

Broadcast Scheduling Problem in TDMA Ad Hoc Networks using Immune Genetic Algorithm

D. Arivudainambi, D. Rekha

D. Arivudainambi, D. Rekha

Department of Mathematics

Anna University, Chennai, India

E-mail: arivu@annauniv.edu, rekhadurai@gmail.com

Abstract:

In this paper, a new efficient immune genetic algorithm (IGA) is proposed for broadcast scheduling problem in TDMA Ad hoc network. Broadcast scheduling is a primary issue in wireless ad hoc networks. The objective of a broadcast schedule is to deliver a message from a given source to all other nodes in a minimum amount of time. Broadcast scheduling avoids packet collisions by allowing the nodes transmission that does not make interference of a time division multiple access (TDMA) ad hoc network. It also improves the transmission utilization by assigning one transmission time slot to one or more non-conflicting nodes such a way that every node transmits at least once in each TDMA frame. An optimum transmission schedule could minimize the length of a TDMA frame while maximizing the total number of transmissions. The aim of this paper is to increase the number of transmissions in fixed Ad hoc network with time division multiple access (TDMA) method, with in a reduced time slot. The results of IGA are compared to the recently reported algorithms. The simulation result indicates that IGA performs better even for a larger network.

Keywords: Ad hoc networks, Broadcast Scheduling, Genetic algorithm, Immune genetic algorithm.

1 Introduction

A wireless ad hoc network is a collection of nodes, which communicate with each other using radio transmissions. In wireless networks, nodes can communicate directly if they are within the transmission range of each other. When a source node is out of the transmission range of destination node then it uses intermediate nodes for routing their message. In ad hoc networks, there are no base stations to act as routers and the nodes themselves perform routing. Therefore, data should be delivered from source to destination through multiple hops. Ad hoc networks rely on multihop transmission among the nodes on the same channel. We have proposed how efficiently broadcasting can be done in ad hoc network.

In this paper, we consider a fixed ad hoc networks with TDMA and its topology can be represented by a graph. TDMA is divided into frames where each frame is further divided into time slots that can be assigned to different nodes. In a single frame, all nodes must be allowed to transmit packets at least once. TDMA scheduling algorithms can be categorized into link scheduling algorithm and broadcast/node scheduling algorithm [10]. In Ad hoc network, a link can be represented as (t, r) , where t is a transmitter and r is a receiver. In link scheduling, the transmission in every slot is assigned to certain links, where as in broadcast scheduling the transmission in every slot is assigned to certain nodes. The aim of the algorithm is to activate all the nodes at least once, to improve the utilization factor of the network channels, i.e., to increase the number of transmissions within minimum number of time slots.

Ahmad et al., [1] proposed an algorithm based on finite state machine synthesis that determines the minimum frame length with the maximum slot utilization. A maximal compatible set of nodes is produced and these are chosen such that the nodes in that set do not have conflicts

with one another. A tight lower bound derived from set of maximal incompatibles forms the basis for deriving minimum frame length. The algorithm applies set of rules on the maximal compatibles in order to maximize utilization of slots.

Genetic algorithm (GA) is population-based stochastic optimization method with an iterative process of generation-and-test. It has been recognized that GA is promising approach for NP-hard or NP-complete problems. GA solves many search and optimization problems effectively. A standard genetic algorithm approach is given by Chakraborty in 1998 for scheduling problem in packet radio networks, though the algorithm able to solve small problems but performs poorly for large networks. This is because classical crossover and mutation operations create invalid population that goes through several generations and delay the progress of search for valid solutions. Special crossover and mutation operations for elite population method is defined by Chakraborty [3], such that members of the population always remain valid solutions for the problem. Even though, it produces optimal solution in less number of generations and it reduces invalid solutions but the computation time is not reduced.

Gunasekaran et al., [4] proposed two different algorithms for spatial reuse in WiMAX networks. First, a dynamic programming (DP) method is adopted to produce maximal collision free set of nodes, but suffered from high memory requirements. The second one is genetic algorithm approach, it is more scalable than DP approach but does not guarantee optimality. Two kinds of population are used in co-evolutionary genetic algorithm that act as competitors for each other and hence increase the evolving rates. This approach seems to have a better performance than classical genetic algorithm approach, which did not produce good result for large networks. The main drawback of co-evolutionary genetic algorithm is every member of test population has to be compared with every member of solution population. This requires many comparisons, calculations and hence may slow down the procedure when population sizes are vast.

