Improved Genetic Algorithm for the Bandwidth Maximization in TDMA-based Mobile Ad Hoc Networks

A. Sirbu[1], G. Breaban[1], I. Cleju[1], I. Bogdan[1]
[1]*Faculty of Electronics, Telecommunications and Information Technology,*
"Gheorghe Asachi" Technical University of Iaşi,
B-dul Carol I, no 11, 700506, Iaşi, Romania
asirbu@etti.tuiasi.ro

***Abstract*—The paper proposes and evaluates an improved genetic algorithm (GA) dedicated to maximize the bandwidth of a specified, previously discovered, route in a Time Division Multiple Access (TDMA)-based Mobile Ad Hoc networks (MANETs). The objective is also to increase the speed of convergence of the GA by adequately choosing the genetic operators, as well as the crossover and mutation probabilities. Simulation results show that the proposed solution outperforms the existing ones, with acceptable computational costs.**

***Index Terms*—Genetic algorithms, scheduling algorithm, time division multiplexing, wireless networks.**

I. INTRODUCTION

The design of routing algorithms in ad hoc networks has attracted extensive research efforts aiming at finding the most appropriate path between a source node and a destination one, a quite challenging task due to their fast changing topologies. Most of the proposed routing algorithms were focused on discovering the shortest path and, until recently [1], [2], did not address issues related to the end-to-end Quality of Service (QoS) requirements of the involved communication services [3]. It was proved that the calculation of the available bandwidth of a path in a TDMA-based ad hoc network is NP-complete (nondeterministic polynomial time) [4] and the optimal solution could not be found in a realistic time interval. As a consequence, efforts were directed towards finding efficient heuristic algorithms for the estimation of the maximal available bandwidth along a discovered path.

The objective of this work is to devise an efficient genetic algorithm (GA) able to maximize the bandwidth of a previously discovered path in a TDMA-based ad hoc network, provisioning resources along the entire path in accordance with the service requirements. We investigate the possibility to improve the performances of GA used to solve the bandwidth calculation problem in TDMA-based ad hoc networks.

Simulation results show that the proposed solution outperforms existing algorithms, with acceptable computational costs.

Manuscript received July 2, 2012; accepted April 3, 2013.

II. TDMA SCHEDULING PROBLEM

A TDMA communication network is very demanding in terms of packet scheduling because transmissions from different nodes are very likely to collide if adequate precautions are not taken into consideration and the hidden terminal problem is not correctly addressed. Hence, more coordination among nodes is required. In our study it is assumed that all the nodes share a single common channel.

Several attempts have been made to solve the problem of bandwidth calculation in TDMA-based MANETs [5]–[8].

In [5] the design of an efficient heuristic algorithm for the calculus of the maximal available bandwidth along a path in a TDMA-based MANET is presented. The available bandwidth is calculated in a centralized manner as it has been proved that the hop-by-hop calculation yields usually a smaller bandwidth than the one obtained by a centralized technique [6]–[8].

One of the first and most valuable approaches to solve optimum TDMA transmission schedule in broadcast packet radio (PR) networks using GA is that in [9]. For an *n*-node PR network, a chromosome is represented by the TDMA frame itself that is an *nxn* binary matrix. A special crossover operator was devised so that invalid solutions are not created. The basic optimization objective was to determine the smallest length TDMA frame, for the case when many nodes are allowed to transmit simultaneously, conflict free, in a single time slot.

In a TDMA-based ad hoc network, the transmission time is divided in fixed length frames consisting of a fixed number of slots. Each of the time slots carries one packet. The bandwidth requirement is fulfilled by reserving time slots on links. Assuming a half-duplex operation mode, a node cannot transmit and receive simultaneously. Each node marks a slot in a frame with either "U," if the slot is unavailable or "F," if the slot is free (available). All the slots that are free at both ends of a link form the set of paired free slots of the link. The available *bandwidth of a link* is the number of slots in this set. In a TDMA-based ad hoc network, in addition to the constraint of the half-duplex operation mode, it is compulsory to take into account the radio interference produced by hidden terminals, as the use of a free slot on a link is dependent on the status of that slot in the neighboring links. The *bandwidth of a path* between two nodes is defined as the minimum number of reserved slots for a link among the links included in the path. The reserved slots are selected from the set of the paired free slots taking into account the constraints imposed by the half duplex operation and the presence of hidden terminals. Hence, the path bandwidth is not simply the minimum link bandwidth along the path; its finding is an NP-complete problem.

