Genetic Algorithm Approach and Buffering Techniques to Reduce Bandwidth Requirements for a Video on Demand System

Garima Varshney
Department of ECE
Gautam Buddha University, Greater Noida

Dr. Neeta Singh
Faculty of Department of ECE
Gautam Buddha University, Greater Noida

Abstract: Advances in computing technology have made the delivery of on-demand video service easy for every user. In this paper, we employ Genetic Algorithm (GA) and use the buffering techniques to reduce bandwidth requirement in a Video on Demand (VoD) system. Two bandwidth requirement schemes are suggested to reduce the number of streams. The scheme 1 is Join and Stream Scheme (JAS), in which the movie is broadcasted in a staggered manner and unicast streams are used. The scheme 2 is Stream Bundling Broadcasting Scheme (SBB), in which bundled channels are used to serve the user. The two multicast schemes are compared on the basis of bandwidth required i.e. optimum number of streams. The optimal batching time and a minimum number of streams required are determined using GA.

Keywords: Video on Demand, Buffering, Broadcasting, Genetic Algorithm.

I. INTRODUCTION

VoD) technology allows a subscriber to view programming in real time. There have been many developments in multimedia technology over the past few decades. They have caused VoD services to arise as a new epitome in home entertainment. These services pose challenges due to a large volume of data processing and transfer involved and strict real-time constraints. There is a necessity to efficiently use network bandwidth which demands innovative and easy schemes for scheduling continuous video streams. In paper [1] discussion on policies based on principles of broadcasting, batching, piggybacking and caching is discussed. A comparison between the various classes of scheduling policies has been made to identify key concepts. A generalized analytical approach is proposed in the paper [2], to evaluate the efficiency of various broadcasting protocols and derive bandwidth required. The paper gives a solution to achieve optimal bandwidth given client waiting time and the length of the video.

In Near-VoD users have to wait a certain period of time before the required movies are actually displayed on their TV sets after they make the request. This time is referred to as 'delay' which should be low to ensure good Quality of Service (QoS). Reducing the start-up delay causes an increase in bandwidth. Therefore, bandwidth can be reduced by providing client buffers. Buffers help the user to pre-fetch some parts of the movie. Multicast transmission is the solution to reduce the cost in large VOD systems. It is difficult to implement operations for every user in a multicast transmission system. The paper [3] proposes a multicast transmission scheme to serve a large number of users and reduce additional channels. This technique has superior performance over the multicast systems in terms of VCR blocking probability. For broadcasting a popular video multiple users should share a few channels. The paper [4] gives a new scheme called Recursive Frequency Splitting (RFS) that improves on existing schemes in terms of viewer waiting time. In paper [5], the authors have presented client buffering techniques used for scalable video broadcasting over networks with low user delay. In these techniques, clients download data from an appropriate channel and watch a movie by playback operation. To provide VoD services in multicast environment broadcasting techniques like skyscraper, patching is used. In paper [6], authors have designed an analytical model to select the most appropriate delivery scheme for transmission according to the popularity of the video and customer behavior. Results minimize overall bandwidth of the system. A video placement scheme is proposed for a VOD system with batching and multiple servers [7]. Given specified blocking probability, an optimal batching interval is derived and hybrid GA is used to obtain a corresponding placement. In [8] a new caching scheme at client's level is proposed and a scheme is introduced for better client management. By forwarding the server stream client by client, a chaining-based scheme can reduce the server streams for streaming applications in networks [9]. In this paper, an

http://dx.doi.org/10.21172/1.71.011

optimal chaining algorithm using dynamic buffer allocation strategy is proposed to utilize both backward and forward buffer. Periodic broadcasting is an efficient technique to support NVoD services for delivering popular videos. It can reduce the bandwidth needed for transmitting video to simultaneously many viewers. VoD streaming has the largest share of Internet traffic, and it relies on P2P–CDN hybrid systems [10]. This paper develops an optimal bandwidth allocation algorithm called *Demand Driven Max-Flow Formulation* and *Free-for-All Push–Lift* algorithm for hybrid VoD streaming. These techniques provide high-quality bandwidth allocation for hybrid VoD systems.

