Wireless Netw (2012) 18:787–798
DOI 10.1007/s11276-012-0433-4

An evolutionary algorithm for broadcast scheduling in wireless multihop networks

D. Arivudainambi · D. Rekha

Published online: 17 May 2012

Abstract A technical challenge in successful deployment and utilization of wireless multihop networks (WMN) are to make effective use of the limited channel bandwidth. One method to solve this challenge is broadcast scheduling of channel usage by the way of time division multiple access (TDMA). Three evolutionary algorithms, namely genetic algorithm (GA), immune genetic algorithm (IGA) and memetic algorithm (MA) are used in this study to solve broadcast scheduling for TDMA in WMN. The aim is to minimize the TDMA cycle length and maximize the node transmissions with reduced computation time. In comparison to GA and IGA, MA actively aim on improving the solutions and is explicitly concerned in exploiting all available knowledge about the problem. The simulation results on numerous problem instances confirm that MA significantly outperforms several heuristic and evolutionary algorithms by solving well-known benchmark problem in terms of solution quality, which also demonstrates the effectiveness of MA in efficient use of channel bandwidth.

Keywords Wireless multihop networks · Broadcast scheduling · Genetic algorithm · Immune genetic algorithm · Memetic algorithm

1 Introduction

In wireless ad hoc network, single hop and multihop networks does not rely on a preexisting infrastructure, such as routers in wired networks or access points in managed wireless networks. Instead, each node participates in routing by forwarding the data for other nodes, so the determination of which node forwards the data is made dynamically based on the network connectivity. In a single hop network, each mobile station (MS) can communicate directly with all other MSs. In wireless multihop networks, one or more intermediate node along with the path receives and forwards the packets via wireless links. Wireless multihop network extends the coverage of a network, improves the connectivity and transmission over multiple short links, which require less transmission power and energy than that required over long links. It also provides robust communication, rapid deployment and responds quickly in dynamic environments.

Figure 1 represents a simple wireless multihop network, each node represents a mobile station and a line connecting two nodes indicates that the two MSs are within the communication range. The neighbors of A are those MSs that can communicate directly with A (i.e., B and C). Node mobility in WMN causes frequent changes in the network topology. The main difficulty in designing WMN is that not all MSs can communicate directly with each other.

TDMA consists of fixed length time slots where each node transmits in at least one slot. A wireless multihop network consists of many MSs, where each MS has a certain number of neighboring MSs. Time is assumed to be divided into slots, each of duration equal to one maximum-length packet transmission time plus the maximum propagation time between any two MSs. MSs are assumed to use omnidirectional antennas. The wireless channels are assumed to be noise free and an unsuccessful reception is due to collisions only. MSs operate in half-duplex mode, i.e., a MS can transmit or receive, but cannot do both at the same time. Two or more non-conflicting MS can share the same slot.

Conflicts in multihop networks may occur in two ways: primary conflicts and secondary conflicts. The primary

D. Arivudainambi (✉) · D. Rekha
Department of Mathematics, Anna University, Chennai, India
e-mail: arivu@annauniv.edu

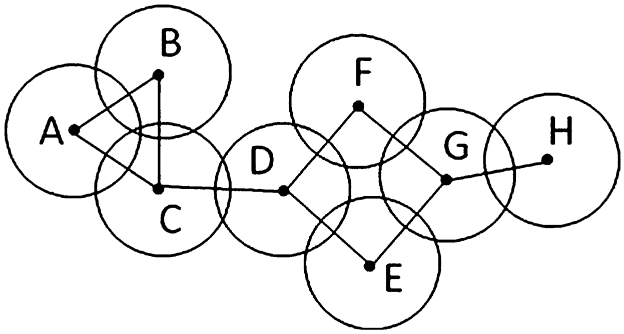

Fig. 1 A simple wireless multihop network

conflict occurs when two connected nodes transmit simultaneously. A secondary conflict occurs when two or more packets arrive at a node in a single time slot. This will occur when two nodes at a distance of two hops allowed transmitting simultaneously. Then, the intermediate node will receive two different packets from two directly connected nodes at the same time slot. Two MSs can transmit in the same time slot without mutual interference, if they are located more than two hops apart. In addition to the scheduler properties mentioned above, it is desirable for a WMN scheduler to possess the feature *Low connectivity information requirement* i.e., some algorithms need global network connectivity information while others require only local (e.g., one or two-hop) connectivity information. Since communicating this information consumes bandwidth, it should be to a minimum.

The fundamental computational and algorithmic issues in the broadcast scheduling problem of wireless multihop networks are discussed in chapter 16 of [1]. Most broadcast scheduling algorithms operate by producing a finite length nominal schedule in which each station has assigned at least one slot for transmission and then indefinitely repeating that nominal schedule. The problem was proven NP-complete [2, 3]. Most WMN schedulers are either node activation or link activation [4], one example of the node activation algorithm is given in [2]. A node activation scheduler selects the nodes for transmission in such a way to ensure that all its neighbors will receive a packet from any node correctly. A link activation scheduler chooses the nodes for transmission to guarantee that the destination node receives the packet successfully.

Various algorithms are proposed to solve the scheduling problem [2, 3, 5–17]. These algorithms are classified as graph theoretic [2, 18], graph coloring [19] and probabilistic approaches such as mean field annealing [3], tabu search [10], genetic algorithms [7, 9, 12], neural networks algorithms [11, 20] and mixed neural-genetic procedure [8]. Most of these algorithms are based on either of two points: one minimizes the frame length without considering the slot usage and the other attempts to maximize the slot utilization within the frame. Optimizing the two objectives separately does not lead to a good solution. A better approach is considering both of these criteria in an integrated fashion to solve the broadcast scheduling problem. The algorithm in [8] combines a Hopfield neural network for the constraints satisfaction and a genetic algorithm for achieving a maximal throughput. An approach based on a modified GA, called genetic-fix is given in [9] that generates and manipulates individuals with fixed size to reduce the search space substantially. A mixed tabu–greedy algorithm is implemented in [10].

