International Journal of Soft Computing 1(3): 170-174, 2006

A Genetic Algorithm Approach to Solve Broadcast Scheduling Problem

Sakthivel, S. and R. Suresh
Department of Computer Science and Engineering,
Alagappa University Karaikudi, India

Abstract: In this study, a Broadcast Scheduling Problem (BSP), a finite set of wireless radio-frequency stations are to be scheduled in a time division multiple access frame. The objective is to provide a collision free broadcast schedule which minimizes the total frame length and maximizes the slot utilization within in frame. The BSP is a combinatorial optimization problem and it is an NP-hard problem. Well-known in the literature as an NP-hard combinatorial optimization problem, this problem requires recourse to heuristic methods in order to obtain good (not necessarily optimal) solutions within a practical amount of time, In this context, heuristic approaches a Genetic Algorithm (GA) can be used. This study proposes a genetic algorithm to solve this problem. The implementation of this algorithm has been subject to extensive tests. The result obtained confirms the efficiency and the effectiveness of GA to provide good solutions to practical sized problems. The goals and the present status of our research are also discussed.

Key words: Broadcast scheduling problem, Ad-hoc networks, combinatorial optimization, nP-hard, heuristics, Genetic Algorithm (GA)

INTRODUCTION

During the last decade, there has been a tremendous growth in the deployment of wireless communication systems. This is due to improved technology and increasing demand. Of particulars interest are called *ad-hoc* network. The need to exchange information with a user, anywhere and anytime, led to the cellular mobile networks, which are wireless networks integrating several services including electronic mail. Having no fixed infrastructure, these networks are particularly useful in areas such as mobile commerce, combat search and rescue and other battlefield scenar networks, to name to a few[1]. However, there are inherent difficulties with *ad-hoc* networks, mainly concerning message scheduling and routing.

Since all stations in the network share the transmission channel, they must be scheduled to transmit messages that prevent destructive interference, or message collisions. There are two types of message collision. The first, referred to as direct collision occurs when two neighboring stations transmit during the same time slot. The second, hidden collision results when two non-neighboring stations transmit simultaneously to a station that can receive messages from both senders. The desired result is a schedule that is guaranteed to produce collision free transmissions[2].

It is shown that the Time Divisions Multiple Accesses (TDMA) protocol can be used to provide a collision free scheduling procedure. In a TDMA network, time is divided into frames. Each frame consisting of a number of fixed length slots. It is acceptable and in fact highly desirable for multiple stations to transmit during the same time slot provided that they do not cause any collisions, either direct or hidden collision. Exact transmission criteria will defined in next section.

Moreover, even if the problem is not a real-time problem, it cannot be solved exactly for many reasons. The first one is that the search for an exact solution is too much time-consuming. For a network with m time slot and n stations, m solutions for a worst case should be examined. With 100 Stations and 3 time slot, if we take 1 ns (1000 MHZ) per solution, the total running time is about $4x10^{13}$ years. The second reason is that, as the station grows, the operator will need to run again from time to time the assignment algorithm. Therefore, he could not afford prohibitive times. For these reasons, only heuristic approaches are used in the literature[3,4] to solve this problem.

PROBLEM FORMULATION AND RELATED WORKS

In the Broadcast Scheduling Problem a finite set of wireless radio-frequency stations are to be scheduled in a time division multiple accesses. An ad-hoc network can be conveniently described by an undirected graph G= (V,E) where the vertex set V={1,2,.....n} represent the

Corresponding Author: S. Sakthivel, Department of Computer Science and Engineering, Alagappa University Karaikudi, India

stations in the network and the edges set E represents the set of transmission link between the vertices. Two stations $i,j \in V$ are said to be one-hop neighbors if and only if they can directly communicate. That is, stations i and j are one-hop neighbors iff there exists an undirected edge $(i,j) \in E$. One-hop neighbors transmitting in the same slot with result in a direct collision. If $(i,j) \notin E$ but there exists an intermediate node $k \in V$ such that $(i,k) \in E$ and $(k,j) \in E$, then we say that stations i and j are two-hop neighbors. Two-hop neighboring stations which transmit in the same time slot will cause a hidden collision[2,5].