The status of each time slot for each host in a path can be represented as a node state matrix. The status of each time slot for each link can be described by a link state matrix (denoted as the *free slot matrix*), whose element *(i, j)* is marked with “F”, if the slot *j* is marked as free (available) by both of the nodes adjacent to the link *i*, and marked with “U” (unavailable), otherwise.

Given a *free slot matrix*, one can construct a *slot schedule matrix* containing for every link the slots which have been reserved for the communication.

For numerical manipulation of the state matrix the elements corresponding to unavailable slots are marked with “2”. A free slot can be reserved for a packet transmission along the corresponding link and the associated state matrix element is changed to “1”. A slot marked with “F” can be reserved only by one of the adjacent links due to the assumed half-duplex operation mode.

So, during the slot reservation process, any time a link reserves a slot and marks it with “1”, the adjacent nodes should mark the same slot with “0”, as it becomes unreservable for them.

Considering, for instance, a path with five links and a frame with eight time slots, the node state matrix is a 6 x 8 matrix (Fig. 1(a)), whereas the link state matrix is a 5 x 8 matrix (Fig. 1(b)).

For the state matrix associated to the case presented in Fig. 1(a) the sets of free slots for the individual links are (Fig. 1b): {*2, 3, 6, 7*} for link *a – b*, {*1, 3, 4, 6, 7*} for link *b – c*, {*1, 3, 5, 7*} for link *c – d*, {*1, 2, 3, 5, 7, 8*} for link *d – e*, and {*2, 4, 5, 6, 8*} for link *e – f*, with link bandwidth equaling 4, 5, 4, 6, and 5, respectively.

Based on the link state matrix in Fig. 1(b) one can obtain different values for the path bandwidth for different slot reservation algorithms: for instance, 1 slot (Fig. 1(c)) or 2 slots (Fig. 1(d)).

Besides a greater path bandwidth, the reservation scheme in Fig. 1d yields also a more balanced distribution of reservable slots among the path nodes. The two reservation schemes use different total numbers of slots in a TDMA frame: 15 in Fig. 1(c) and 13 in Fig. 1(d).

A well designed slot reservation algorithm should yield a total number of reserved slots as close as possible to the total free slots in a frame (24 in the above presented example, Fig. 1(b).

As it was shown that GA techniques could be successfully used to obtain efficient near-optimal solutions for complex NP-complete optimization problems, including TDMA scheduling problems, we propose a heuristic reservation algorithm; its performances are evaluated through computer simulations.

The theory of genetic algorithm suggests that heuristic optimization problems are extremely sensitive to the appropriate design of solution representation, genetic operators and parameter settings (probabilities of crossover, mutation).

In this study we propose an improved GA and, through simulation experiments, demonstrate that we obtain a larger bandwidth, as compared to the results in [5] and [8], in a smaller number of generations as compared to [5].

		Slot							
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*
Node	*a*	U	F	F	U	F	F	F	U
	b	F	F	F	F	U	F	F	F
	c	F	U	F	F	F	F	F	U
	d	F	F	F	U	F	U	F	F
	e	F	F	F	F	F	F	F	F
	f	U	F	U	F	F	F	U	F

(a)

		Slot								No. of free slots
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	
Link	*a-b*	U	F	F	U	U	F	F	U	*4*
	b-c	F	U	F	F	U	F	F	U	*5*
	c-d	F	U	F	U	F	U	F	U	*4*
	d-e	F	F	F	U	F	U	F	F	*6*
	e-f	U	F	U	F	F	F	U	F	*5*

(b)