A TDMA frame with less number of time slots, maximum number of transmission with elite population method and modified crossover operator in genetic algorithm is proposed in [12]. Even though the optimal solution is identified in less number of generations, the execution time is not reduced. Based on the concepts from the field of finite state machine synthesis is given in [14]. The stations that can broadcast without collisions among themselves grouped as maximal compatibles. 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 the utilization of slots. In [16] a simple and fast randomized algorithm to find a pool of valid solutions of the scheduling problem is proposed. Even though considering both the criteria by these algorithms the computation time is not reduced. In [21] linear integer-programming formulation is proposed to this problem, which performs in reduced execution time but the maximum number of stations taken in their approach is 50 stations.

A brute force approach with dynamic programming is used in [17] to improve the efficiency by eliminating the repeated states and co-evolutionary genetic algorithm approach is used to solve the collision free set for WiMAX mesh network. The main drawback of this co-evolutionary algorithm is that every member of test–case–population had to be compared with every member of solution-population. This requires many comparisons and calculations hence might slow down the process when the population sizes are huge.

Gradual noisy chaotic neural network (G-NCNN) to solve the NP-complete broadcast scheduling problem in packet radio networks is given in [22]. A two-phase optimization approach is adopted to achieve the two objectives with two different energy functions. In the first phase, a G-NCNN which combines the noisy chaotic neural network (NCNN) and the gradual expansion scheme to find a minimal TDMA frame length. In the second phase, the NCNN is used to find maximal node transmissions in the TDMA frame obtained in the first phase.

Hysteretic noisy chaotic neural network (HNCNN) is proposed in [15] by controlling noises of the equivalent model. 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.

In [23], a shortest path based load balanced internet protocol routing scheme with hose model (SLBIP) is proposed. Networks with varying number of nodes, links and average degree is taken for performance evaluation. The computation time to solve the routing problems is analyzed with other algorithms to illustrate the goodness of the algorithm.

In [24], a chaotic neural network is used to compute the delay-constrained multicast routing tree. Twelve different networks with different node size, links, destination nodes and delay bound is taken to evaluate the algorithm. The results mainly focused on the computation time.

A simple distributed algorithm that is both stabilizing and inherently stabilizing under a realistic model to route messages over all shortest node disjoint paths from a process to another in a n-dimensional hypercube network is proposed in [25]. Sequence of lemmas is given to recognize the time of the distributed algorithm.

For the broadcast scheduling problem, the approaches analyzed above where suffered from the trade-off between solution quality and running time. Computation time is an important factor to validate an algorithm. Therefore, an algorithm is still needed for broadcast scheduling problem that improves the solution quality in reduced computation time even for a large network. The objective of this work is to reduce the time slots and to maximize the total number of transmissions, in an acceptable execution time.

The WMN scheduler considered here is node activation in addition to the one-hop and two-hop with low connectivity information requirement. The scheduler identifies a schedule transmission so that the channel utilization is maximized with guaranteeing the QoS for all MSs. A scheduler with the three evolutionary algorithms are carried out and a series of simulations is conducted to evaluate the performance of the proposed MA in terms of solution quality and running time, and to verify its superiority over GA and IGA.

Genetic algorithms solve many search and optimization problems, effectively. However, they may drop into local optimal solutions or they may find the optimal solution by low convergence speed and GA blindly wanders over the search space. To overcome these problems, we used the immune concept to enhance the GA. Immune genetic algorithm gets the knowledge from hop matrix during vaccination process. IGA increases the number of transmission in a reduced time slot but not in a good computation time, MA reduces the processing time. Memetic algorithm is a blooming dialect of evolutionary algorithm (EA). In addition to Darwinism, MA adopts the Lamarckian theory that offspring can inherit the knowledge or characteristics that their parents acquire during their lifetime. The MA implements this idea by integrating a local enhancement, such as local search and repair operator, into the canonical EA, and making the enhancement inheritable, this integration significantly improves the exploitation ability of EA. In genetic algorithm, the mutation creates new genes for the population and the crossover operator orients seeking the best solution from the genes in the population. In memetic algorithm, this orientation is achieved by local search. Local search reduces the search space and reaches to high quality solution faster. MA actively aims on improving solution and explicitly concerned with exploiting all available knowledge about the problem.

The rest of this paper is organized as follows: Sect. 2 gives a formal definition of the problem, along with the constraints. In Sect. 3, we describe the three algorithm and its operators. The details of simulation results, comparison of time slot, channel utilization, average time delay and computation time by MA, IGA with other competitive algorithms are in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Representation of WMN scheduler

The WMN scheduler determines a collision free schedule with minimum TDMA frame length and maximum slot utilization by the nodes, in an acceptable running time. The scheduler assumes that each MS has network connectivity information within a two-hop radius. Initially, each MS allocated a time slot in a frame (e.g., MS i assigned in the ith time slot). MSs more than two hops away from the MS i also eligible to transmit during slot i. Some pre-established rule used to select an eligible MS to transmit in slot i. The selected MS sends a broadcast message to inform other MSs that are using slot i. The algorithm progresses in such a way that it allows as many MSs as possible to transmit in each slot. A node may interfere with another node, so these nodes should not transmit simultaneously. However, it does not ensure fair slot allocations among all MSs and is not topology transparent.

The WMN can be represented by undirected graph $G = (N, E)$ where N is the set of nodes and E is the links (transmission) assumed bidirectional. $|N|$ represents the number of nodes in the given network i.e., $|N| = \{n_1, n_2, \ldots, n_x\}$ and $|M|$ is number of time slots. In Fig. 1. $N = \{A, B, C, D, E, F, G, H\}$ and $|N| = 8$.

Connectivity matrix [CM] represents a direct link between the nodes, hop matrix [HM] says about the one-hop and two-hop connectivity information of each node, scheduler matrix [SM] is the allotted time slots of the given network without any interference.

$$[CM] = \begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

This connectivity matrix is identified for the network given in Fig. 1. Column represents nodes of the network and row represents the link existence between the nodes, i.e., the row one says about the connectivity information of node A, likewise for the remaining nodes. The matrix has the value 0 or 1, where 1 represents the existence of a link.

$$[HM] = \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \end{array}$$

This hop matrix is recognized for the network given in Fig. 1. Row value represents the one-hop and two-hop information between the nodes. The matrix takes the value 0 or 1, where 1 says low connectivity information for the node.

$$[SM] = \begin{array}{c} \begin{array}{cccccccc} A & B & C & D & E & F & G & H \end{array} \\ \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \end{array}$$

The optimal TDMA scheduler matrix generated for the network in Fig. 1. Row represents the number of time slots. It takes the value 0 or 1, where 1 represents the node allowed for transmitting in that time slot. In first time slot, nodes A and E allowed to send their packets without interference.