The topology of the network can then be described by an N x N symmetric binary matrix C, where $N = |V|$. Let $C = \{c_{ij}\}$ be the adjacency matrix and is defined as follows:

$$c_{ij} = \begin{cases} 1 & \text{if } (i,j) \in E \text{ and } i \neq j \\ 0 & \text{otherwise} \end{cases}$$

We assume that there are M time slots in each TDMA frame and that the time required to transmit one packet of information is equal to one packet length in time[6]. We also assume that packets are transmitted at the beginning of each slot and that the message is received during the same slot in which they are transmitted (sent). With this, we may define our broadcast schedule as an M x N binary matrix $S = \{S_{mn}\}$, where

$$S_{mn} = \begin{cases} 1 & \text{if station n is scheduled to transmit in slot m} \\ 0 & \text{otherwise.} \end{cases}$$

In order to perform some analysis on the efficiency of a schedule, it is helpful to know what percentages of the available slots are being assigned in a transmission frame. Let U_n be the slot utilization for station n. Then

$$U_n = \frac{1}{M}$$

It follows that U, the total slot utilization of the network is as follows:

$$U = \frac{\sum_{n=1}^{N} U_n}{N}$$

$$U = \frac{\sum_{m=1}^{M} \sum_{n=1}^{N} S_{mn}}{NM}$$

With this we can define the Broadcast Scheduling Problem as follows:

Minimize M
Subject to

$$\sum_{m=1}^{M} S_{mn} = 1, \quad \forall n, \quad ..... \quad (1)$$

$$C_{ij} + S_{mi} + S_{mj} \leq 2, \forall i, \forall j \text{ and } i \neq j, \text{ -----} \quad (2)$$

$$C_{ik} S_{mi} + C_{kj} S_{mj} \leq 1, \forall i, \forall j, \forall k \text{ and } \forall m$$
$$I \neq j, j \neq k, k \neq i, \text{ -------} \quad (3)$$

The Constraint (1) implies that at least one station transmits in each slot. The constraint (2) prevents direct collisions from occurring. Finally, constraint (3) prevents stations from broadcasting in a manner that cause hidden collisions[2].

Therefore, the objective is to provide a broadcast schedule free from both collision types which minimizes the total frame length and maximizes the slot utilization within that frame.

The recognition version of the BSP was proven to be NP-complete by the authors in[7]. This implies that an algorithm which optimally solves the problem in polynomial time is unlikely to exist. Thus the need for heuristics which provide high quality solutions within reasonable computation times arises.

It is possible to establish some lower bound for M which can provide about the minimum number of slots which will be required for a given broadcast schedule. In[8], propose a lower bounding lemma based on the degrees of the vertices in the graph. Specifically, for a given network, G= (V,E), define the degree of a given vertex $v \in V$, denoted deg(v), to be the number of edges incident to v. then, the frame length M satisfies the following inequality :

$$\max_{v \in V} . \deg(v) + 1 \leq M \text{ ------} \quad (4)$$

Though the bound in (4) is relatively easy to calculate and gives a hint for initializing of the search process for the BSP. However, it does not provide a tight lower bound for M.

ROPOSED GENETIC ALGORITHM

As computers become more and more powerful and omnipresent, they are expected to solve increasingly difficult problems. Since the Scheduling problem defined in the previous section as a special case, they turn out to be NP-hard. The development and use of optimization models is well established. Our research in this area is primarily concerned with using Genetic Algorithms to solve the scheduling problem. Genetic Algorithm (GA) is an evolutionary optimization approach which is an alternative to the traditional optimization methods[9]. GAs is a fairly easy and effective method of computing an nondeterministic problem. A GAs can be used to find a solution in much less time.

A Genetic Algorithm (GA) is an adaptive search technique based on the principles of natural evolution[10]. GAs iteratively applies genetic operators for selection, crossover, mutation and reproduction on a given population to improve its average fitness generation by generation. GAs has been successfully used to solve many combinatorial optimization problems such as Travelling Salesman Problem (TSP), the Assignment Problem and the Knapsack Problem, Scheduling problem. GAs[11] are composed of three phases: A phase of creation of an initial population, a phase of alteration of this population by applying various genetic operators on its elements and finally a phase of evaluation of this population during a certain number of generations. Each generation is supposed to provide new elements that are better than those of the preceding generation. It is hoped that the last generation will provide a good solutions. In our adaptation, we opted for a non-binary representation of the chromosomes[12]. As shown in Fig. 1, the genes (squares) represent the stations and the integers they contain represent the time slot to which the station i is assigned.

