MULTI-OBJECTIVE OPTIMIZATION OF MULTIMEDIA PACKET SCHEDULING FOR AD HOC NETWORKS THROUGH HYBRIDIZED GENETIC ALGORITHM

R.Muthu Selvi and R.Rajaram

Department of Information Technology, Thiagarajar College of Engineering, Madurai, Tamilnadu, India
rmsit@tce.edu
rrajaram@tce.edu

ABSTRACT

This paper presents a new approach to optimize the schedule of the variable length multimedia packets in Ad hoc networks using hybridized Genetic Algorithm (HGA).Existing algorithm called Virtual Deadline Scheduling (VDS) attempts to guarantee m out of k job instances (consecutive packets in a real-time stream) by their deadlines are serviced. VDS is capable of generating a feasible window constrained schedule that utilizes 100% of resources. However, when VDS either services a packet or switches to a new request period, it must update the corresponding virtual deadline. Updating the service constraints is a bottleneck for the algorithm which increases the time complexity. HGA overcomes the problem of updating the service constraints that leads to the increased time complexity. The packet length and the number of packets to be serviced are the two conflicting criteria which are affecting the throughput of scheduling. Using HGA, a trade off can be achieved between the packet length and the number of packets to be serviced. HGA produces an optimized schedule for the multimedia packets. Journals.

KEYWORDS

Hybrid genetic algorithm, window- constrained scheduling, multiobjective optimization, multimedia streaming, variable length packets

1. INTRODUCTION

Recently, we are able to see an increasing interest in the field of multimedia. Multimedia applications include Video-on-demand, video authoring tools, news broadcasting, video conferencing, digital libraries and interactive video games. In these applications, the presence of many consecutive non-ordered multimedia packets in video and audio streams sent over a network might result in significant delay while buffering and receiving it. Hence, arranging the packets efficiently is necessary and it can be done by window-constrained scheduling. Window-constrained scheduling is an algorithm in which a minimum number of consecutive job instances (e.g., periodic tasks or consecutive packets in a real-time stream) must be processed by their deadlines in every finite window.

Existing algorithm, Virtual Deadline Scheduling (VDS) guarantees resource sharing to a specific fraction of all job instances, even when resources are 100% utilized and request periods differ between jobs. That is, two jobs J_i and J_j, may have different request periods, T_i and T_j. It is also able to limit the extent to which a fraction of all job instances are serviced late. However, if the service times of all the jobs are not same and the request period is not a multiple of service time

DOI : 10.5121/ijma.2011.3309

then VDS is not capable of producing a feasible schedule. Updating the service constraints is a major bottleneck in VDS.

Due to the fast convergent property of Multi-objective Genetic Algorithm (MOGA), it is possible to schedule the variable length packets efficiently in high speed networks. However, MOGA sometimes trapped with local optima and the result is affected. Hence, we hybridize MOGA with the local search technique, Simulated Annealing (SA) and schedule the packets in a multimedia stream. In this work, we use MOGA and HGA for optimal scheduling of packets. The length of the packets and the number of packets to be serviced are used as schedule parameters. They are also the conflicting objectives. The results are compared with the results of VDS and MOGA. HGA schedules the packets without violating the window-constraint. It is found that hybrid Genetic algorithm (HGA) outperforms both algorithms.

This paper is organized as follows: In Section 2, the related work is discussed. In section 3, problem is defined. Section 4 briefs the methodology MOGA and HGA. Section 5 presents the comparison results. Section 6 concludes.

2. RELATED WORK

Window-constrained scheduling is a form of weakly-hard [1, 2] service, in which a minimum number of consecutive job instances (e.g., periodic tasks or consecutive packets in a real-time stream) must be processed by their deadlines in every finite window

Scheduling a given number of variable length packets effectively in order to achieve a maximal network throughput i.e. Optimal Packet Scheduling is well known as a NP-hard problem [3].

