Thus the shift crossover has the tendency to improve the virtual sessions with relatively small bandwidth generation by generation.

Computational experiments are performed in a multicast network with 10, 20, 30 and 50 virtual sessions. The proposed GA demonstrates outstanding performance in all problems compared to the best known existing algorithm [7]. The GA provides much better solutions and the effectiveness becomes more powerful as the network size increases. What is more noticeable is that even with 100 trials the existing algorithm never searches the best solutions found by the proposed GA in all ten problems in the network with 50 virtual sessions.

References

[1] Shannon N. Multipoint communication by hierarchically encoded data. Proceedings of IEEE INFOCOM '92, 1992. p. 2107-14.

[2] Li X, Paul S, Ammar M. Layered video multicast with retransmissions (LVMR): Evaluation of hierarchical rate control. Proceedings of IEEE INFOCOM '98, 1998. p. 1062-72.

[3] McCanne S, Jacobson V, Vetterli M. Receiver-driven layered multicast. Proceedings of ACM SIGCOMM’96, 1996. p. 117-30.

[4] Sarkar S, Tassiulas L. Fair Allocation of Resource in Multirate Multicast Trees. Global Telecom Global Telecommunications Conference GLOBECOM '99, 1999. p. 240-45.

[5] Bertsekas D, Gallager R. Data Networks. Englewood Cliffs, NJ: Prentice Hall, 1987.

[6] Sarkar S, Tassiulas L. Distributed algorithms for computation of fair rates in multirate multicast trees Proceedings of IEEE INFOCOM '00, 2000. p. 52-61.

[7] Sarkar S, Tassiulas L. Fair allocation of Discrete Bandwidth Layers in Multicast Networks. Technical research report 99-43, 1999.

[8] Sarkar S, Tassiulas L. Fair allocation of discrete bandwidth layers in multicast networks. Proceedings of IEEE INFOCOM '00, 2000. p. 1491-500.

[9] Sarkar S, Tassiulas L. Fair allocation of utilities in multirate multicast networks a framework for unifying diverse fairness objectives. IEEE Transactions on Automatic Control, 2002;47(6):931-44.

[10] Kar K, Sarkar S, Tassiulas L. A scalable low-overhead rate control algorithm for multirate multicast sessions. IEEE J. Select. Areas Commun., 2002;20(8):1541-57.

[11] Kelly F. Changing and rate control for elastic traffic. Euro. Trans. Telecommun., 1997;8:33-37.

[12] Tzeng H, Siu K. On max-min fair congestion control for multicast ABR service in ATM. IEEE J. Select. Areas Commun., 1997;15:545-56.

[13] Goldberg D, Deb K. A comparative Anaylsis of Selection Schemes Used in Genetic Algorithms. Proceedings of 5th international Conference on GAs, 1991. p. 69-93.

[14] Eshelman L, Schaffer J. Real-Coded Genetic Algorithms and Interval-Schemata. Proceedings of Foundations of GAs, 1993. p. 187-202.