3 Evolutionary algorithms

The GA is a heuristic search technique that simulates the processes of natural selection and evolution [26]. John Holland first proposed Genetic Algorithms (GAs) in the 1960s. GAs are effective, robust search procedure for NP-complete problems [27]. The selection, crossover, mutation, fitness function and termination condition discussed below is common for all three algorithms.

The TDMA scheduler matrix is a $M \times N$ matrix where M is the number of time slots and N is the total number of nodes in the network. The scheduler matrix is represented as bit string chromosome containing 0s and 1s. Each row and column of the scheduler matrix represents to time slot and node transmission. The value 1 in the position (i, j) in the matrix indicates that jth node is allowed for transmission in the ith time slot. The initial TDMA frames are constructed using the Elite population method of Chakraborty [12]. GA, IGA and MA manipulate a set of chromosomes to search for an optimal solution.

The selection operators for parent selection and survivor selection follow the Darwinian principle of survival of the fittest. First, parent selection is for the reproduction process, ordinarily based on an alternative explanation of natural selection i.e., fitter individuals should have a higher probability of reproducing. This study performs k-tournament selection for parent selection, chooses the winner among k individuals that drawn randomly from the population. The number k controls selection pressure, a higher k gives higher selection pressure. Second, the survivor selection applies the principle of survivor of the fittest. Only the fittest individuals selected as parents for the next-generation. Idea of elitism is to retain some of the best individuals in each generation. In this study, a small percentage of best fitness individuals retained to the next generation. It increases the performance of algorithm, by preventing the loss of best found solution. From each generation 10 % of best solution retained to the next iteration.

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

The fitness function evaluates the quality (fitness) of candidate solutions. The fitness function for the scheduling problem is based on the variables channel utilization and tight lower bound. The termination point determines whether the best feasible solution is identified in that generation or not. The best feasible solution is the one, which satisfies both the criteria. When the generation of evolution reaches this termination point, the algorithm stops and outputs the optimal solution for the given network.

3.1 Genetic algorithm

After initializing the population, the selection operator picks two chromosomes from the population to serve as parent. The crossover operator then exchanges the

Algorithm 1 Genetic algorithm

```
initialize population GAPop;
evaluate GAPop;
while (not terminated)
{
    GAP_s = Select (GAPop);
    GAP_c = Crossover (GAP_s);
    GAP_m = Mutate (GAP_c);
    GAP' = evaluate GAP_m;
    GAPop = Survival (GAPop, GAP');
};
```

information between these two parents to produce their offspring. A predetermined crossover rate defines the probability of performing crossover. Mutation is performed with a probability, called mutation rate, to alter slightly some genes in the offspring. Algorithm 1 presents the framework of genetic algorithm.

The generated populations are evaluated with the fitness conditions. If the optimal solution is identified in the generation then the algorithm is terminated with the solution, else elitism method done on the populations and proceeds to the next generation. At the end of iteration, the 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 Immune genetic algorithm

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 the 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,

Algorithm 2 Immune genetic algorithm

```
initialize population IGAPop;
evaluate IGAPop;
while (not terminated)
{
    IGAP_s = Select (IGAPop);
    IGAP_c = Crossover (IGAP_s);
    IGAP_m = Mutate (IGAP_c);
    Immunization (IGAP_m)
    {
        IGAP_sel = ImmuneSelection (IGAP_m);
        IGAP_v = Vaccination (IGAP_sel);
    };
    IGAP' = evaluate IGAP_v;
    IGAPop = Survival (IGAPop, IGAP');
};
```

Immune Genetic Algorithm proposed. Algorithm 2 shows the structure of immune genetic algorithm. The solution after the reproduction stage 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 knowledge added IGA algorithm performed in the following way.

3.2.1 Immune selection

The newly created population after reproduction, which satisfies the primary and secondary constraints, is selected for duplicate row elimination. The resulting populations are arranged according to the channel utilization variable and stored in the vaccine pool.

3.2.2 Vaccination

Vaccination is used for improving the fitness by modifying the genes of an individual population with the prior knowledge to gain higher fitness with greater probability. A chromosome from vaccine pool is taken for vaccination. The IGA identifies the node transmits first in the population. During the same time slot, some other node, which does not create interference with the transmitting node, can be allowed to transmit in the same time slot. To perform this, a node is selected randomly and checked with the hop matrix whether it creates an interference with the currently transmitting node, if not the node value is mutated to one, allowing the selected node to transmit in the same time slot. The genes of the selected chromosome are modified based on the knowledge obtained from the hop matrix of the given network hence the vaccination process increases the number of transmissions.

3.3 Memetic algorithm

Memetic algorithms (MA) are extensions of evolutionary algorithms (EA) that apply local search processes in the agents and trying to improve their fitness [28–31]. Compared with other approaches, Memetic algorithms are superior, because of wide applicability. Despite the good results obtained by some MA, the process of designing efficient MA often depends on the problem-specific details. The construction of Memetic algorithm is given in Algorithm 3.

The initial population is constructed using the Elite population method and the parent selection for reproduction is done using *k*-tournament selection. On the selected chromosomes, a single point crossover operator is performed and the mutation operator is carried out based on the given mutation probability. After crossover and

Algorithm 3 Memetic algorithm

```
initialize population MPop;
evaluate MPop;
while (not terminated)
{    MP_s = Select (MPop);
     MP_c = Crossover (MP_s);
     MP_m = Mutate (MP_c);
     MemeticAlg (MP_m)
     {
          MP_op = Optimizer (MP_m);
          MP_I = Improver (MP_op);
     };
     MP' = evaluate MP_I;
     MPop = Survival (MPop, MP');
};
```

mutation, the following optimizer and improver is applied on the chromosomes in MA.