West et al [4, 5, 6] have developed Dynamic Window Constrained Algorithm (DWCS) which attempts to guarantee no more than k out of a fixed window of m deadlines are missed for consecutive job instances as long as the total utilization of all required job instances does not exceed 100%.However, DWCS is only capable of guaranteeing a feasible schedule when all jobs have the same request period.

West et al [7] have improved DWCS and developed VDS. VDS guarantees to service m out of k job instances by their virtual deadlines that may be some finite time after the corresponding real-time deadlines. Notwithstanding, VDS is capable of outperforming DWCS and similar algorithms, when servicing jobs with potentially different request periods. Additionally, VDS is able to limit the extent to which a fraction of all job instances are serviced late. Unlike DWCS, VDS is capable of scheduling jobs with different request periods.

Both, Low bit rate cellular systems and high bit rate local area networks are more prone to bit errors. The quality degradation is observable even at the transport and application layers in the form of increased packet loss. Specifically for the applications like multimedia streaming where timely delivery of data is required, both effects are harmful. Nabeel A. Al-Saber et al [8] have shown that large packets are more likely than small packets to be discarded due to bit errors. On the other hand, small packets lead to higher proportional protocol header overhead. Therefore packet size optimization is an essential research problem in Ad hoc Networks.

Jari Korhonen and Ye Wang [9] have shown that application level packet size optimization could facilitate efficient usage of wireless network resources, improving the service provided to all end users sharing the network.

Dragao Savic [11] has shown the comparison between single objective and multi- objective optimization for integrated decision support systems.

Hisao Ishibuchi and Tadashi Yoshid [12] examined how the search ability of evolutionary multi-objective optimization algorithms can be improved by the hybridization with local search through computational experiments on multi-objective permutation flow shop scheduling problems.
Malek Rahoual and Rachid Saad [13] presented a hybrid of two metaheuristics (genetic algorithm and tabu search) to tackle the timetabling problem and got promising experimental results. Fang-Chih Tien Kuang-Han Hsieh and Chi-Shuan Liu [14] applied HGA to solve discrete location allocation problems with rectilinear distance. The performance was evaluated relative to the Simulated Annealing method. HGA was excellent in quality of solutions and speed of computation.

3. PROBLEM FORMULATION

In this paper, to formulate the problem, we consider P_n packet streams n={1,2,3,....n} with an objective of maximizing the lengths of the packets to be serviced and maximizing the number of packets scheduled by their deadline and to minimization of packet arrival time

The objectives are formulated as follows.

$$f1 = \sum_{i=1}^{n} \max\ k1 t_{ai} \qquad (1)$$

$$f2 = \sum_{i=1}^{n} \max\ k2 L_i \qquad (2)$$

$$f3 = \sum_{i=1}^{n} \max\ k3(k_i - m_i) \qquad (3)$$

L: The packet length

N: Number of packets

t_a: The arrival time of the packet

K_i -m_i: *T*he maximum number of packets to be serviced

k1: A positive constant, ranges between 0 and 5

k2: 66.67% of k1, k3: 7% of k1

4. METHODOLOGY

4.1 HGA for window-constrained scheduling to optimize multiple objectives

4.1.1 Representation/Encoding Scheme

P_1	P_1	P_3	P_1	P_1	P_3	P_2	P_2	P_3	P_2	P_2	P_3
1	2	3	4	5	6	7	8	9	10	11	12

↑Packet instances with ids ↑Gene

Figure 1. Gene representation of an individual.

A chromosome is an array of *n* integers where *n* is the total number of packets to be scheduled. The allele value at the *ith* entry of a chromosome represents one packet instance in a stream. The chromosome represents the packet sequence in which the various packets are to be processed. The representation is given in Figure 1.

4.1.2 Initialization

The selected the population size is 200. The chromosomes (i.e.) various sequences of packet streams are randomly generated. The candidate solutions in the population have a direct representation. The size of each chromosome will be equal to the window size.