		Slot								No. of reservable slots
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	
Link	*a-b*	2	1	0	2	2	1	0	2	*2*
	b-c	1	2	1	1	2	0	1	2	*4*
	c-d	0	2	0	2	1	2	0	2	*1*
	d-e	1	1	1	2	0	2	1	1	*5*
	e-f	2	0	2	1	1	1	2	0	*3*

(c)

		Slot								No. of reservable slots
		1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	
Link	*a-b*	2	1	1	2	2	0	1	2	*3*
	b-c	0	2	0	1	2	1	0	2	*2*
	c-d	1	2	1	2	0	2	1	2	*3*
	d-e	0	0	0	2	1	2	0	1	*2*
	e-f	2	1	2	1	0	1	2	0	*3*

(d)

Fig. 1. Example of slot allocation for a 6 nodes path and an 8 slots TDMA frame: a – Sample of a node state matrix for a 6 node path and an 8 slot TDMA frame; b – Link state matrix corresponding to the above node state matrix; c – Possible slot reservation, path bandwidth = 1 slot; d – Possible slot reservation, path bandwidth = 2 slots.

III. GA FOR TDMA –BASED MANET BANDWIDTH MAXIMIZATION

Genetic algorithms are well-known optimization techniques based on the principles of evolution and natural genetics, performing search operations in complex, large and multimodal domains and providing near-optimal solutions for the objective (fitness) function of an optimization problem [10].

The evolutionary algorithm maintains a population of individuals, denoted as chromosomes, representing potential solutions of the optimization problem. One has to devise a method to give some measure of the adequacy of an individual for the given problem, a so called *fitness*. A new population is formed by appropriately selecting the more fitted individuals, transforming them by means of genetic operators to form new solutions. There are unary transformations, mutations, which create new individuals by a small change in a single individual and higher order transformations, such as crossover, which create new individuals by combining parts from several individuals.

After generating the offspring, a generation replacement strategy must be proposed to keep the population size constant. There are two approaches: the individuals in the current population are completely replaced by the offspring, making the best chromosome of the population fail to reproduce offspring in the next generation, or there is an elitist group from the current population which "survives" in the next generation. If the elitist rate is β, a proportion of 1−β fittest individuals of the offspring are selected, where β ($0 < \beta < 1$) is an important parameter affecting the convergence property [11]. After several generations the algorithm converges and the best individual represents a near-optimum solution. Usually the optimization algorithm will be terminated if a specified maximum generation number is reached or the best individual keeps invariant for a sufficiently large number of successive generations.

In order to set-up the GA optimization framework we have to follow the necessary stages, implying to devise [11]:

1) a genetic representation of solutions to the problem,
2) an evaluation function to rate the solutions in terms of their fitness,
3) selection methods to decide on the individuals which will produce offspring,
4) genetic operators that alter the genetic composition of individuals during reproduction,
5) replacement methods in order to create a new population.

Further on we have to establish a way to create an initial population of solutions, to set the values for the GA parameters (population size, selection method, probabilities of applying genetic operators) and to take care of infeasible solutions.

In order to address the representation of the solutions to the problem of bandwidth maximization in TDMA-based MANETs we make the assumption that the total number of time slots is constant throughout the transmission process.

B. Chromosome representation

Several strategies for adequately representing the chromosomes in TDMA broadcast scheduling related GA optimization algorithms have been reported. In [12] a binary matrix representation based on within-two-hops connectivity matrix is presented. Real number representation and arithmetic real-coded variation operators tailored for time slot and turn optimization on TDMA-scheduled resources with evolutionary algorithms are used in [9].

The principle of chromosome representation we used in this study is inspired from the one described in [5]. Each individual is associated to an m_1xm_2 matrix, M_c, corresponding to the *slot schedule matrix*, where m_1 is the number of hops and m_2 is the number of time slots of each link along the path. Each position (*i,j*) of the matrix is obtained from the *free slot matrix* in accordance with the process described in the previous section. The value of each gene, which in fact corresponds to a slot, has only one of the following values: 0 – if the slot is unreservable, '1' – if the slot is reservable or '2' – if the slot is unavailable. Due to the constraints imposed by the TDMA based MANET (half-duplex and interference), a valid individual is represented by a *slot schedule matrix* containing at most one '1' in any three consecutive rows of each column. The path bandwidth is defined as the minimum of the number of reserved slots in each row.