Initial population formation: The first element of the initial population is the one obtained when all stations are assigned to the time slot. This first chromosome is created in a probabilistic way. The creation of other chromosomes of the population is also in a probabilistic way. Various operators and functions are then applied to this population.

Station 2 is assigned to time slot 3 and Station 6 is assigned to time slot.

Crossover operator: This operator creates two new child chromosomes by crossing the parent chromosomes, taking genes 1,2...i of one parent and genes (i+1), ... n of the other parent for some randomly chosen i. We randomly choose a pair of chromosomes from the population and then create two new chromosomes by applying the crossover operators. It is used for reproduction.

Mutation operator: The mutation procedure randomly modifies the genes of chromosome by swapping two genes of randomly chosen. Sometimes a mutation can lead to a better solution that a crossover would not have found.

Stations →

1	2	3	4	5	6
2	3	2	2	3	1

Fig. 1: A Chromosome representation for a network of 3 time slots and 6 stations

do
*** generate a new population**
*** apply crossover operator**
*** apply mutation operator**
*** evaluate the obtained population**
*** verify both types of collision constraints if any one collision constraints fails, generate a new population**
until population is satisfies all constraints

Fig. 2: Genetic process flowchart

Evaluation procedure: The evaluation procedure (fitness function) measures the fitness of each individual solution in the population. It determines how well the chromosomes suit the needs of the problem domain. In this stage of evaluation, we check the both types of collision constraints. The chromosomes are satisfies both constraints, we keep that solutions, otherwise create a new chromosomes could make them feasible. This is a cycle. Each cycle runs several successive genetic processes. At every cycle, a new initial population is created. Figure 2 shows the flowchart of the genetic process.

RESULTS

In order to evaluate its performance, we have implemented the algorithm and applied it to solve problems that were randomly generated. The results of these experiments are reported below. In all the experiments, the implementation was conducted in C and all the experiments were run on a Personal Computer (PC) with a AMD 2400+ 266 MHz CPU and 256 MB RAM.

The performance of the algorithm was often evaluated based on the speed of improving the fitness of its solution and based on its complexity. To verify the performance of our algorithm, we performed some tests on networks of different sizes ranging from 4 stations to 6 stations. Each

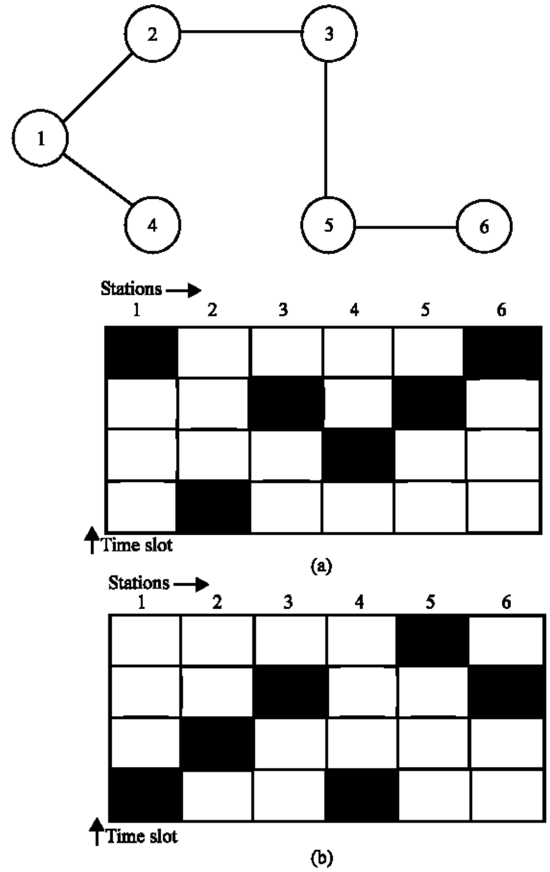

Fig. 3: Six station network problem (a) solution 1 (b) solution 2 (with 4 time slot)

test was performed 5 times and we report the average computing time.

Considering the following network with 6 stations in Fig. 3.

We compared the results obtained with our algorithm (GA) with those obtained by computer algorithms for H.Taha, Operation Research; An Introduction 7 th ed, 2003 of Integer Programming Method (TORA windows version 1.00 June 2002). We performed the tests on both methods. The methods have been solved by the Integer programming and we use the same sets of data to achieve the comparison. The results are reported in Table 1. These represent the computation times of different time slot of the same networks and all reported solutions are feasible.