4.1.3 Fitness Function

The fitness function used to evaluate individuals is the function given by the equation (4)

.

$$f(L,k,m) = \max\{\mathrm{k1t}_{ai} + \mathrm{k2L_i} + \mathrm{k3}(ki - mi)\} \qquad (4)$$

4.1.4 Crossover

Crossover operator selects two chromosomes at a time and generates offspring by combining both individual's features. A simple way to achieve crossover is to choose a random cut-point and generate the offspring by combining the segment of one parent to the left of the cut-point with the segment of the other parent to the right of the cut-point.

4.1.5 Mutation

Mutation is usually used as a background operator, which produces spontaneous random changes in various individuals. A simple way to achieve mutation would be to alter one or more genes

4.1.6 Elitism

Elitism is the process of selecting the best chromosomes in a particular generation and retaining them unaffected for the next generation. This is done to overcome the loss of best chromosomes due to the process of cross-over and mutation. The elitism rate that we have chosen is 0.1%.

4.1.7 Replacement

The new population generated by the previous steps, update the old population.

4.1.8 Stopping criterion

If the number of generations equals to the pre-specified number then stop, otherwise the above process is repeated. In this work, the process is repeated for 30 generations.

4.1.9 HGA

To develop the hybrid genetic algorithm, we have combined features of Simulated Annealing (SA) to the basic Multiobjective GA (MOGA) work. The capability of SA for selecting the fittest candidate solutions are used as input to the cross-over module. Though SA takes some time to cool down to the equilibrium state, it eliminates the dependency of the selection process on a complete pool of candidate solutions required in conventional method at the selection stage. Both SA and MOGA are randomized guided search methods which when combined result in HGA.

4.1.10 Simulated Annealing

Simulated annealing, like genetic algorithm, is an optimization procedure that performs randomized search in large, complex and multimodal search space for providing a near optimal solution.

In SA a problem state is defined by the values of a number of parameters. The state transition is done by changing the values of the parameters using the Boltzmann distribution function in thermodynamics. The objective is to maximize the value of our objective function. At each state transition the temperature of the system is reduced by a small amount. The temperature schedule is designed so that the state of the system freezes after hundreds of transitions. A logarithmically decreasing temperature is found useful for convergence without getting stuck to a local maximum state.

At each iteration of the process, the temperature T is reduced by a small amount. The whole process is repeated until a near optimal solution is found. A set of control parameters govern the convergence of the algorithm. These parameters are initial temperature T_0, a decrement function for decreasing the value of T at every iteration, a lower limit for T or the maximum number of iterations needed for convergence.

```
begin
g = 0
initialize (T, P (g))
evaluate P (g) using fitness function
fmax maximum fitness of P (g)
termination_condition = false
while (NOT termination_condition) do
begin
g = g + 1
for i = 1 to N do
begin
if fmax-f(xj)<=0
then select xj from P (g) and set fmax to f (xj)
else if (exp [-(f(y)-f(x)) / T ] >= random [0, 1] )
then select xj from P (g)
else select x corresponding to fmax
end
crossover
mutation
evaluate P (g + 1) using fitness function
lower T
end
end
```

Figure 2. The Proposed Hybrid algorithm

In the new method of selection, called stochastic selection a chromosome with a value xi is considered from a pool P (g) of generation g and is selected based on Boltzmann probability distribution function. Let it be assumed that fmax is the fitness of the currently available best chromosome. If the next chromosome has fitness f (x) such that it is greater than fmax, then the new chromosome is selected otherwise it is selected with Boltzmann probability

$$P(\exp\left[-(f(y)-f(x))/T\right]) \qquad (6)$$

where $T = T_0\,(1-\alpha)^k$ *and* $K = (1+(g/G)*100)$

'g' is the current generation number; G, the maximum value of g. The value of α can be chosen from the range [0, 1], and T_0 from the range [5,100]. The above equation shows that the value of T decreases exponentially or at logarithmic rate with increase in the value of g and hence the value of the probability P. This is significant in terms of convergence. The final state is reached when computation approaches zero value of T, i.e., the global solution is achieved at this point. In the proposed HGA algorithm, the probability that the best string is selected and automatically included as a member of the selected population is very high. However, elitism is suggested to eliminate the chance of any undesired loss of information during the mutation stage.