The input of the GA is the *free slot matrix* of a given path. From this matrix one generates a set of chromosomes to form the initial generation, taking into account the previously mentioned restrictions. So, the derivation procedure is as follows: scan each element in each row of the given matrix, from the first to the last element. If an entry is marked with '0' ('unreservable'), then its value is reassigned to '2' ('unavailable'), otherwise, its value is randomly reassigned to '1' or '0'. As explained before, for the case when an element of the chromosome (matrix) is assigned to '1' ('available'), it has to be checked that at most one of the two upstream consecutive entries is '1'. Otherwise, the assigned value is not valid and it has to be changed to '0'.

C. Fitness function

Evaluation plays a very important role in the GA process. Fitness function, effectively, maps all the properties of an individual to a number - giving it a rank and a place among the other individuals in the pool [10]. In our study, the fitness function, similar to that in [5], computes a value connected to the bandwidth of the path. Denoting by $\hat{r}_i$ the number of the elements of the chromosome matrix Mc with value '1' in each row and by $\hat{M}_C = \min\{\hat{r}_i\}, i = 1..m_1$, the fitness function is defined as

$$f(\hat{M}_c) = (\hat{M}_c + 1)\ x\ (\hat{M}_c + 1). \quad (1)$$

Obviously, a higher fitness value for a given individual implies a larger bandwidth corresponding to the evaluated path.

D. Selection and reproduction

It is well known that in order to generate good offspring, a good parent selection mechanism is necessary. For each generation, a proportion of the existing population is selected to breed a new generation. Individual solutions are selected through different processes, where fitter solutions (as measured by a fitness function) are typically more likely to be selected. There are several well-known selection procedures [10], [12]: Roulette wheel selection (RWS), Tournament selection (TS), Stochastic universal sampling (SUS).

For the proposed optimization problem, both RWS and TS proved to be adequate. In order to decrease the number of generations necessary for algorithm convergence we have adapted, tested and proved to be effective for this application the Emperor-selective method, EMS [13]. This mating scheme sorts the individuals in a descending order of fitness. The best individual (emperor) mates with the second best and then, starting from the third fittest, from every pair of adjacent fitted individuals one is selected randomly and is mated with the emperor. The method allows the fittest individual to mate preferentially with practically the rest of the population. As it was reported that EMS mating scheme could converge prematurely, we have investigated different methodologies to circumvent this drawback. So, first, we have modified the classical EMS, selecting the emperor mating partners by Tournament selection, but a better solution proved to be that of mixing classical evolution of the GA, using TS, with EMS, after a predetermined number of generations, defined as percentage of the maximum number of generations. In this way we have obtained solutions very closed to the optimal ones, but in a significantly smaller number of generations.

In order to produce a new generation, a pair of "parent" solutions selected for breeding, produce "child" solutions using genetic operators (crossover and mutation), the new solutions typically sharing many of the characteristics of its "parents".

As mentioned in [11] the existence of feasible and unfeasible individuals in a population can influence the GA in different stages such as: the processing/evaluation of unfeasible chromosomes or the elitist selection. There are several ways to approach the problem of unfeasible individuals, which is far from trivial. The “death penalty” heuristic, popular in many evolutionary algorithms, implies the rejection of unfeasible individuals, simplifying in this way the algorithm. Another method is to devise a technique to “repair” the unfeasible individuals, transforming them in feasible ones. Finally, maybe the most complex approach is the design of the evaluation function for unfeasible individuals – including an appropriate penalty. In this paper we present the results obtained using the second method: replacement of the unfeasible individuals (slot schedule matrices) by the corresponding “repaired” ones.

The crossover operator consists of two operations [5]: column exchange and row exchange, according to the crossover probability. The columns and rows to be exchanged are randomly selected. The operation of row exchange can generate unfeasible solution, that is, matrices in which there is more than one ‘1’ within any three consecutive values of the same column. In this case, the error is fixed by changing to ‘0’ the value of the elements in the nonselected rows.