A novel hysteretic noisy chaotic neural network (HNCNN) by controlling noises of the equivalent model for broadcast scheduling problem in packet radio networks is proposed by Ming Sun et al., [7]. They combine the HNCNN with the gradual expansion scheme to find the minimal frame length in the first phase, and to maximize the conflict-free transmission in the second phase.

Ngo and Li [8] proposed an approach based on a modified GA called genetic-fix. They formulate the problem based on a within-two-hop connectivity matrix and proposed a centralized scheduling algorithm using a modified genetic-fix algorithm. Traditional GA generates subsets of all possible sizes whereas genetic-fix algorithm generates fixed-size subsets i.e., in binary representation number of one's is fixed.

A mixed tabu-greedy algorithm has been implemented to solve the broadcast scheduling problem in packet radio networks is given by Peng et al., [9]. Improvements are achieved in terms of both channel utilization and packet delay by using a two-step algorithm.

The problem of determining an optimal minimum length TDMA schedule for a general multihop radio network is NP-complete for both link and broadcast scheduling [10]. Both link scheduling and broadcast scheduling are considered in their approach.

Salcedo et al., [11] propose a procedure that combines Hopfield neural network for the constraints satisfaction and genetic algorithm for achieving maximal throughput. Their approach to solve broadcast scheduling problem is by dividing problem in two sub problems. The first is to find minimum frame length without interference using discrete Hopfield neural network. The second increases the throughput for given frame length is done by combining Hopfield neural network with genetic algorithm.

A linear integer programming formulation for the composite problem of maximizing channel utilization while minimizing the length of the frame is given by Syam menon [12] that performs in reduced computation time but maximum number of stations taken in their approach is 50

stations.

In [13], Wang and Ansari propose a broadcast scheduling algorithm based on mean field annealing neural networks. They proposed a three-step algorithm first step reduces solution space by presetting some neurons according to the topology of the scheduling network. Second step executes MFA procedure to maximize channel utilization. A heuristic approach is the final step to arrange transmissions of unassigned stations.

A competent permutation encoded genetic algorithm is executed to solve optimum time division multiple access broadcast scheduling problem for mobile ad hoc networks is proposed by Wu et al., [14]. The problem search space is reduced mostly and genetic algorithm becomes more capable in searching the optimum solutions.

In genetic algorithm, the mutation creates new genes for the population and crossover operator orients seeking the best solution from genes in the population. However, they may drop into local optimal solutions or they may find optimal solution by low convergence speed and GA blindly wanders over search space. In GA, a normal mutation operator takes chance to change a best solution obtained from previous operation. To overcome these problems, we used immune concept to enhance the GA. Immune genetic algorithm (IGA) does not perform mutation in a normal way and mutation operator is carried out by two steps in this algorithm 1) Immune selection and 2) Vaccination. Immune selection performs reduction of time slots whereas vaccination gets the knowledge from hop matrix to mutate a bit. IGA increases number of transmission in a reduced time slot. The aim of proposed immune genetic algorithm is to activate all nodes at least once, while minimizing number of time slots. Another goal is to increase utilization factor of network channels, i.e., the usage of available channels should be maximized by increasing number of transmissions. The method based on immune genetic algorithm for the broadcast scheduling problem in Ad hoc networks has not been studied so far. A comparison with existing methods for the test instance reported in the literature shows that our algorithm identifies optimal solution in less number of generations and the average time delay is reduced even for large network.

This paper is organized as follows: Section 2 gives a formal definition of broadcast scheduling problem, along with constraints to be satisfied. An immune genetic algorithm approach is provided in Section 3. Simulation, results and analysis of the proposed algorithm are provided in Section 4. Section 5 concludes the paper.

2 Formulation of Broadcast Scheduling Problem

Ad hoc topology can be represented as undirected graph $G = (N, E)$, where N is the set of nodes or mobile stations and E is the links or transmissions assumed bidirectional. A link (i, j) exists between nodes i and j if they are within transmission range of each other. When node i transmits data, other nodes within the transmission range of i will receive it. If both i and j transmit packet in the same time slot then it leads to primary conflict. When a node receives two or more packets from directly connected different nodes in a single time slot then it cause to secondary conflict. For a collision free transmission primary conflict and secondary conflict should not occur. Two mobile stations can transmit in the same time slot without mutual interference, if they are located more than two hops apart. Three important matrixes used in this study are Connectivity matrix $[CM]$ is to represent a direct link between nodes, Hop information matrix $[HM]$ is to represent one-hop and two-hop connectivity information of each node and TDMA frame matrix $[TM]$ is to represent the allotted time slots of given network without any interference.