The proposed approach is given in Figure 2.By the above procedure for HGA better scheduling patterns can be arrived.

5. RESULTS AND DISCUSSION

5.1. Experimental Setup

We collected the packets from a multimedia presentation using Wireshark 1.2.1. Wireshark is a network packet analyzer which analyses the real-time network packets and gives information about the packets such as protocol, header information, packet length, arrival time of the packet etc. We used the packet length and arrival time of the packet from Wireshark.

200 packet sets are taken as the initial population, each consisting of various sequences with many packet instances. Elitism rate is selected as 0.05%. The single point crossover has been chosen. The crossover rate is 0.9. The uniform mutation with a low probability (0.01) is used. For HGA, initial temperature is set as 30 and α value is 0.5. 30 generations are generated.

The results are tabulated in table 2. In each test case, and the percentage of resource utilization, percentage of average delay, percentage of average context switches and percentage of violations per task are found and HGA and micro-GA are compared to VDS.

5.2. Empirical Results

5.2.1. Violation of window-constraint

In window-constrained scheduling, the packet streams are defined by a tuple (Ci, Ti, mi, ki).C

denotes the service time, T denotes the request period, m denotes the minimum number of packets to be serviced by their deadline and k denotes the total packets to be serviced. For some packet instances, there is a violation of window-constraints (i.e.) (Servicing the various packet instances within the deadline) in the case of VDS whereas the schedule produced by MOGA and HGA, there is no such violation of window-constraint.

According to the window- constraint for P2 in Table 1, out of 9 request periods of P2 at least 5 request periods have to be satisfied. Figure 3 shows that VDS could not handle the processes with varying service times and request period, Only 4 request periods are serviced and the window-constraint of J2 is violated. Figures 4 and 5 show the result of scheduling using MOGA and HGA respectively. Here the window-constraints of both the packets are satisfied.

Table 1. Window-constraints for packets

Packet	**C**	**T**	**m**	**K**
P1	2	3	2	3
P2	1	1	5	9

P2	P2	P1	P2	P2	P1	P1	P1	P1

↓P2 violates

Figure 3 Schedule produced by VDS.

P2	P1	P1	P1	P1	P2	P2	P2	P2

Figure 4 Schedule produced by MOGA

P2	P1	P1	P1	P1	P2	P2	P2	P2

Figure 5 Schedule produced by HGA

5.2.2. Average waiting time of packets

We have tabulated the average waiting time of packets for VDS, MOGA and HGA in Table 2. The average waiting time of packets scheduled using HGA is very less than that of VDS and MOGA. Since the throughput depends on the average waiting time, the performance is improved when we use HGA for scheduling the packets.

Table 2 Average waiting times of packets for VDS, MOGA and HGA

Number of Packets	**Average waiting time of packets (using VDS) (ms)**	**Average waiting time of packets (using MOGA) (ms)**	**Average waiting time of packets (using HGA) (ms)**
5	3.6	1.6	1.3
10	4.2	2.2	1.8
15	4.9	1.9	1.8
20	5.1	2.3	2.2

The comparison is shown in Figure 6. As the number of increases, the packets have to wait more in the queue and throughput is decreased.