A similar problem occurs when designing the mutation operator. The random changing of an element of the chromosome matrix, according to the mutation probability, should provide a valid individual. So, after performing each mutation, all the two upstream and downstream elements of the same column should be checked. If the imposed restriction of having at most one ‘1’ in three consecutive elements of the same column is not fulfilled, the appropriate element in a nonselected row is set to ‘0’.

The newly obtained individuals are selected to form the next generation [10]. In addition, elite-preservation strategy is employed, in which the best individuals are kept during the generations, in different proportions. For each iteration, these processes produce a next generation of chromosomes. It is proved that, generally, the average fitness of the population increases by this procedure, since only the best individuals from the previous generation are selected for breeding, along with a small proportion of less fit solutions.

The simulation studies we have carried out showed that the chromosome matrix representation we have chosen combined with the TS outperforms the results reported in [5] and [8]. The mixing between classical GA and the EMS mating scheme provides near optimal solutions in significantly smaller number of generations.

IV. SIMULATIONS AND RESULTS

For the implementation of the proposed algorithm we have used GaLib, A C++ Library of Genetic Algorithm Components [14], implemented in Visual Studio 2010. The GaLib is a powerful open source C++ library, containing a robust set of classes for genomes and genetic algorithms. It also provides facilities for setting the algorithm parameters.

The library uses mainly two classes: a genome and a genetic algorithm. Every genome instance represents a single solution to the given problem. The genetic algorithm defines the way in which the evolution will take place. It uses an objective function to determine how “fit” every genome is for survival. Genome operators (implemented as methods of the Genome class) and selection strategies (implemented as methods of the Genetic Algorithm class) are used to generate new individuals.

For the bandwidth calculation problem, two classes were derived: one for the genome and the other one for the genetic algorithm. The genome class was derived from the *GA2DArrayAlleleGenome* class which uses a bidimensional array to represent the genome, each gene having a predefined allele set. For the TDMA genome, the allele set comprises only 3 values: {0, 1, 2} which represent the three described states of a time slot. Specific genetic operators were defined for the TDMA genome as static methods which ensure that the resulted genome is a valid one that can be further used as a slot schedule matrix.

The second derived class implements the genetic algorithm. Within this class, a *step* function was defined that applies the genetic operators to the current population and obtains the next population of individuals. Implementation of this function is specific for each of the tested mating strategies. Additionally, an objective function was defined to evaluate each genome according to the proposed fitness function formula (1).

The simulation experiments aimed to study the performances of the proposed algorithm with various GaLib operator parameters. As argued in the previous section, the solution as well as the convergence properties of a GA strongly depends on the adequate choosing of algorithm parameters. The settings of these parameters are generally ad hoc. The success of genetic search also depends on the optimum balancing between the population diversity defining the search space and the selection pressure to get the optimum point fast [15].

As presented in the previous section, the input of the algorithm is a randomly generated *common free slot matrix*, adequately converted into a slot schedule matrix by means of the restrictions for slot allocation.

Although in [11] it is also mentioned that “for some problems, the process of repairing unfeasible individuals might be as complex as solving the original problem”, for our application, “repairing” a slot schedule matrix was not problematical, implying checking some matrix entries and, possibly, resetting them to ‘0’.

More precisely, for the crossover operation, performing the column and row exchange will generate two offspring. Each of them is then tested for validity as follows: all the elements on every column are analyzed one by one and each time a '1' is found, the two upstream consecutive entries are checked. In case any of them is also equal to ‘1’, then the entry placed in a nonselected row is changed to '0' (see also Section II.A).

Similarly, for the mutation operation, each time an individual's gene suffers mutation, the two upstream and the two downstream consecutive entries of the mutated entry have to be checked. If any of them is found '1', it is replaced by '0'.

For the beginning, in the attempt to obtain larger values for the path bandwidth, we have tested different selection methods for the GA. By means of simulation studies, we have established the values of the first three parameters for which best solutions were obtained for the entire set of experiments (number of nodes and number of time slots), crossover probability = 0.1, mutation probability =0.05 and population size = 400. The best method proved to be Tournament Selection.