3.3.1 Optimizer

The optimizer phase of MA reduces the number of time slots by determining the channel utilization for each node. ρ_x is the performance of node x in the current population, i.e., the total number of transmissions carried out by the node x in the given time slot is identified using Eq. (2). The optimizer phase obtains each node transmissions, then it identifies whether the same node is transmitting in some other time slot j. In the jth time slot, if the nodes that are transmitted contains $\rho_x > 1$ then the row is removed from that population. In [12] the rows that are subset of a row generated after crossover are eliminated. In this paper, duplicate row elimination in GA and IGA performs reduction of time slots, if a time slot is repeated in that population, while in MA the optimizer performs reduction based on the value of ρ_x. This phase generates population with minimum number of time slots with the constraint that every node has transmitted at least once in that TDMA frame.

3.3.2 Improver

Improver is a greedy way of improving solution and it reduces the solution diversity. The populations from optimizer are taken to the improver phase where the total number of transmissions is increased in the reduced time slots. Obtaining knowledge from hop matrix, improver phase increases the number of transmissions after reduction of time slots. Since memetic algorithm operations are carried out in each iteration, the optimal solution is identified in less number of generations. Hence, running time of the algorithm is also reduced compared to other recently proposed competitive algorithms.

4 Simulation results

A series of simulations are carried to evaluate the performance of the proposed MA to solve the broadcast scheduling problem, in comparison with mean field annealing [3], GA [12] and competent permutation genetic algorithm [13]. The following sections discuss the simulation results regarding the number of nodes |N|, the number of timeslots |M| and the degree of networks. The fitness function factors are defined as,

1. Channel utilization variable

For the entire network:

$$\sigma = \frac{1}{|M| * |N|}\left(\sum_{i=1}^{|M|}\sum_{j=1}^{|N|}[\mathrm{SM}_{ij}]\right) \tag{1}$$

For each node:

$$\rho_x = \sum_{i=1}^{|M|}[\mathrm{SM}_{ix}] \tag{2}$$

$$\sigma_x = \frac{\rho_x}{|M|} \tag{3}$$

2. Tight lower bound

$$\mathrm{ND} = \max_{n \in N}|D(n)| \tag{4}$$

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

$$\Delta = |M| - \mathrm{ND} \geq 1 \tag{5}$$

If $\Delta = 1$ then the solution is optimal. The terminal conditions for the three algorithms discussed in this study are $\Delta = 1$ or the maximum number of generations, which is taken as 500 in all our experiments.

Various randomly generated networks with different degree and nodes test the three algorithms where each represents a multihop topology. For a particular setting of parameters, the algorithm is carried out for 150 times, the average value of the results is given in the following simulation results. The simulation results is based on the following parameters population size 50, maximum number of generations 500, crossover rate 0.30, mutation probability 0.001, and on these three measures,

1. Tight lower bound Δ value is one.
2. Channel Utilization variable to find the improvement in the number of transmission.

Table 1 Simulation results of genetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
15	25	3.3	4	6	7	0.219	30.6	0.80 s
30	49	3.3	4.8	8	9	0.156	27.2	01.10 min
80	156	3.9	5.8	8	9	0.154	238	16.08 min
100	200	4	7.5	9	10	0.104	422	32.00 min

Table 2 Simulation results of immune genetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
15	25	3.3	4	6	7	0.289	18.6	0.5 s
30	60	4	4	8	9	0.199	21.0	12 s
40	80	4	6	8	9	0.187	35.2	3.8 s
50	100	4	6	8	9	0.180	49.7	10.23 s
70	140	4	7	8	9	0.172	64.9	2.49 min
80	160	4	7	8	9	0.167	89.0	4.0 min
100	200	4	7	9	10	0.141	92.3	12.61 min
100	250	5	8	9	10	0.117	98.0	27.37 min

Table 3 Simulation results of memetic algorithm

No. of nodes	No. of links	Average degree	Avg. ND	Maximum ND	Minimum TDMA frame length	Avg. σ	Avg. no. of generations	Computation time
50	85	3.4	4	6	7	0.203	5.12	7 s
80	152	3.8	7	9	10	0.175	7.48	11 s
100	150	3	6.9	9	10	0.170	9.03	1.7 min
100	200	4	7.5	10	11	0.162	16.58	2.0 min
100	250	5	7	10	11	0.147	17.0	2.0 min
100	300	6	7	8	9	0.121	19.76	2.3 min
200	400	4	8	10	11	0.150	29.76	12.2 min
200	500	5	8	9	10	0.139	38.02	20.8 min
300	600	4	7	10	11	0.176	55.54	30.5 min
400	800	4	12	16	17	0.159	60.58	65.3 min
500	1,000	4	9	14	15	0.128	89.04	72.11 min

3. Running time is measured on a simulation platform that uses Matlab code on Windows XP/Intel Core 2 Duo T6600 2.2 GHz machine.

4.1 Simulation result of GA

The purpose of first simulation was to investigate the performance of genetic algorithm for different networks shown in Table 1. The number of nodes taken for simulation ranges from five to hundred. Smaller node networks executed with more number of transmission in an

Table 4 Comparison of MA with other familiar algorithms in terms of number of transmissions

No. of nodes	Time slots $\|M\|$	HNN-GA [8]	TS-GR [10]	BSP [21]	MA
15	8	20	20	20	28
30	10	35	37	37	48
30	11	40	43	–	51
30	12	47	48	–	54
40	8	67	68	69	76
40	9	77	77	–	84

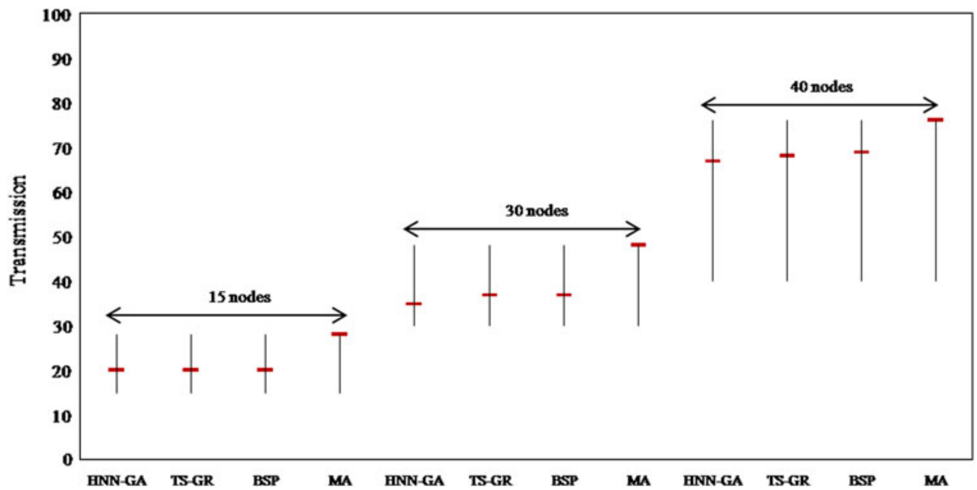

Fig. 2 Transmission comparison graph of existing algorithm with MA

acceptable generation. However, for a 100 node network with 200 edges with degree of nine identifies the optimal solution TDMA frame after 489 generations. The average number of generations for 100-node network is 422. This has to be reduced in order to reduce the execution time.