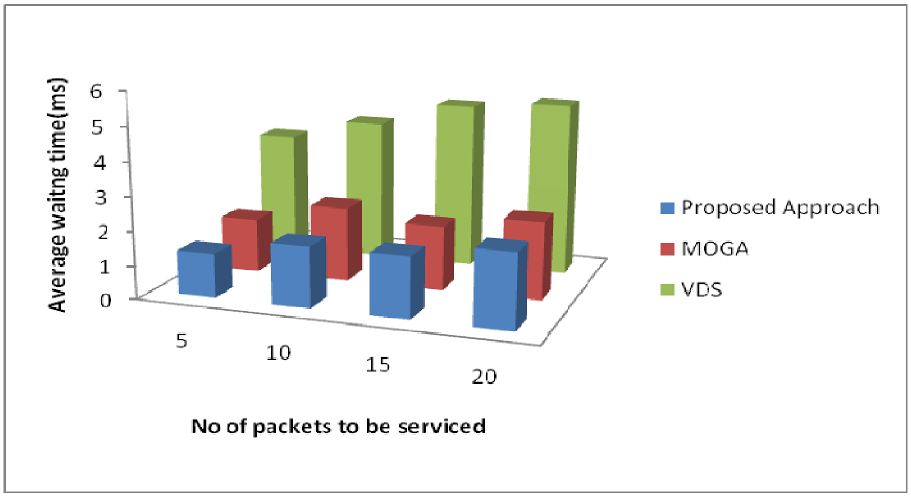

Figure 6. Comparison of average waiting times of packets for VDS, MOGA and HGA

5.2.3. Missed deadlines

The missed deadlines are found out for each schedule by using the values ki and mi. i.e. ki-mi. From the results, it can be found out that minimal number deadlines are missed in HGA. However, as the number of packets increases, HGA also misses more packets in the packet stream.. The results are tabulated in Table 3.

Table 3 Missed deadlines of packets for VDS, MOGA and HGA

Number of Packets	Missed deadlines (using VDS)	Missed deadlines (using MOGA)	Missed deadlines (using HGA)
5	6	5	3
10	8	7	4
15	8	9	6
20	12	10	11

Figure 7 shows the comparison of missed deadline of packets for VDS, MOGA and HGA

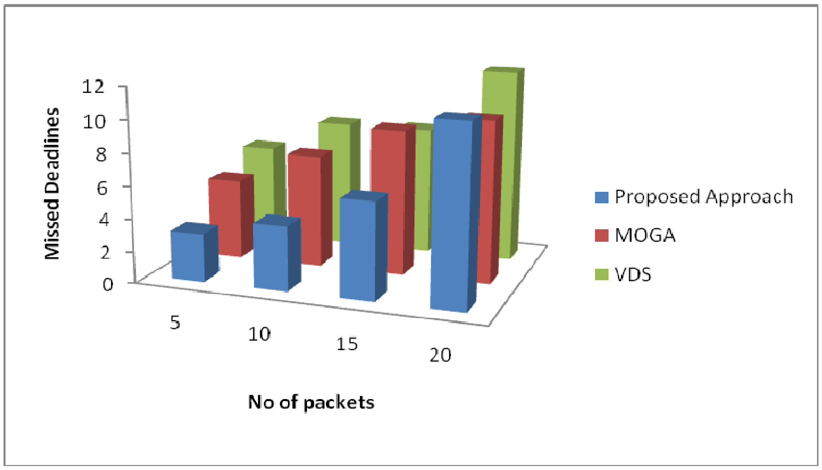

Figure 7. Comparison of missed deadline of packets for VDS, MOGA and HGA

6. Conclusion

Though the objectives variable packet length and the minimum number of packets that must be serviced for a given window are the conflicting objectives, HGA is able to find an optimal solution and it produces an optimized scheduling pattern. Updating the service constraints repeatedly is a major bottleneck in VDS. But this difficulty is overcome by HGA. Even when the service time is not a multiple of request period, HGA is able to produce a schedule without violating the window-constraint. HGA also outperforms MOGA in reducing the average waiting time and number of missed deadlines of packets. When scheduling the packets minimizing the average waiting time and minimizing the number of missed deadlines is very crucial for multimedia applications. Hence, HGA is suitable for scheduling multimedia packets.