Table I presents the results we have obtained for 25 time slots and variable number of hops (5, 10, 25) in 300 generations for comparison with the method presented in [5]. The algorithm was tested for path lengths ranging from 5 to 15 (increased in steps of 5), and time slots in the range 5 to 25 increased in steps of 5).

In order to fairly compare the proposed algorithm to those in [5] and [8], for each set of parameters the simulation experiment was repeated 50 times and the output of the algorithm was recorded and then averaged. The obtained results are presented in Fig. 2, Fig. 3 and Fig. 4.

TABLE I. AVERAGE PATH BANDWIDTH FOR 25 TIME SLOTS USING PROPOSED IMPLEMENTATION WITH RWS AND TS, PC = 0.1, PM = 0.05, PS = 400, NG = 300.

Hops	RWS	TS
5	6.3	6.44
10	5.18	5.46
15	4.88	5

From Fig. 2, Fig. 3 and Fig. 4 one can note that the proposed algorithm provides a larger bandwidth when compared to the method proposed in [5] and to the forward algorithm [8]. This is due to the chromosome implementation, based on GaLib *GA2DArrayAlleleGenome* class, adequate choosing of the selection algorithm (TS) and GA parameters.

As the results obtained for the average path bandwidth outperform those reported in the existing reference approaches ([5] and [8]), we have limited our research for the moment only to the solution of “repairing” unfeasible individuals. We intend to study the possibility of designing an evaluation function for unfeasible chromosomes able to penalize them adequately, but expect that running evaluation functions with penalty for unfeasible individuals will imply longer execution times for the GA.

Next step was devoted to finding the possibility of decreasing the number of generations for which the near-optimal solution is obtained.

Our simulation studies revealed that the GA behaves better for smaller crossover probabilities. This suggests that in order to obtain faster convergence, fitted individuals have to be maintained in larger proportions in the new generations. The tailoring of the elitist rate β proved to be not enough for the proposed objective. That is why we have modified the classical EMS. For our experiment, after sorting the individuals in a descending order of fitness and mating the best individual (emperor) with the second best one, the rest of the mating pairs of the emperor were selected by TS or RWS, with setting crossover probability 1. Using this approach, the algorithm could be stopped earlier, providing near optimal solutions.

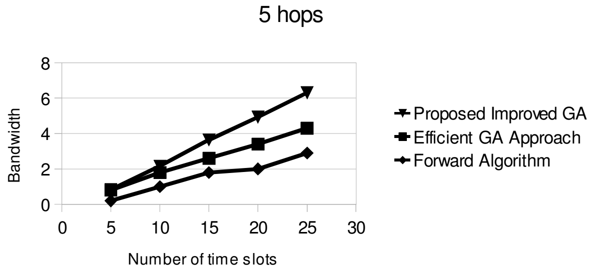

Fig. 2. Average path bandwidth for path length = 5 hops.

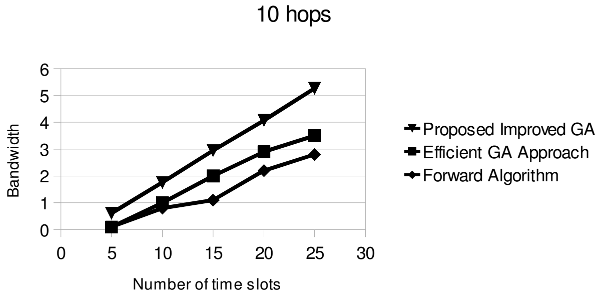

Fig. 3. Average path bandwidth for path length = 10 hops.

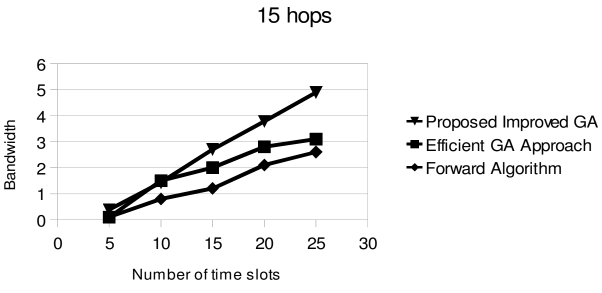

Fig. 4. Average path bandwidth for path length = 15 hops.