In this paper, we study two schemes for determining the optimal batching time and thereby, minimizing the average bandwidth i.e. the number of streams required for any NVoD system [11]. The analysis for these schemes is done as per the arrival rate (intermediate and high). A GA is used for determining the optimum number of streams. The paper is organized as follows: Section II gives the basics for GA technique and the toolbox used. In section III and IV model description and analysis of the two schemes, JAS and SBB is done. Section V provides the numerical illustrations for validating the analytical results and GA results. Conclusions are drawn in Section VI.

II. GENETIC ALGORITHMS

Optimizing the bandwidth for the video streaming can be done by using the conventional linear or non-linear programming approaches depending on the nature of the function. But sometimes, finding an analytical solution to such problems becomes difficult since the objective function becomes too complex to solve. Methods like GA based on evolutionary programming are used to deal with such functions. GA has various search and optimization algorithms that are based on natural genetics of humans. A combination of GA and modified bin-packing algorithm was developed for the multimedia file placement [12]. GA is used for tackling a file assignment problem for a large-scale VoD system [13]. In the paper [14], GA technique is employed to minimize the average bandwidth required in NVoD system.

The optimization Toolbox in MATLAB provides functions that can be used for finding parameters to minimize or maximize a certain objective while satisfying pre-defined constraints. The toolbox includes solvers for linear programming, quadratic programming, nonlinear least squares, mixed-integer linear programming, and nonlinear optimization. One can use these solvers to find optimal solutions to continuous and discrete problems or to incorporate optimization methods into algorithms and applications. We have used the solver for the GA to solve our objective function.

Fitness Function: A fitness function is an objective function that is used to summarize, how close a given solution is to the set aim. In the field of GA each solution is represented as a string of numbers also referred to as a chromosome. After each testing round the aim is to delete 'n' worst design solutions and breed 'n' new solutions from the best ones. Each design solution needs a figure of merit to indicate how close it came to meeting the required specification. This is generated by applying the objective function i.e. the fitness function to test and simulation results are obtained from that.

Genetic Operators: Genetic operators are used in GA to guide the algorithm towards a solution to a given problem. There are three main types of operators which must work in combination with one another in order for the algorithm to be successful.

Selection: Selection operator gives preference to better solutions, allowing them to pass on their 'genes' to the next generation of the algorithm. Best solutions are determined using the fitness function before being passed to the crossover operator. The selection operator can also pass the best solutions from the current generation, directly to the next generation without applying any mutation; this is known as *elitism* or *elitist selection*.

Crossover: In crossover technique more than one parent solutions are taken and a child solution is produced from them by recombining certain portions of good solutions. This way the GA is more likely to create a better solution. The crossover method is used when it is desired that the chromosome's representation closely matches the solution.

Mutation: Mutation operator prevents the GA converging to a local minimum by stopping the solutions from becoming too close to one another. GA can give an improved solution by mutating the solutions.

III. SCHEME DESCRIPTION AND ANALYSIS

In this paper, an arbitrary movie in the system with duration L_m minutes is considered for modeling. It is played in a staggered manner at every T_s minutes to satisfy the delay goal. Following notations are used for formulating the mathematical models of the below-mentioned schemes:

λ	Arrival rate
N_s	Number of streams required
L_m	Length of movie
D_{max}	Maximum user delay
T_s	Offset time (in minutes) or batching time for scheme JAS
B	Amount of client buffer
T_s*	Optimal Batching time
N_s*	Minimum number of streams
B*	Optimal buffer size

Scheme 1: Join-and-Stream (JAS): In this scheme, the video is multicast in a staggered manner at the regular offset of T_s minutes. If a request (by a customer) arrives in less than D_{max} minutes before the start of a multicast point, user has to wait till the start of the multicast point to 'join' the stream. If the request arrives more than D_{max} minutes before the start of the multicast point, request is served immediately with a unicast stream. The unicast stream supplies the beginning part of the video and after a short while the user starts to retrieve video data from its own buffer. For JAS, the total number of streams is the sum of multicast and unicast streams. The number of multicast streams is L_m/T_s. By conditioning on the interval between the arrival time of a request and the start of multicast point, unicast streams are calculated. The distribution f(x) of inter-arrival time is given by;

$$f(x) = \lambda e^{-\lambda x}$$

Therefore, the number of concurrent unicast stream is given by;

$$N = \int_0^{Ts-Dmax} \frac{xf(x)}{Ts}dx$$

For JAS, average number of unicast streams is;