References

[1] G.Bernat, A.Burns,A.Llamosi, (2001) "Weakly-hard real-time systems", IEEE Transactions on Computers, 50(4):308–321

[2] G.Bernat & R.Cayssials, (2001) "Guaranteed on-line weakly-hard real-time systems", Proceedings of the 22nd IEEE Real-Time Systems Symposium. pp. 25-35

[3] Ron-gu waa, Yang-lee, Shian Tsong shu, Hsian-wei Tseng, Ming-Huseh Chuang, Yuang haung waung, (2008) "Application of hardware Architecture of Genetic Algorithm for optimal packet scheduling", International Journal of Fuzzy Systems, Vol. 10, No. 3, pp 202-206

[4] R. West, C. Poellabauer, (2000) "Analysis of a window constrained scheduler for real-time and best-effort packet streams", Proceedings of the 21st IEEE Real-Time Systems Symposium 239 - 248

[5] R. West, K. Schwan, C. Poellabauer, (1999) "Scalable scheduling support for loss and delay constrained media streams", Proceedings of the 5th IEEE Real-Time Technology and Applications Symposium, RTAS '99, pp. 24-35

[6] R.West, Y. Zhang, K. Schwan, C. Poellabauer, (2004) "Dynamic window-constrained scheduling of real-time streams in media servers", IEEE Transactions on Computers, Vol. 53, No. 6, pp 744 – 758

[7] Y. Zhang, R. West, X. Qi, (2004) "A virtual deadline scheduler for window constrained service guarantees", 25th IEEE International Real-Time Systems Symposium, pp:151-160

[8] Nabeel A. Al-Saber, Saurabh Oberoi, Roberto Rojas-Cessa, Sotirios G.Ziavras, (2009) “Concatenating Packets in Variable-Length Input-Queued Packet Switches with Cell-Based and Packet-Based Scheduling” Proceedings of the 32nd international conference on Sarnoff symposium

[9] J. Korhonen, Y. Wang, (2005) “Effect of Packet Size on Loss Rate and Delay in Wireless Links”, Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC ’05), pp. 1608-1613

[10] Dragao Savic, (2002) “Single-objective vs. Multi-objective Optimization for Integrated Decision Support”, Proceedings of the First Biennial Meeting of the International Environmental Modeling and Software Society,7-12

[11] A H.Ishibuchi and T.Yoshida, (2002), “Hybrid Evolutionary Multi-Objective Optimization Algorithms”, in A. Abraham et al. (eds.) Soft Computing Systems: Design, Management and Applications: Frontiers in Artificial Intelligence and Applications (Vol. 87), , pp.163-172.

[12] Malek Rahoual, Rachid Saad (2006)“Solving Timetabling Problems by Hybridizing Genetic Algorithms and Tabu Search” PATAT 2006, pp. 467–472.

[13] Fang-Chih Tien, Kuang-Han Hsieh, Chi-Shuan Liu (2007) “Using Hybrid Genetic Algorithms to solve discrete location allocation problems with rectilinear distance” Journal of the Chinese Institute of Industrial Engineers, Vol. 24, No. 1, pp. 1-19

Authors

R.Muthu Selvi, Assistant Professor, Dept of IT has BE degree in Electronics and Communication Engineering from Madurai Kamaraj University in 1989. She secured the ME degree in Computer Science and Engineering in 2002 from Madurai Kamaraj University. She is pursuing PhD in Information and Communication Engineering from Anna University, Chennai. Her areas of interest are Operating Systems, Multicore and Genetic Algorithms

Dr. Rajaram Ramasamy**,** Professor& Head**,** Department of IT, Thiagarajar College of Engineering, has BE degree in Electrical and Electronics Engineering from Madras University in 1966. He secured the M Tech degree in Electrical Power Systems Engineering in 1971 from IIT Kharagpur and PhD degree on Energy Optimization from Madurai Kamaraj University in 1979. He and his research scholars have published more than 45 research papers in Journals and Conferences. Nine of his scholars secured the Ph.D. degree in Computer Science and Communication. His current areas of interest are Cryptography, Data mining and Genetic Algorithms. He has published more than 13 books on Computer language and Basic communications. He has served the Makerere University at Uganda during 1977 –78 and University of Mosul during 1980 -81. He secured two best technical paper awards from the Institution of Engineers, India and one from Indian Society for Technical Education.