The connectivity matrix in Fig.2.(a) represents direct link between nodes given in the network Fig.1. Each column represents the nodes of network and row represents link existence between nodes. In Fig.2.(a), first row says about connectivity information of node 1, likewise for remaining

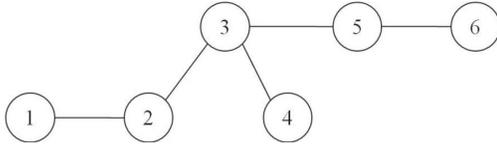


Figure 1: A sample 6-node network

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	1	0	1	0	0	0
3	0	1	0	1	1	0
4	0	0	1	0	0	0
5	0	0	1	0	0	1
6	0	0	0	0	1	0

(a)

	1	2	3	4	5	6
1	0	1	1	0	0	0
2	1	0	1	1	1	0
3	1	1	0	1	1	1
4	0	1	1	0	1	0
5	0	1	1	1	0	1
6	0	0	1	0	1	0

(b)

Figure 2: (a) Connectivity matrix and (b) Hop information matrix for a 6-node network

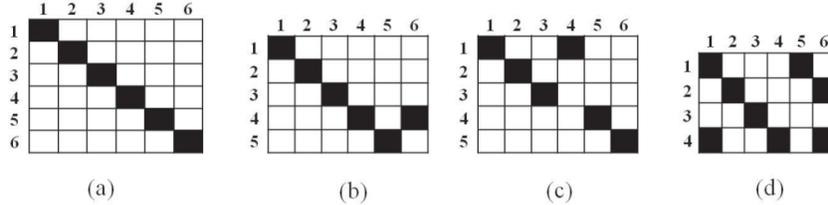


Figure 3: (a), (b), (c) and (d) Sample solution TDMA frames created for a 6-node network

nodes. The matrix has value 0 or 1, where 1 represents existence of a link. The hop information matrix for the given 6-node network is shown in Fig.2.(b), where row value represents one-hop and two-hop information between nodes. The matrix takes value 0 or 1 from first row, it is identified nodes 2 and 3 is either one-hop or two-hop away from node 1.

The TDMA frame matrix is a $|M| \times |N|$ matrix where $|M|$ is the number of time slots and $|N| = \{n_1, n_2, \dots, n_x\}$ is total number of nodes in the network. For the 6-node network, possible TDMA frame matrix is shown in Fig.3. The first $[TM]$ is a trivial solution where there is no chance to get conflict since each node is assigned in each time slot. Fig.3.(b) & (c). represents solution frame with reduction in time slot but it is not an optimal solution, whereas the solution frame in Fig.3.(d) is generated with 4 time slots which is optimal solution for the given 6-node network. The optimal solution is determined based on fitness criteria Tight lower bound and Channel utilization variable.

Tight lower bound is

$$MD = \max_{n \in |N|} |DS(n)| \quad (1)$$

Let $DS(n)$ be the degree set of n nodes, MD represents maximum degree of the network, based on this value tight lower bound is generated as,

$$\Delta = |M| - MD \geq 1 \quad (2)$$

If $\Delta = 1$ then the solution is optimal.

Channel utilization variable for entire network

$$\rho = \frac{1}{|M| * |N|} \left(\sum_{i=1}^{|M|} \sum_{j=1}^{|N|} [TM_{ij}] \right) \quad (3)$$

The total number of transmission taken by each node is calculated using

$$\rho_x = \sum_{i=1}^{|M|} [TM_{ix}] \quad (4)$$

Applying the criteria in the sample TDMA solutions shown in Fig.3 it is identified solution in Fig.3.(d) satisfies tight lower bound value with high channel utilization value compared to solution in Fig.3.(a).(b) & (c). MD value for the given 6-node network is 3, so $\Delta = 1$ for solution in (d) and channel utilization of entire network is 0.33, where as in solution (a) channel utilization ρ is 0.17, for (b) and (c) ρ is 0.2.