$$Nu = \int_0^{Ts-Dmax} \frac{x\lambda e^{-\lambda x}}{Ts}dx$$

Total number of streams = Unicast + Multicast, given as;

$$N = Nu + Nm$$

This becomes,

$$Ns = \frac{1}{\lambda Ts}[1-(A+1)e^{-A}] + \frac{Lm}{Ts}$$

Where,

$$A = \lambda(Ts - Dmax)$$

By differentiating N_s with respect to T_s and equating it to zero, we get the optimal batching and a minimum number of streams. Buffer requirement (B) is given as;

$$B = Ts - Dmax$$

Scheme 2: Stream-bundling Broadcasting (SBB): This scheme groups or bundles the server streams into channels of increasing bandwidth. When a user request arrives, these high-speed bundles deliver the beginning portion of the video to the clients, so that the client can merge with an ongoing broadcast stream quickly. For stream-bundling, the total number of stream required is the sum of the broadcast/multicast streams and the streams in the bundled channels. By taking into consideration the average time for which a bundled channel is used in the movie delivery, the average bundled streams used can be calculated as,

$$\sum_{i=1}^{\frac{Ts}{Dmax}-1} \frac{iDmax}{Ts} - \frac{Ts}{2Dmax} - \frac{1}{2}$$

Therefore,

$$Ns = \frac{Lm}{Ts} + \frac{Ts}{2Dmax} - \frac{1}{2}$$

Ns achieves its minimum at,

$$Ns* = \sqrt{\frac{2\,Lm}{Dmax}} - \frac{1}{2}$$

and

$$Ts* = \sqrt{2\,Dmax\;Lm}$$

Largest buffer requirement for scheme is given as,

$$B = Ts* - Dmax$$

IV. OPTIMAL NUMBER OF STREAMS USING GA

This section covers the concept of GA for evaluating the optimal number of streams required for the two schemes. To achieve the objective to minimize the number of streams, the fitness function for GA is taken as;

$$F(x) = 1/(1 + f(x))$$

Where *f(x)* = $N_{si,}$ i=1, 2 corresponding to Scheme 1, Scheme 2 respectively. Best-fit value of the variable T_s is calculated using the GA.

V. NUMERICAL ILLUSTRATIONS

In this section, numerical illustration has been done to verify GA results with analytical results. For the two schemes, L_m is taken as 100 minutes. The optimal batching time and minimum number of streams are obtained using GA optimization tool in MATLAB with parameters as mentioned in Table 1.

Table 1: GA Parameters

GA Parameters	Value	GA Parameters	Value
Solver	GA	Mutation: Mutation function Rate	 Uniform 0.03
Problem: Fitness Function Number of variables	 $F(x) = 1/(1 + f(x))$ 1	Crossover: Crossover function	 Scattered
Constraints: Bounds	Nil Lower [1]; Upper [100]	Migration: Direction Fraction Interval	 Forward 0.2 20
Population: Type Size Creation Function	 Double vector 6 Constraint dependent	Constraint parameters: Initial penalty Penalty factor	 10 100

Fitness scaling:		Hybrid function	None
Scaling function	Rank		
Selection:		Stopping criteria:	
Selection function	Stochastic uniform	Generations	50
		Time limit	Inf
		Fitness limit	-Inf
		Stall generations	50
		Stall time limit	Inf
		Function tolerance	1e-6
		Nonlinear constraint tolerance	1e-6
Reproduction:			
Elite count	2		
Crossover fraction	1.0		

Table 2 gives values obtained through analytic and GA optimization method with varying D_{max} for both the schemes (taking $\lambda = 10$/min). Batching time increases for both schemes as delay increases, which implies that low start-up delay requires lesser batching time and high start-up delay requires larger batching time. Table 3, gives values obtained through analytic and GA optimization method with varying λ for Scheme 1 and Scheme 2. Batching time is higher for high arrival rates for scheme 1. Since batching time for scheme 2 is independent of arrival rate, its value remains constant throughout.