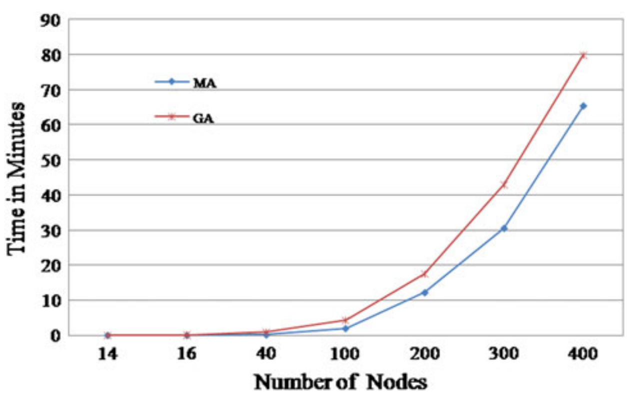

Fig. 3 Comparison of computation time taken by MA and GA [12]

4.2 Simulation result of IGA

Table 2 represents the output produced by immune genetic algorithm for varying number of nodes and edges. Compared to genetic algorithm, knowledge added IGA could improve the searching ability and adaptability, greatly increase the converging speed [32–34]. During vaccination process, the selected antigen is improved with more number of transmissions so that the channel utilization is increased. Comparing the simulation results of IGA in Table 2 with GA in Table 1, the number of generations is reduced, and the average number of transmission of each network is improved. For network with 80 nodes and 100 nodes, the solution is identified with acceptable generation. For a 100-node network with average degree of four and five, the optimal solution is identified in 16 min and 25 min. However, first two measures are satisfied by IGA, but third one, i.e., running time for a large network is not reduced.

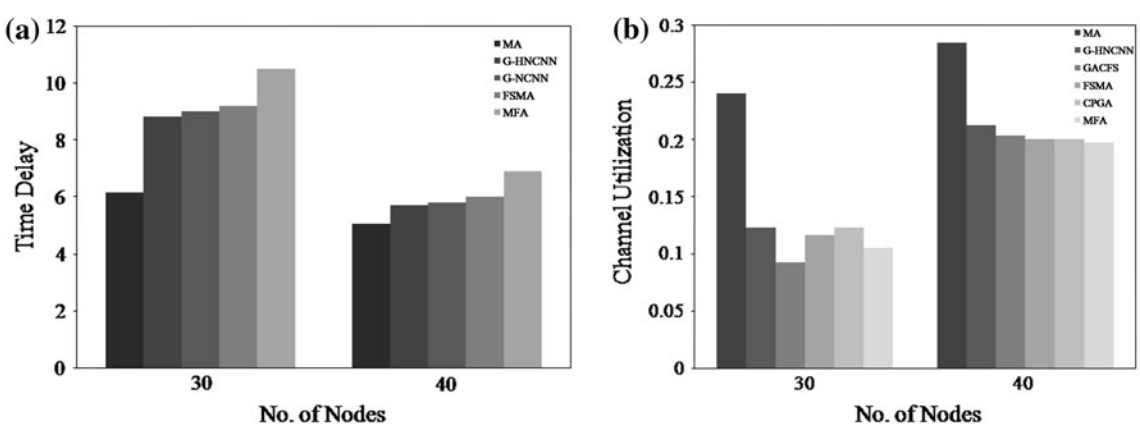

Fig. 4 **a** Comparison of average time delay, **b** comparison of channel utilization