3 Proposed IGA

3.1. Analysis of Genetic Algorithm

Genetic algorithm is a heuristic search technique that simulates the processes of natural selection and evolution. Genetic algorithms are effective, robust search procedure for NP-complete problems. GA is a nontraditional search and optimization method. It works for the problem that has large number of solutions, out of which some are feasible and some are infeasible. The task is to get best solution out of feasible solutions. Standard GA starts with a set of solutions called population. A population is a collection of chromosomes. Solutions for one population are taken and used to form a new population, which are selected according to their fitness. This is repeated until some conditions for improvement of best solution are satisfied. Three main operators are used to create a new population 1) selection, 2) crossover, and 3) mutation. The new population is further evaluated and tested for termination. If termination criteria are not satisfied, the population is iteratively operated by three operators and evaluated. One sequence of these operations and subsequent evaluation procedure is a generation in GA.

3.1.1. Initial Population

TDMA scheduler matrix is represented as bit string chromosome containing 0's and 1's in broadcast scheduling problem. Each row and column of scheduler matrix represents to time slot and node transmission. The value 1 in position (i, j) in scheduler matrix indicates j^{th} node is allowed for transmission in i^{th} time slot. In classical GA approach, chromosome contains random string of 0's and 1's that does not perform well when number of nodes exceeds 40 [3]. The reason is when number of generations increases number of invalid individuals in population keeps increasing as the validity of every individual is not ensured while following classical GA crossover and mutation operations. Finally, invalid members dominate total population and thus optimal solution is not found. This was overcome by changing initial population, crossover and mutation methods to keep members of population valid by Chakraborty [3]. In our algorithm, initial TDMA frames are constructed using Elite population method of Chakraborty [3].

3.1.2. Selection

The selection operators for parent selection and survivor selection follow Darwinian principle of survival of the fittest. For parent selection two chromosomes is selected randomly from the population to serve as parents for reproduction process. Second, survivor selection applies the principle of survivor of the fittest. Only fittest individuals selected as parents for next-generation, to achieve this k-tournament selection is done. Survivor selection is also called as elitism, which is to retain some of the best individuals in each generation. In this study, a small percentage of best fitness individuals retained to next generation. It increases the performance of algorithm, by preventing loss of best found solution. From each generation 10% of best solution is retained to next iteration.

3.1.3. Crossover and Mutation

The selected chromosomes for reproduction are gathered in mating pool. The single-point crossover operator is done on rows of the population. Once a crossover point is identified, a random row from first parent PR1 is crossed over with a random row from second parent PR2. The resultant chromosome CH1 is replaced with PR1 and CH2 is replaced with PR2. After replacing, if the solution violates constraints then it is removed from population. The

mutation operator behaves in a different manner depending on the fitness of selected gene. The mutation operator changes one bit in selected chromosome depending on individual fitness. Simple mutation is done by flipping a bit. At every bit of all members of population, a random number between 0 and 1 is generated. If it is less than or equal to mutation probability, it is flipped from 0 to 1 and vice versa.

3.1.4. Fitness function

The fitness function evaluates quality (fitness) of candidate solutions. The fitness function for scheduling problem is based on the variables tight lower bound and channel utilization. The termination point determines whether optimal solution is determined in that generation or not. The optimal solution is the one, which satisfies both criteria. When the generation of evolution reaches this termination point, algorithm stops and output optimal solution for the given network, else elitism method is done on populations and proceeds to next generation. At the end of iteration before evaluating fitness function, populations produced in the generation are taken for duplicate row elimination i.e., time slot which is repeated is removed from the population in order to produce optimized TDMA frame.

3.2. Optimization properties of Immune Genetic Algorithm compared to GA

In GA two main genetic operators, crossover and mutation, not only give each individual's the evolutionary chance to obtain global optimum but also cause the degeneracy to some extent because of random and unsupervised searching during the entire process. On the other hand, GA is lack of capability of making use of some basic and obvious characteristic or knowledge in pending problem. Based on the considerations above, immune Genetic Algorithm is proposed. Algorithm 1 shows the structure of immune genetic algorithm. The solution after crossover is taken for immune operations. IGA is an intelligent optimization algorithm, which mainly constructs an immune operator accomplished by two steps: Immune selection and Vaccination. The initial populations are created using elite population method of Chakraborty [3]. Random selection method for parent selection and 10% of best solution is taken to next generation. The knowledge added IGA algorithm performed in the following way.