Table 2: Optimal Batching Time for Scheme 1 and Scheme 2 with varying D_{max} ($\lambda = 10$/min)

D_{max}	Scheme 1		Scheme 2	
	T_s* (Analytical)	T_s* (GA)	T_s* (Analytical)	T_s* (GA)
10	10.01116	10.12	44.72136	44.72
15	15.00465	15.65	54.77226	54.77
20	20.00256	20.49	63.24555	63.86
25	25.00163	25.76	70.71068	72.08
30	30.00112	30.33	77.45967	77.46
35	35.00082	35.88	83.66600	83.71
40	40.00063	40.25	89.44272	89.44
45	45.00050	45.21	94.8683	94.87
50	50.00040	50.54	100	100
55	55.00033	55.79	104.8809	104
60	60.79909	60.4	109.5445	109.5

Table 3: Optimal Batching Time for Scheme 1 and Scheme 2 with varying λ ($D_{max} = 2$ min)

λ	Scheme 1		Scheme 2	
	T_s* (Analytical)	T_s* (GA)	T_s* (Analytical)	T_s* (GA)
15	1.890845	1.89	20	19.8001
20	1.911698	1.912	20	19.8001
25	1.926936	1.925	20	19.8001
30	1.937929	1.926	20	19.8001
35	1.946128	1.945	20	19.8001
40	1.952447	1.952	20	19.8001
45	1.957455	1.957	20	19.8001
50	1.960768	1.96	20	19.8001
55	1.964253	1.964	20	19.8001
60	1.967700	1.967	20	19.8001

Fig.1 shows the variation of T_s* with respect to D_{max} for scheme 1 and scheme 2. It is clear that as delay increases batching time required for JAS is less than that for SBB scheme. In fig. 2, the optimum buffer requirement (B*) for both the schemes is shown by varying D_{max}. SBB requires more buffer than JAS throughout. Buffer requirement of SBB scheme first increases with increasing delay rates and then it decreases for very high D_{max}. Fig. 3 represents the optimal number of streams (N_s*) required for varying values of $D_{max.}$ N_s* decreases as D_{max} increases for both the schemes. It is noted that for lower delay rate, the optimal stream requirement of JAS is more than that of SBB. For higher D_{max}, both schemes require similar number of streams. Clearly, scheme 2 is better in terms of bandwidth required for low delay rates. In case of high delay rates any of the two schemes are equally suitable. Fig.4 represents optimal batching time for varying arrival rate for both scheme 1 and scheme 2. Batching time required for SBB is more than for JAS. Fig. 5 shows the buffer requirement for JAS is less than SBB for variable arrival rates, given D_{max}. Fig. 6 represents the optimal number of streams required for both schemes with varying arrival rates. Clearly, JAS requires more streams than SBB with a little user delay proving the latter to be a better scheme. Overall, for higher arrival rates, SBB proves to be more bandwidth efficient since it requires lesser number of streams than JAS. The GA results are quite closer to the analytical results for all T_s, B* and N_s* which verifies our results. Fig. 7 and Fig. 9 illustrate the fitness value for the fitness function of N_s in each generation of the GA for scheme 1 and scheme 2 respectively. In fig. 8 and fig. 10, best, worst and mean scores have been calculated for scheme 1 and scheme 2 respectively using each generation of the fitness function for number of streams.