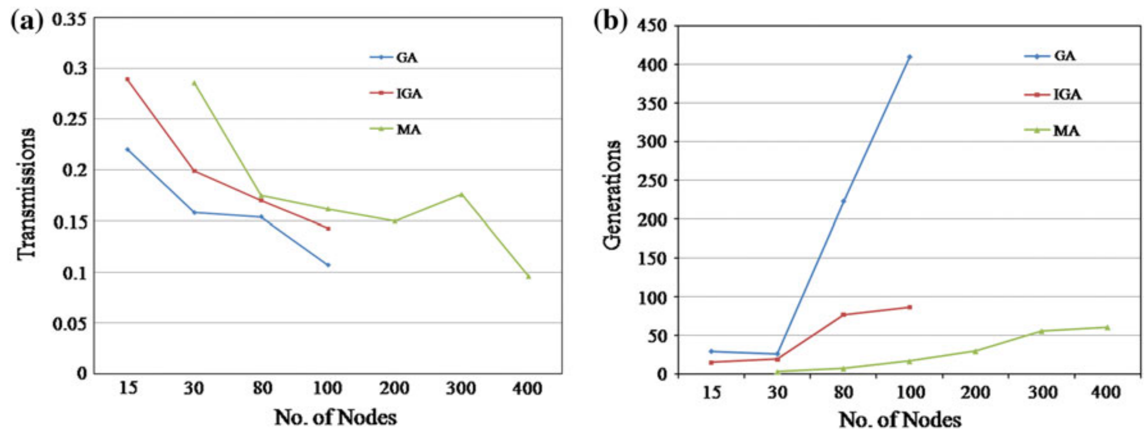

Fig. 5 **a** Comparison of total transmissions, **b** comparison of generations

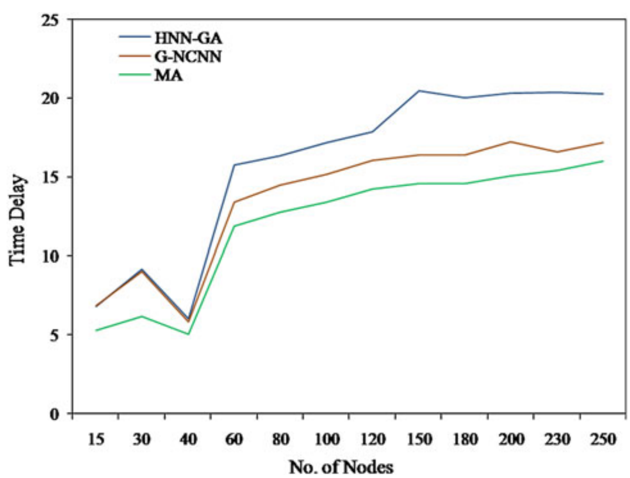

Fig. 6 Comparison of average time delay

4.3 Simulation result of MA

The methods discussed so far mainly focused on the convergence of the algorithms in terms of tight lower bound and increase in number of transmissions. Therefore, the question arises: what is the relation of these methods compared to each other in terms of time? This has been set as the main question of Memetic algorithm. The simulation result of MA aims to find the efficiency i.e., the speed of convergence.

It is clear from the Tables 1 and 2, IGA has improved the channel utilization in reduced number of generations also in less computation time while compared to GA. The average channel utilization, average number of generations and the computation time of various networks using MA for network with different degree is analyzed in Table 3. The time taken for 100-node network in MA is 2.0 min, is more efficient than the time taken by IGA for the same network is 12.61 min. For network with more than 100 nodes, computation time is not much efficient in IGA and other recently proposed efficient methods when we compare with MA. This is the main advantage of MA.

Total number of transmissions generated by MA for different node networks with different time slots is compared with HNN-GA [8], TS-GR [10] and BSP [21] is shown in Table 4. Transmission comparison graph of MA with these algorithms is also shown in Fig. 2. The starting point of vertical line in the graph represents lowest transmission value for given network, the ending point of the line represents highest transmission value and a small horizontal line represents number of transmissions generated by each algorithm. The Table and Figure proves that MA produces higher number of transmissions for varying networks compared to existing algorithms and produces with difference of 6–11 in total number of transmissions. The computation time and number of generations to identify optimal solutions are reduced where as channel utilization is increased in MA compared to GA [12]. The comparison of average time taken by MA and modified genetic algorithm [12] is given in Fig. 3.

Two benchmark problems discussed in [3] are solved using MA and the results are compared with other algorithms such as, gradual hysteretic noisy chaotic neural network G-HNCNN [15], gradual noisy chaotic neural network G-NCNN [22], co-evolutionary genetic algorithm for collision free set GACFS [17], the finite state machine based algorithm FSMA [14], the competent permutation genetic algorithm CPGA [13] and mean field annealing algorithm MFA [3] is shown in Table 5. 30 nodes with 70 edges is analyzed in instance #1 and 40 nodes with 66 edges is analyzed in instance #2 by considering the number of time slots, channel utilization and time delay. The time delay is calculated using

$$\text{Time delay} = \frac{|M|}{|N|}\sum_{i=1}^{|N|}\left(\frac{1}{\sum_{j=1}^{|M|}\left|\mathrm{SM}_{ij}\right|}\right)$$

As seen in Table 5, MA increases the channel utilization with the smallest time delay this indicates that MA