Algorithm 1. Immune Genetic Algorithm

```

Immune Genetic Algorithm( $G=(V,E)$ ,  $Psel$ ,  $Pcr$ ,  $Psize$ ,  $maxgen$ )
{
    Generate compatibility and hop information matrix;
    Find the degree of the network;
    for loop=1:Psize
        population = generate initial population using Elite population;
    end for
    while (until reaches maxgen or optimal solution is identified )
        // two parents are selected based on the selection probability
        Spop = selection ( $Psel$ , population);
        // single point crossover is done on the selected parents
        Cpop = crossover ( $Pcr$ , Spop);
        // the chromosomes after reproduction is taken for immunization
        Immunization (Cpop)
        {
            // solutions that satisfies primary and secondary constraints are taken by immune selection
            ISel = ImmuneSelection (Cpop);
            // resulting solutions are arranged according to channel utilization variable
            // vaccination is performed based on hop information matrix
            Vpop = Vaccination (ISel);
        }
}

```

```

    Evaluate the fitness of each Vpop
    If optimal solution is identified then
    break;
    // come out of the while loop and print the solution
    else
    // replace the best few offspring with the initial population and continue the loop
    Survival (initial population, Vpop);
    end if
    end while
}

```

3.2.1. Crossover

The modified crossover operator given by Chakraborty [3] is performed in IGA. Crossover is done on rows of the TDMA cycle. Rows from members of the population, i.e., different TDMA frames are selected using predetermined crossover probability and are marked to be members of mating pool. A pair of rows PR1 and PR2 selected randomly from the mating pool for crossover. The objective of crossover is to create an offspring with a better schedule for a time slot with more transmissions by combining parent schedules. By crossover operation, only one child is created which may or may not replace the parent, depending on how good it is. First, PR1 is logically AND with PR2, then logically exclusive OR operator is performed for PR1 and PR2. Both new populations are scanned from left to right, the first occurred one is copied to same position of new row. The next one is checked with corresponding row of hop information matrix if it does not make conflict then it is copied to new row else, it is rejected. In the same way, it will continue until end of both population and a new row is created from PR1 and PR2. The offspring may replace with PR1 and PR2 based on the condition set given by Chakraborty [3].

3.2.2. Mutation

Random mutations alter a certain percentage of bits in the list of chromosomes. A single point mutation changes 1 to 0, and vice versa. Increasing number of mutations increases algorithm's freedom to search outside the current region of variable space. It also tends to distract the algorithm from converging on a correct solution so mutations are not allowed for best solutions. They are designated as elite solutions destined to propagate unchanged. Such elitism is very common in GAs. Why throw away a perfectly good solution? However, in previous algorithm solutions generated after crossover is taken for mutation operation and random mutation is done on populations, which may change best population. To avoid this and to make algorithm knowledgeable, immune concept is added to genetic algorithm. A normal mutation operation is not done in IGA instead, knowledge added two steps are performed in place of mutation. The two steps of IGA are Immune selection and vaccination.

A. Immune selection

The newly created population after crossover, which satisfies the primary and secondary constraints, is selected for reduction of time slots. Those populations selected are stored in vaccine pool. From the vaccine pool, one population is taken and the algorithm finds repeated time slot in that population. If so then the repeated time slot is deleted from the population. This step is carried out for the remaining populations that are in vaccine pool. The resulting time slot reduced populations are arranged according to the channel utilization variable and stored in the vaccine pool by replacing the old populations.

B. Vaccination

Vaccination is used to improve fitness by modifying the genes of an individual population with prior knowledge to gain higher fitness with greater probability. A chromosome from vaccine pool is taken for vaccination. IGA identifies a node transmits first in the population in a time slot. During same time slot some other node, which does not create interference with transmitting

node can be allowed to transmit in the same time slot. To perform this a node is selected randomly and checked with Hop information matrix whether it creates an interference with currently transmitting node, if not node value is mutated to one allowing the selected node to transmit in same time slot. The genes of selected chromosome are modified based on the knowledge obtained from Hop information matrix of given network. Hence, the vaccination process increases number of transmissions.