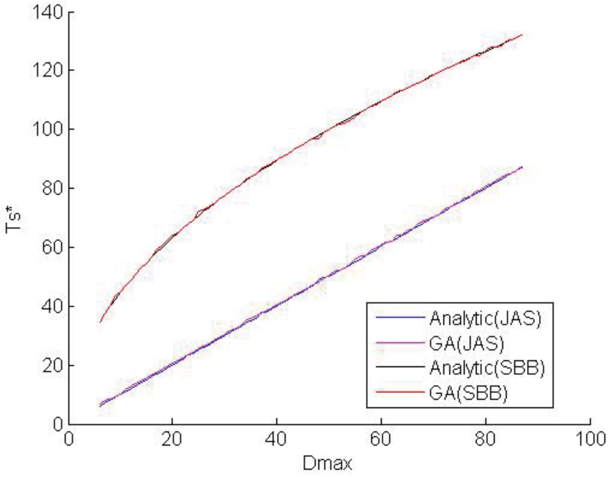

Fig. 1: T_s* by varying D_{max} for both schemes

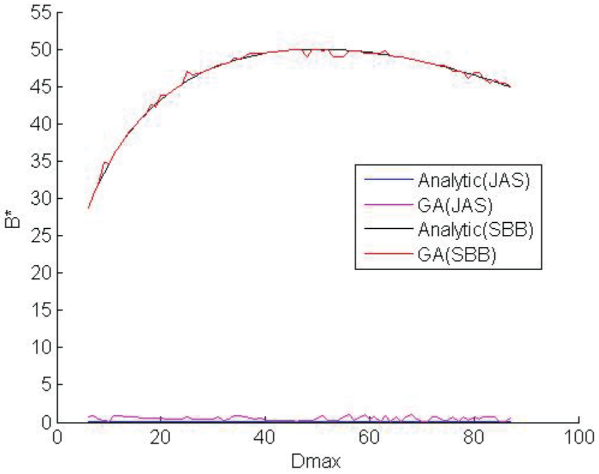

Fig. 2: B* by varying D_{max} for both schemes

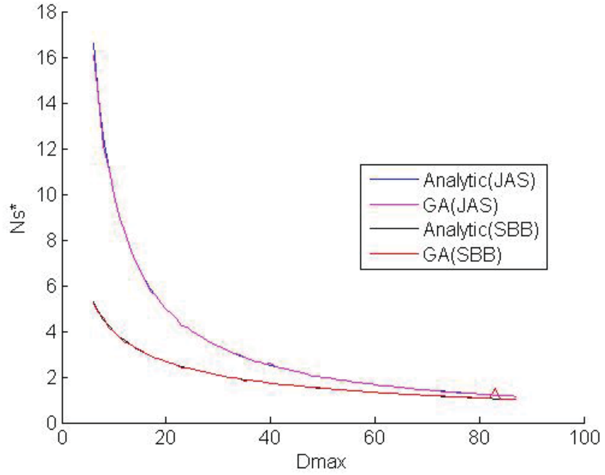

Fig. 3: N_s* by varying D_{max} for both schemes

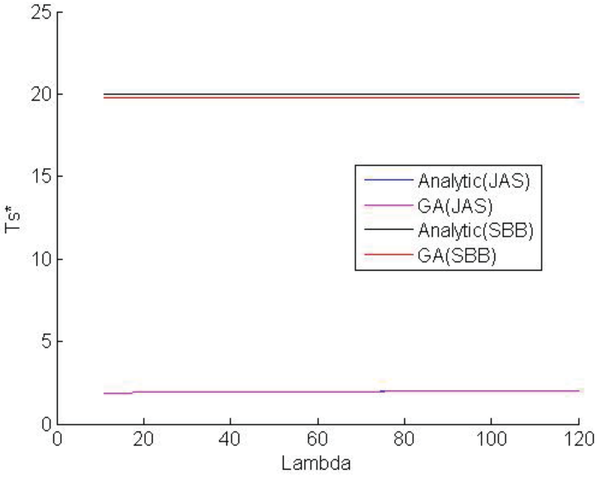

Fig. 4: T_s* by varying λ for both schemes

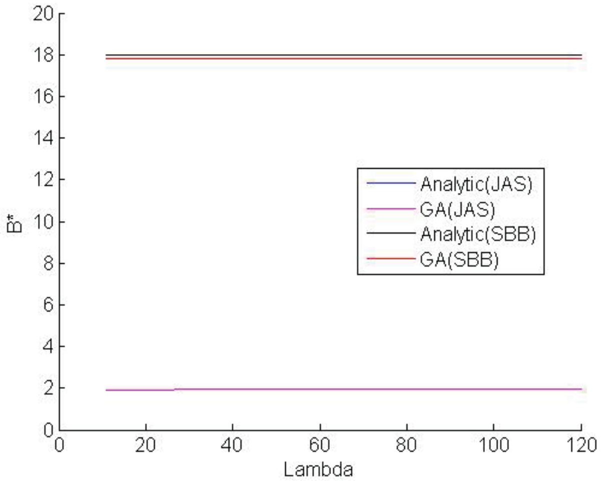

Fig. 5: B* by varying λ for both schemes

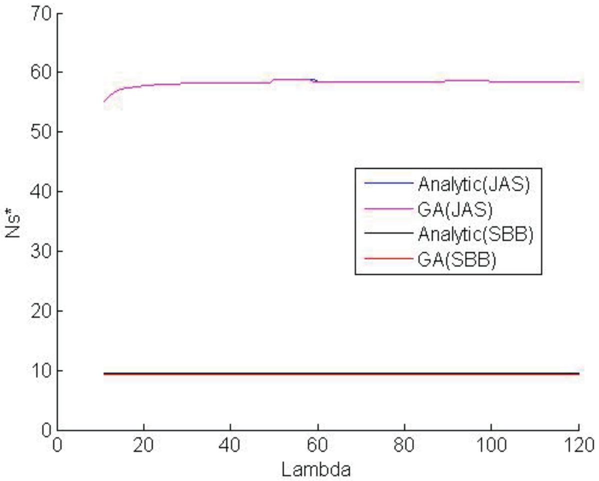

Fig. 6: N_s* by varying λ for both schemes

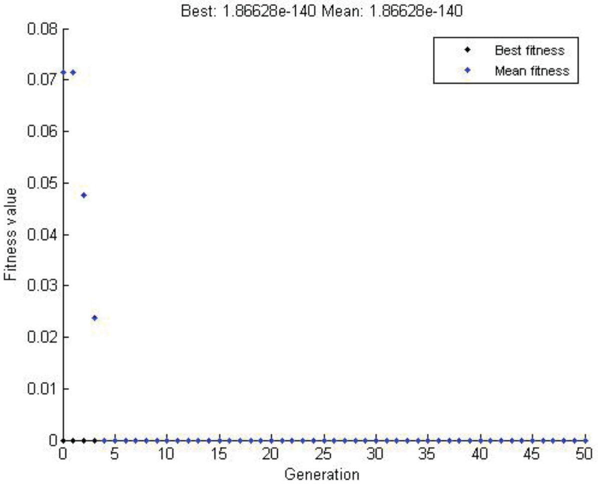

Fig. 7: Fitness value for N_s for JAS

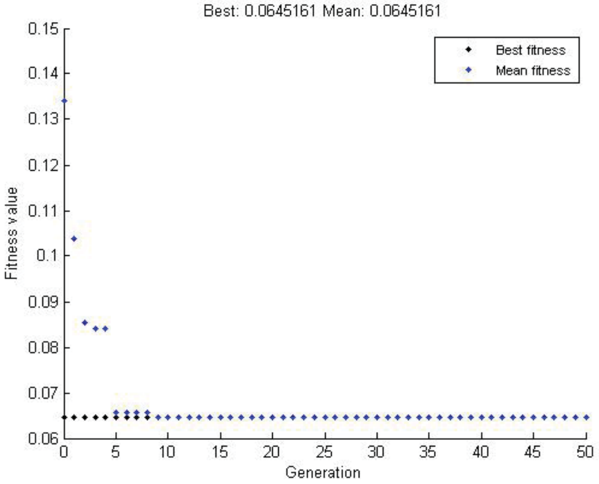

Fig. 9: Fitness value for N_s for SBB

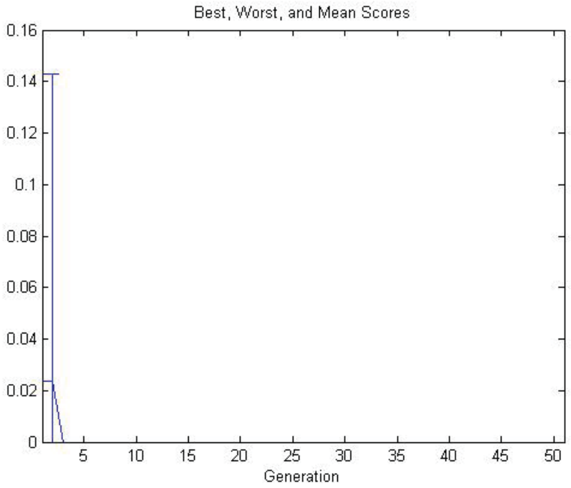

Fig. 8: Best and worst value for N_s for JAS

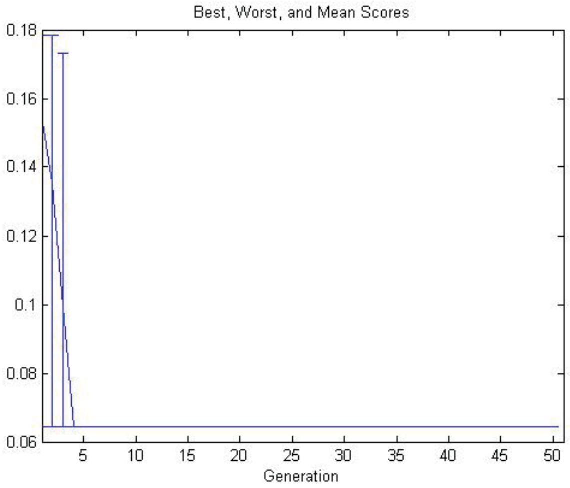

Fig. 10: Best, worst and mean value for N_s for SBB

VI. CONCLUSION

In this paper, the problem of minimizing bandwidth or the number of streams required in an NVoD system is studied. Two multicast schemes are proposed and the optimal batching time for these schemes has been determined using GA. The broadcasting schemes are compared on the basis of number of streams required for a certain maximum user delay and arrival rate. The optimal batching time was higher for scheme 2 in all cases as compared to scheme 1. The number of streams required for scheme 1 is more than scheme 2, which makes scheme 2 more bandwidth efficient. Buffer requirement of scheme 2 is higher