The results of this comparative survey, our GA provides better results than the H.Taha's Computer algorithms of Integer programming methods for practical-sized networks. In summary, considering the overall performance of the H. Taha's Computer algorithms of Integer programming methods, the proposed GA generally gives better results than the H.Taha's Computer algorithms of Integer programming methods. Since, the H. Taha's Computer algorithm of Integer programming methods supports only 300 variable and 300 constraints.

Table 1: Comparative results for genetic algorithm with H. Taha's computer algorithm of integer programming method

# of slots m	# of stations n	GA (seconds)	H. Taha (seconds)
4	6	36.087912	247.28642
5	6	18.3516482	539.642
6	6	7.8461538	1339.86156

RESEARCH STATUS

The research we propose is the problem is modeled as a complex integer programming problem and it is an NP-hard problem and cannot be solved by exact methods for large sized problems. Well-known in the literature[13] as an NP-hard combinatorial optimization problem, this problem requires recourse to heuristic methods in order to good (not necessarily optimal) solutions within a practical amount of time. We hope that this study will serve as a starting point for further research and development.

CONCLUSION

In this study, we have investigated a multi-population GA to solve Broadcast scheduling problem. An experiment has been conducted to measure the quality of solutions provided by this algorithm. We showed that our results compete very well with existing method, producing broadcast schedules for several networks with optimal frame lengths. The result obtained confirms the efficiency and the effectiveness of GA to provide good solutions to practical sized problems. This heuristic can be used to solve NP-hard problems, like designing and planning, in the next-generation mobile networks. Further, new techniques are also under investigation. It is up to the designer to decide if such improvement methods are profitable for their individual problems.

REFERENCES

1. Yi-Bing Lin and Imrich Chlamtac, 2001. Wireless and Mobile Network Architectures: Wiley Computer publishing.
2. Yeo, J., H. Lee and S. Kim, 2002. An Efficient Broadcast Scheduling Algorithm for TDMA Ad-hoc Networks. Computers and Operations Res., 29: 1793-1806.
3. Sakthivel, S. and R. Suresh, 2004. A Genetic Algorithm approach to solve Mobile Base Station Location Problem. UGC sponsored National conference on Networking and Multi-agent systems, pp: 24.

4. Sakthivel, S. and R. Suresh, 2005. A Genetic Algorithm approach for assigning Mobile Base Stations to Switches in Cellular Mobile Networks. National level seminar on emerging trends in Communication and Networking, pp: 5-6.
5. Commander, C.W., S.I. Butenko and P.M. Pardalos, 2004. A Greedy Randomized Adaptive Search Procedure for Broadcast Scheduling Problem, Submitted.
6. Commander, C.W., S.I. Butenko and P.M. Pardalos, 2004. On the Performance of Heuristics for Broadcast Scheduling. In D. Grundel, R. Murphy and P.M. Pardalos (Eds.), Theory and Algorithms for Cooperative Systems, pp: 63-82, World Scientific.
7. Nobuo, F. and T. Yoshiyasu, 1993. A Parallel Algorithm for Broadcast Scheduling Problems in Packet Radio Networks. IEEE Transactions on Communications, 41: 6.
8. Saha, D., A. Mykherjee and P. Bhattacharjee, 2000. A simple heuristic for assignment of cell to switches in a PCS network. Wireless Pers. Commun., 12: 209-224.
9. Mitchell, M., 1999. An Introduction to Genetic Algorithms: Cambridge USA, London UK: MIT Press.
10. Davis, L., (Ed)., 1991. Handbook of Genetic algorithms: Van Nostrand Reinhold: New York, NY, U.S.A.
11. Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimisation and Machine Learning: Addison-Wesley, Don Mills.
12. Gondim, R.L.P., 1996. Genetic algorithms and the location area partitioning problem in cellular networks. Proceedings of the Vehicular Technology Conference '96, Atlanta, VA, pp: 1835-1838.
13. Salcedo-Sanz, S., C. Busono-Calzon and A.R. Figueiral-Vidal, 2003. A Mixed Neural-Genetic Algorithm for the Broadcast Scheduling Problem. IEEE Transactions on wireless Communications, 2: 2.