4 Solving BSP Using IGA: Simulation and Results

A series of simulations are carried to evaluate the performance of IGA to solve broadcast scheduling problem, in comparison with finite state approach [1], GA [3], GA with collision free set (GACFS) [4], Mean field annealing [13] and competent permutation genetic algorithm [14]. The performance of the GA and IGA were tested for a large number of times, using Ad hoc networks of different sizes. In the following sections, we discuss simulation results regarding the number of nodes $|N|$, the number of timeslots $|M|$ and the computation time. The IGA was tested by 100 randomly generated graphs, each representing an Ad hoc network topology.

The simulation results are based on population size 50, maximum number of generations 500, crossover rate 0.30, mutation probability 0.001 and on the measures,

1 Tight lower bound Δ value is one.

2 Channel utilization variable to find the improvement in number of transmission.

4.1 Results obtained from GA

The purpose this simulation was to investigate performance of genetic algorithm for different networks. The number of nodes taken for simulation ranges from five to hundred. Smaller node networks executed with more number of transmission in an acceptable generation. However, for a 100 node network with 200 edges identifies the optimal solution TDMA frame after 489 generations. The average number of generations for 100-node network is 410. This has to be decreased in order to reduce execution time.

4.2 Results obtained from IGA

The simulation results based on IGA is given in the following Table 1. Compared to genetic algorithm, knowledge added IGA could improve the searching ability, adaptability and greatly increase the converging speed. During vaccination process, selected antigen is improved with more number of transmissions so channel utilization is increased. Comparing simulation results of IGA with GA number of generations is reduced and utilization of each network is improved. Even for large network, the solution is identified with acceptable generation.

The values of Table 2 clearly imply IGA performs better than standard GA. One main aim is to reduce number of time slots so there is a steady decrease in utilization index as the number of nodes increases.

4.3 Optimum schedule performance with other methods

The channel utilization of IGA is high compared to the channel utilization of elite population genetic algorithm given by Chakraborty [3] is shown in Fig.4. Similarly, for networks with nodes 14, 16 and 40 number of transmissions generated by IGA is somewhat closer to GA [3] whereas for large networks IGA has improved on an average of 20 to 30 transmissions is shown in Fig.4. Table 3 gives a detailed comparison between IGA and existing algorithms. It represents maximum number of transmissions produced by some recent algorithms for the given number of nodes and time slots. The results indicate for smaller networks the number of transmission differs slightly, whereas for a 100-node network number of transmission generated by IGA create a variation of 10 to 20 transmissions.

Two benchmark problems discussed by [13] are solved using IGA and results are compared with other algorithms such as, the finite state machine based algorithm FSMA [1], co-evolutionary

genetic algorithm for collision free set GACFS [4], gradual hysteretic noisy chaotic neural network G-HNCNN [7], Mean field annealing algorithm MFA [13] and the competent permutation genetic algorithm CPGA [14] is shown in Table 4. 30 nodes with 70 edges is analyzed in problem instance # 1 the channel utilization is largely improved compared to other algorithms similarly the time delay is also reduced reasonably by IGA compared to other methods. 40 nodes with 66 edges and maximum degree of 7 is analyzed in problem instance # 2 to this the channel utilization is increased moderately and the time delay is reduced by IGA. The results of other algorithm G-HNCNN, GACFS, FSMA, CPGA and MFA are given in [7], [4], [1], [14] and [13]. The algorithm GACFS and CPGA have not calculated the average time delay, which is represented by hyphen in Table 4.

The average time delay is calculated by

$$\eta = \frac{|M|}{|N|} \sum_{i=1}^{|N|} \left(\frac{1}{\sum_{j=1}^{|M|} |TM_{ij}|} \right) \quad (5)$$

The average time delay η for each node represents the average availability of the network, and the minimal η is very important for the optimal Ad hoc network design and its evaluation is described by Ming sun et al., [7].

In Fig.5 (a) the input graph topology for a 100-node network with 200 edges with average degree of 4 and its optimum solution produced by IGA is shown in Fig.5 (b). The solution is identified with 9 time slots and 152 transmissions compared to recent algorithms the number of transmissions produced by IGA is largely improved that attain the aim of this paper.

Problem Number	No. of Nodes	Average Degree	No. of time slots $ M $	Total Transmissions	Channel Utilization ρ	Computation time
1	14	3	6	24	0.285	0.5 sec.
2	16	3.6	5	23	0.287	0.5 sec.
3	40	4	8	67	0.209	2.1 sec.
4	100	4	10	152	0.152	5.2 min.
5	200	4	9	282	0.157	15.7 min.
6	300	3	10	489	0.163	36.1 min.
7	400	4	10	623	0.156	69 min.