than scheme 1. The scheme 1, JAS seems to be appropriate for movies of intermediate arrival rate, whereas scheme 2, SBB is more appropriate for movies with high request rate. GA provides very accurate results for the optimal batching time for both schemes.

REFERENCES

[1] Ghose, D., & Kim, H. J. (2000): Scheduling video streams in video-on-demand systems: A survey. *Multimedia Tools and Applications*, *11*(2), 167-195.

[2] Hu, A. (2001): Video-on-demand broadcasting protocols: A comprehensive study. In *INFOCOM 2001. Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, 1, 508-517.

[3] Choi, J. M., Lee, S. W., & Chung, K. D. (2001): A multicast delivery scheme for VCR operations in a large VOD system. In *Parallel and Distributed Systems, 2001. ICPADS 2001. Proceedings. Eighth International Conference,* 555-561.

[4] Tseng, Y. C., Yang, M. H., & Chang, C. H. (2002): A recursive frequency-splitting scheme for broadcasting hot videos in VOD service.*Communications, IEEE Transactions on*, *50*(8), 1348-1355.

[5] Chan, S. G. H., & Yeung, S. I. H. (2002): Client buffering techniques for scalable video broadcasting over broadband networks with low user delay.*Broadcasting, IEEE Transactions on*, *48*(1), 19-26.

[6] Poon, W. F., Lo, K. T., & Feng, J. (2003): Determination of efficient transmission scheme for video-on-demand (VoD) services. *Circuits and Systems for Video Technology, IEEE Transactions on*, *13*(2), 188-192.