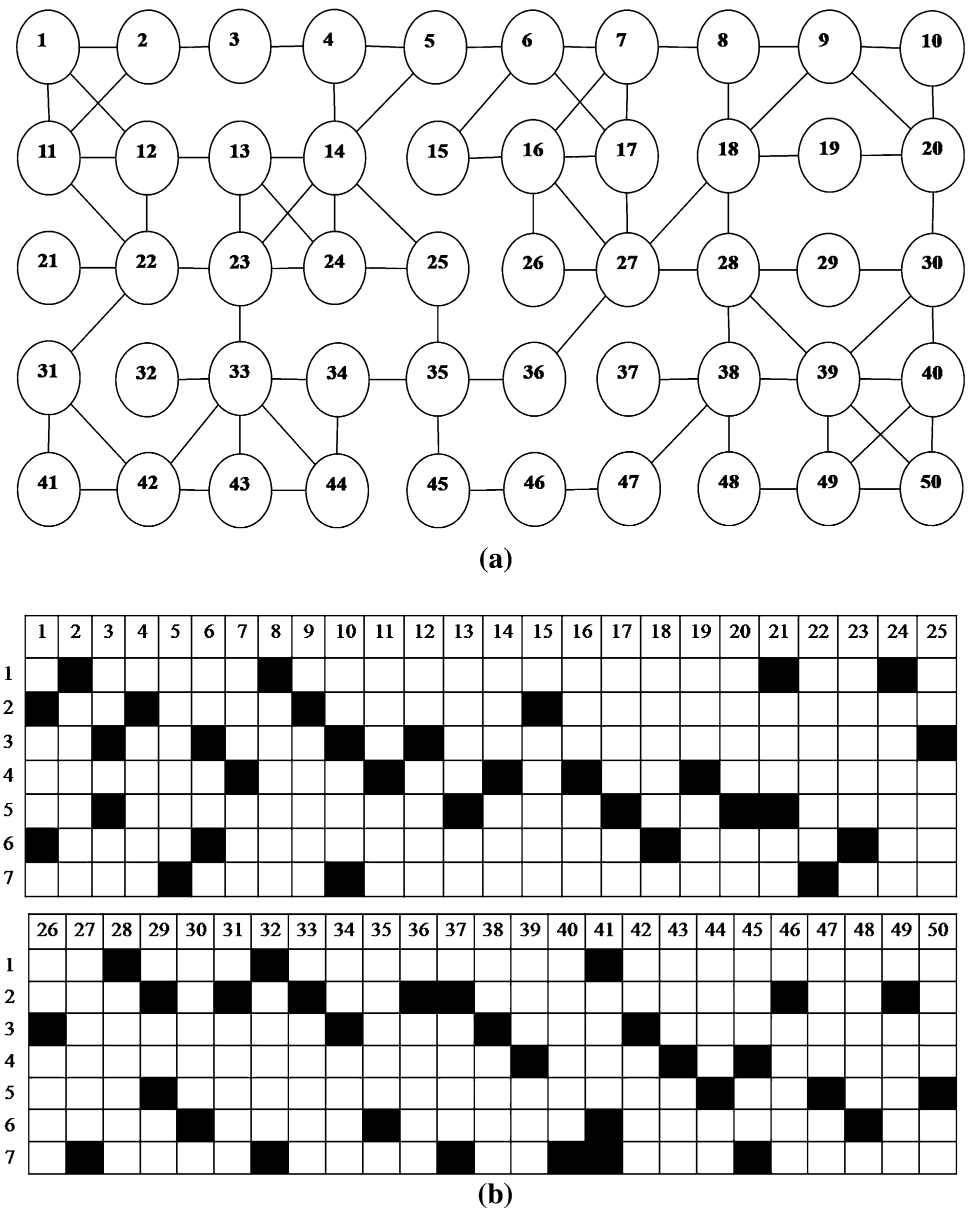

Fig. 7 **a** Network with 50 nodes, 85 edges and the average degree of 3.4, **b** optimal solution found by MA for the same network

Table 5 Comparison of MA with other competitive algorithms in terms of time slot $|M|$, channel utilization σ and time delay

Instance	Parameter	MA	G-HNCNN	G-NCNN	GACFS	FSMA	CPGA	MFA
#1	$\|M\|$	10	10	10	10	10	10	12
	σ	0.24	0.1233	–	0.093	0.1167	0.1233	0.1056
	Time delay	6.1529	8.83	9.0	–	9.2	–	10.5
#2	$\|M\|$	8	8	8	8	8	8	9
	σ	0.2844	0.2125	–	0.203	0.200	0.200	0.197
	Time delay	5.0433	5.7056	5.8	–	6	–	6.9

performs efficiently when compared to the other recently proposed algorithms. Figure 4 compares the time delay and channel utilization of various algorithms with MA.

Table 6 compares the computation time of MA with algorithms HNN-GA [8] and BSP [21] and shows that computation time is greatly reduced.

Table 6 Comparison of MA with other competitive algorithms in terms of computation time

No. of nodes	Average degree	HNN-GA [8] (s)	BSP [21] (s)	MA (s)
10	3	5.1	0.01	0.01
20	3	12.97	0.06	0.01
30	3	85.4	0.07	0.02
40	3	165.12	0.3	0.14
50	3	194.4	0.64	0.27
10	4	5.24	0.01	0.01
20	4	20.04	0.33	0.01
30	4	152.06	1.45	2.04
40	4	280.37	4.74	3.41
50	4	320.06	14.2	7.00

Table 7 Comparison of MA with SLBIP [23] in terms of computation time

No. of nodes	No. of links	SLBIP [23] (s)	MA (s)
6	24	0.031	0.011
12	36	1.015	0.073
12	48	3.119	1.492
15	56	16.558	8.961
20	68	6.289	2.133
35	100	1.613	0.918

Figure 5(a) compares number of transmissions generated by GA, IGA and MA with total number of nodes. Figure 5(b) compares number of generations taken by GA, IGA and MA with number of nodes of various networks. These results illustrate that memetic algorithm performs efficiently in terms of tight lower bound and increases the channel utilization in an acceptable computation time.

The average time delay taken by HNN-GA [8], G-NCNN [22] and MA for various networks ranging from 15 to 250 is compared in Fig. 6. If there is more number of transmissions then there is a decrease in the time delay, from Fig. 6 also identified the total number of transmission produced by MA is high when compare to other two algorithms.

A sample 50-node network with 85 edges with average degree of 3.4 and its corresponding optimal TDMA frame identified using MA is shown in Fig. 7(a, b). Since maximum degree of the network is six MA produces optimal schedule in seven time slots with 61 transmissions. Channel utilization for this network is 0.174 and average time delay is 6.27.

In Table 7, the computation time of MA for varying number of nodes and links is compared with SLBIP [23]. From all the Tables and Figures it is identified that MA performs better compared not only to GA and IGA but also with the recently proposed algorithms as discussed earlier.

5 Conclusion

The basic genetic algorithm, knowledge added immune genetic algorithm and a domain specific memetic algorithm are discussed to solve wireless multihop network broadcast scheduling. Compared to GA and IGA, MA actively aims on improving solutions, while GA blindly wanders over the search space. MA exploits all available knowledge about the problem, while immune genetic algorithm gets knowledge from hop matrix during vaccination process. IGA increases the number of transmissions in a reduced time slot but not in a good computation time, MA overcomes it. In previous papers, the main drawback quoted by authors was computation time for a large network, which is greatly reduced in this paper. The simulation results confirm the advantages of MA in terms of channel utilization, number of generations, and running time. MA achieves the tight lower bound in shorter running time compared with other algorithms. The outcome validates the effectiveness and efficiency of MA for the broadcast scheduling problem. Further enhancement can be done to reduce computation time even for large networks with more than 500 nodes.

Acknowledgments We gratefully acknowledge Department of Science and Technology, INDIA for providing financial support to carry out this research work under PURSE scheme. We would like to thank sincerely the Editor and the Reviewers for their constructive comments and suggestions that helped to improve the manuscript significantly.

References

1. Lloyd, E. L. (2002). Broadcast scheduling for TDMA in wireless multihop networks. In *Handbook of wireless networks and mobile computing* (pp. 347–370). New York: Wiley.
2. Ephremides, A., & Truong, T. V. (1990). Scheduling broadcasts in multiple radio networks. *IEEE Transactions on Communications, 38*(4), 456–460.
3. Wang, G., & Ansari, N. (1997). Optimal broadcast scheduling in packet radio networks using mean filed annealing. *IEEE Journal on Selected Areas in Communications, 15*(2), 250–260.
4. Ju, J., & Li, V. O. K. (1998). An optimal topology-transparent scheduling method in multihop packet radio networks. *IEEE/ACM Transactions Networking, 6*(3), 298–306.
5. Wang, C.-Y., Atkinson, M. W., Fertig, K. W., & Sastry, A. R. K. (1986). Performance evaluation of multi-hop packet radio networks using simulation. In *MILCOM '86—Military Communications Conference*, Monterey, CA, October 1986, pp. 28.5.1–28.5.6.
6. Funabiki, N., & Takefuji, Y. (1993). A parallel algorithm for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Communications, 41*(6), 828–831.
7. Goutam, C., & Hirano, Y. (1998). Genetic algorithm for broadcast scheduling in packet radio networks. In *Proceedings of IEEE World Congress Computational Intelligence*, Anchorage, AK, May 1998, pp. 183–188.
8. Salcedo-Sanz, S., Bousono-Calzon, C., & Figueiras-Vidal, A. R. (2003). A mixed neural-genetic algorithm for the broadcast scheduling problem. *IEEE Transactions on Wireless Communications, 2*(2), 277–283.