Table 1: Simulation results of IGA with total number of transmission, channel utilization ρ , and the computation time.

Problem Number	No. of Nodes	Average Degree	No. of time slots $ M $	Channel Utilization ρ
1	10	3	4	0.3
2	20	3.6	7	0.286
3	30	4	8	0.208
4	50	4	9	0.2
5	100	4	10	0.152
6	120	4	9	0.149

Table 2: Results obtained by IGA for varying number of nodes, number of time slots and channel utilization ρ .

No. of Nodes	No. of time slots $ M $	IGA	GACFS	GA	Finite state approach	Component permutation GA
15	8	20	18	17	20	20
30	9	37	28	33	35	37
40	8	69	65	65	64	64
100	9	152	139	133	134	136

Table 3: Comparison of time slots $|M|$ and total number of transmissions using IGA with other competitive algorithms.

Instance	Parameter	IGA	G-HNCNN	GACFS	FSMA	CPGA	MFA
#1	$ M $	10	10	10	10	10	12
	ρ	0.19	0.1233	0.093	0.1167	0.1233	0.1056
	Avg. Time delay	7.54	8.83	—	9.2	—	10.5
#2	$ M $	8	8	8	8	8	9
	ρ	0.237	0.2125	0.203	0.200	0.200	0.197
	Avg. Time delay	5.212	5.7056	—	6	—	6.9

Table 4: Comparison of time slot $|M|$, channel utilization ρ and average time delay of each node by IGA with other competitive algorithms.

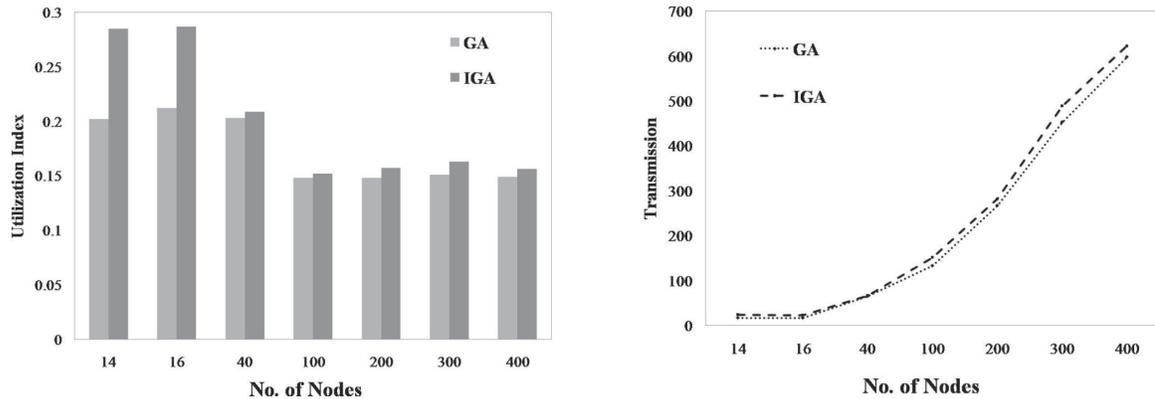


Figure 4: (a) Channel utilization vs. number of nodes, (b) Number of transmissions vs. number of nodes for GA [3] and IGA

5 Conclusion

The basic genetic algorithm and knowledge added immune genetic algorithm are discussed to improve broadcast scheduling problem in Ad hoc. According to our knowledge, this paper is the first one to study immune genetic algorithm to the broadcast scheduling problem. Compared to GA, IGA actively aims on improving solutions, while GA blindly wanders over search space. Immune genetic algorithm gets knowledge from hop matrix during vaccination process and increases number of transmissions in a reduced time slot in an acceptable computation time compared to recently proposed algorithms. The simulation results confirm advantages of IGA in terms of channel utilization, number of generations and running time. The outcome validates the effectiveness and efficiency of IGA for broadcast scheduling problem. Further research may