[7] Tang, W. K., Wong, E. W., Chan, S., & Ko, K. T. (2004): Optimal video placement scheme for batching VOD services. *Broadcasting, IEEE transactions on*, *50*(1), 16-25.

[8] Do, T. T., Hua, K. A., & Tantaoui, M. A. (2004): P2VoD: Providing fault tolerant video-on-demand streaming in peer-to-peer environment. In*Communications, 2004 IEEE International Conferenc,* 3, 1467-1472.

[9] Su, T. C., Huang, S. Y., Chan, C. L., & Wang, J. S. (2005). Optimal chaining scheme for video-on-demand applications on collaborative networks.*Multimedia, IEEE Transactions on*, *7*(5), 972-980.

[10] Tian, C., Sun, J., Wu, W., & Luo, Y. (2015). Optimal bandwidth allocation for hybrid Video-on-Demand streaming with a distributed max flow algorithm.*Computer Networks*, *91*, 483-494.

[11] Jain, M. and Singh, N. (2011): Buffering Schemes for Video Server in Video-on-Demand (VoD) System with Low User Delay. *International Journal of Information and Computing Science*, *13*(2), 57-65.

[12] Tang, K. S., Ko, K. T., Chan, S., & Wong, E. W. (2001). Optimal file placement in VOD system using genetic algorithm. *Industrial Electronics, IEEE Transactions on*, *48*(5), 891-897.

[13] Tang, W. K., Wong, E. W., Chan, S., & Ko, K. T. (2004). Optimal video placement scheme for batching VOD services. *Broadcasting, IEEE transactions on*, *50*(1), 16-25.

[14] Gupta, K. P.(2014). Genetic Algorithm Approach for Bandwidth Optimization in Near Video on Demand System. ***International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 7430-7438***