9. Ngo, C. Y., & Li, V. O. K. (2003). Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms. *IEEE Transactions on Communications, 51*(9), 1439–1441.
10. Peng, Y., Soong, B. H., & Wang, L. (2004). Broadcast scheduling in packet radio networks using mixed tabu-greedy algorithm. *Electronics Letters, 40*(6), 375–376.
11. Shi, H., & Wang, L. (2005). Broadcast scheduling in wireless multihop networks using a neural-network-based hybrid algorithm. *Neural Networks, 18*, 765–771.
12. Chakraborty, G. (2004). Genetic Algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks. *IEEE Transactions on Communications, 52*(5), 765–777.
13. Wu, X., Sharif, B. S., Hinton, O. P., & Tsimenidis, C. C. (2005). Solving optimum TDMA broadcast scheduling in mobile ad hoc networks: A competent permutation genetic algorithm approach. *IEE Proceedings: Communications, 152*(6), 780–788.
14. Ahmad, I., Al-Kazemi, B., & Das, A. S. (2008). An efficient algorithm to find broadcast schedule in ad hoc TDMA networks. *Journal of Computer Systems, Networks, and Communications, 12*, 1–10.
15. Sun, M., Zhao, L., Cao, W., Xu, Y., Dai, X., & Wang, X. (2010). Novel hysteretic noisy chaotic neural network for broadcast scheduling problems in packet radio networks. *IEEE Transactions on Neural Networks, 21*(9), 1422–1433.
16. Chakrabortya, G., Chakraborty, D., & Shiratori, N. (2005). A heuristic algorithm for optimum transmission schedule in broadcast packet radio networks. *Computer Communications, 28*, 74–85.
17. Gunasekaran, R., Siddharth, S., Krishnaraj, P., Kalaiarasan, M., & Rhymend Uthariaraj, V. (2010). Efficient algorithms to solve broadcast scheduling problem in WiMAX mesh networks. *Computer Communications, 33*, 1325–1333.
18. Ramanathan, S., & Lloyd, E. L. (1993). Scheduling algorithms for multihop radio networks. *IEEE/ACM Transactions on Networking, 1*(2), 166–177.
19. Yeo, J., Lee, H., & Kim, S. (2002). An efficient broadcast scheduling algorithm for TDMA ad-hoc networks. *Computers & Operations Research, 29*(13), 1793–1806.
20. Bi, W., Tang, Z., Wang, J., & Cao, Q. (2005). An improved neural network algorithm for broadcast scheduling problem in packet radio. *Neural Information Processing-Letters and Reviews, 9*(1), 23–29.
21. Menon, S. (2009). A sequential approach for optimal broadcast scheduling in packet radio networks. *IEEE Transactions on Communications, 57*(3), 764–770.
22. Wang, L., & Shi, H. (2006). A gradual noisy chaotic neural network for solving the broadcast schedule problem in packet radio networks. *IEEE Transactions on Neural Networks, 17*(4), 989–1000.
23. Oki, E., & Iwaki, A. (2010). Load-balanced IP routing scheme based on shortest paths in hose model. *IEEE Transactions on Communications, 58*(7), 2088–2096.
24. Wang, L., Liu, W., & Shi, H. (2009). Delay-constrained multicast routing using the noisy chaotic neural networks. *IEEE Transactions on Computers, 58*(1), 82–89.
25. Sinanoglu, O., Karaata, M. H., & Albdaiwi, B. (2010). An inherently stabilizing algorithm for node-to-node routing over all shortest node-disjoint paths in hypercube networks. *IEEE Transactions on Computers, 59*(7), 995–999.
26. Michalewicz, Z. (1995). *Genetic algorithms + data structure = evolution programs*. New York: Springer.
27. De Jong, K. A., & Spears, W. M. (1989). Using genetic algorithms to solve NP-complete problems. In *Proceedings of the 3rd International Conference on Genetic Algorithms* (pp. 124–132).
28. Banos, R., Gil, C., Reca, J., & Montoya, F. G. (2010). A memetic algorithm applied to the design of water distribution networks. *Applied Soft Computing, 10*, 261–266.
29. Krasnogor, N., & Gustafson, S. (2002). Toward truly memetic algorithms: Discussion and proof of concepts. In *Proceedings of PPSN VII*.
30. Moscato, P. (1999). *Memetic algorithms: A short introduction new ideas in optimization* (pp. 219–234). New York: McGraw-Hill.
31. Lu, Z., & Hao, J. K. (2010). A memetic algorithm for graph coloring. *European Journal of Operational Research, 203*, 241–250.
32. Liu, M., Pang, W., Wang, K. P., Song, Y. Z., & Zhou, C. G. (2006). Improved immune genetic algorithm for solving flow shop scheduling problem. In *Computational methods* (pp. 1057–1062). Springer.
33. Jiao, L., & Wang, L. (2000). A novel genetic algorithm based on immunity. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems And Humans, 30*(5), 552–561.
34. Wang, D., Fung, R. Y. K., & Ip, W. H. (2009). An immune-genetic algorithm for introduction planning of new products. *Computers & Industrial Engineering, 56*, 902–917.

Author Biographies

D. Arivudainambi is currently an Associate Professor in the Department of Mathematics, Anna University, Chennai. He had been at University of Toronto, Toronto. His main research interests include network optimization, network models, queueing models with computer applications and communication systems. He is reviewer of many journals including IEEE, Elsevier, Springer and Wiley. He published many papers in reputed journals. He served as General and Technical Program Chair of numerous conferences.

D. Rekha received her B.Sc. and M.Sc. degrees in computer science from Bharathiyar University in 2001 and 2003 respectively. She is currently working towards the Ph.D. degree in Anna University, Chennai. Her research areas cover wireless multihop networks and evolutionary algorithms.