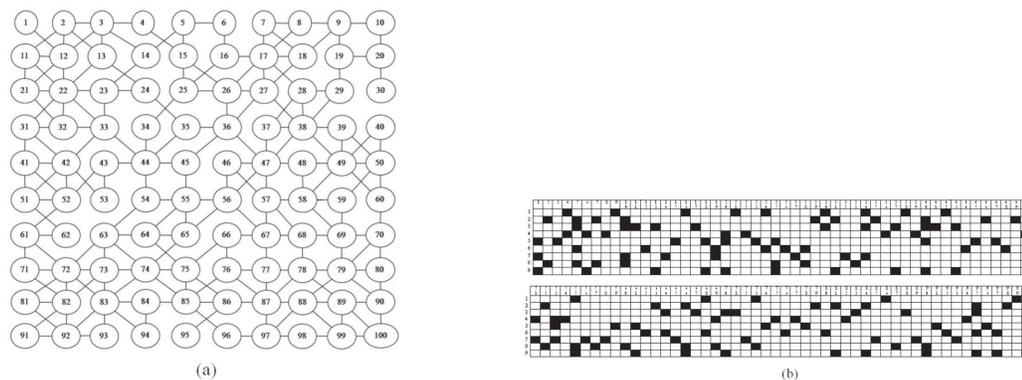


Figure 5: (a) Graph topology with 100 nodes, 200 edges and average link degree of 4, (b) Solution found by IGA for the same network

be performed to improve the performance of IGA by finding variations in crossover and altering the immune operator by adding refined knowledge to vaccination.

Acknowledgement

We gratefully acknowledge Department of Science and Technology, INDIA for providing financial support to carry out this research work under PURSE scheme.

Bibliography

- [1] I. Ahmad, B. Al-Kazemi, and A.S. Das. (2008); An efficient algorithm to find broadcast schedule in ad hoc TDMA networks, *Journal of Computer Systems, Networks, and Communications*, 12 : 1-10.
- [2] Dingwei Wang, Richard Y.K. Fung, and W.H. Ip. (2009); An immune-genetic algorithm for introduction planning of new products, *Computers and Industrial Engineering*, 56 : 902-917.
- [3] Goutam Chakraborty. (2004); Genetic algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks, *IEEE Transactions on Communications*, 52 (5) : 765-777.
- [4] R. Gunasekaran, S. Siddharth, P. Krishnaraj, M. Kalaiarasan, and V. Rhymend Uthariaraj.(2010); Efficient algorithms to solve broadcast scheduling problem in WiMAX mesh networks, *Computer Communications*, 33 : 1325-1333.
- [5] Licheng Jiao and Lei Wang.(2000); A novel genetic algorithm based on immunity, *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems And Humans*, 30(5) : 552-561.
- [6] M. Liu, W. Pang, K.P. Wang, Y.Z. Song, and C.G. Zhou.(2006); Improved immune genetic algorithm for solving flow shop scheduling problem. *Computational Methods*, 1057-1062.
- [7] Ming Sun, Lin Zhao, Wei Cao, Yaoqun Xu, Xuefeng Dai, and Xiaoxu Wang.(2010); Novel hysteretic noisy chaotic neural network for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Neural Networks*, 21(9).

- [8] C. Y. Ngo and V. O. K. Li.(2003); Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms, *IEEE Transactions on Communications*, 51(9) : 1439-1441.
- [9] Y. Peng, B.H. Soong, and L. Wang.(2004); Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm, *Electronics Letters*, 40 (6) : 375-376.
- [10] S. Ramanathan and E. L. Lloyd.(1993); Scheduling algorithms for multihop radio networks, *IEEE/ACM Transactions on Networking*, 1(2) : 166-177.
- [11] S. Salcedo-Sanz, C. Bousoño-Calzon, and A.R. Figueiras-Vidal.(2003); A mixed neural-genetic algorithm for the broadcast scheduling problem, *IEEE Transactions on Wireless Communications*, 2 : 277-283.
- [12] Syam Menon.(2009); A sequential approach for optimal broadcast scheduling in packet radio networks, *IEEE Transactions on Communications*, 57(3) : 764-770.
- [13] G. Wang and N. Ansari.(1997); Optimal broadcast scheduling in packet radio networks using mean field annealing, *IEEE Journal on selected areas in Communications*, 15 : 250-260.
- [14] X. Wu, B.S. Sharif, O.R. Hinton, and C.C. Tsimenidis.(2005); Solving optimum TDMA broadcast scheduling in mobile ad hoc networks: a competent permutation genetic algorithm approach, *IEE Proceedings: Communications*, 152(6) : 780-788.