In order to circumvent the premature convergence of the EMS, a better approach was to combine the classical evolution of the GA for a specified number of generations (using TS or RWS) with the EMS for the remaining generations. We will denote these algorithms as EMS+TS and EMS+RWS respectively. In Table II we show the results obtained for the same experiment presented in Table I for a number of 100 generations, using for 30 generations TS or RWS (PC = 0.1, PM = 0.05) and EMS for the next 70 generations ($\beta = 0.05$), PS = 400 and NG = 100.

Due to its specific operation mode (that is, by keeping one of the parents fixed and equal to the best individual in the current population throughout each reproduction iteration), the selection algorithm for the case of EMS converges faster and so, the whole algorithm runs faster.

TABLE II. AVERAGE PATH BANDWIDTH FOR 25 TIME SLOTS USING PROPOSED IMPLEMENTATION FOR EMS+TS AND EMS +RWS, PC = 0.1, PM =0.05, PS = 400, NG =100.

Hops	EMS+TS	EMS + RWS
5	6.38	6.31
10	5.34	5.27
15	4.9	4.89

Comparing the results from Table II to those in Table I, one can see that reducing drastically the number of generations, form 300 to 100, the near optimal solution is still very close to the optimal one obtained using the proposed implementation with TS. More precisely, the average path bandwidths for the three analyzed cases differ from the obtained optimum with less than 2.2% .

The possibility of obtaining near optimal solutions in fewer generations could be exploited in real-time implementations of the proposed algorithm.

V. CONCLUSIONS

We have improved the GA framework for the problem of finding the maximum bandwidth of a path for MANETs. The algorithm was implemented using an open source GA Library, GaLib [14].

The obtained results were compared with two existing GA approaches ([5] and [8]) proving that the proposed algorithm outperforms the existing approaches. We have also proposed a new combined GA algorithm, using EMS and providing near optimal solutions in a significantly smaller number of generations as compared to those currently reported.

As the hardware performances of the processing units are continuously increasing and there have been reported even hardware implementations of GA, we consider that investigating the possibility of using heuristic algorithms in real-time applications for bandwidth maximization in TDMA-based mobile ad hoc networks is worthwhile.

REFERENCES

[1] A. Vasilakos, M. Saltouros, A. Atlasis and W. Pedrycz, "Optimizing QoS routing in hierarchical ATM networks using computational intelligence techniques", *IEEE Transactions on Systems, Man and Cybernetics*, Part C, vol. 33, no. 3, pp. 297–312, 2003. [Online]. Available: http://dx.doi.org/10.1109/TSMCC.2003.817354

[2] C.-J. Huang, W. K. Lai, Y.-T. Chuang and S.-Y. Hsiao, "A dynamic alternate path QoS enabled routing scheme in mobile ad hoc networks", *International Journal of Wireless Information Networks*, vol. 14, no. 1, pp. 1–16, 2011. [Online]. Available: http://dx.doi.org/ 10.1007/s10776-006-0048-8

[3] T. Bheemarjuna Reddy, I. Karthigeyan, B.S. Manoj, and C. Siva Ram Murthy, "Quality of service provisioning in ad hoc wireless networks: a survey of issues and solutions", *Ad Hoc Networks*, vol. 4, pp. 83–124, 2006. [Online]. Available: http://dx.doi.org/10.1016 /j.adhoc.2004.04.008

[4] A. Ephremides, T. V. Truong, "Scheduling broadcasts in multihop radio networks", *IEEE Transactions on Communications,* vol. 38, no. 4, pp. 456–460, 1990. [Online]. Available: http://dx.doi.org/10.1109/ 26.52656

[5] P. R. Sheu, K. L. Chang, C. C. Huang, "An efficient genetic algorithm for the bandwidth calculation problem in TDMA-based mobile ad hoc networks", *Journal of Information Technology and Applications*, vol. 2, no. 3, pp. 125–134, 2007.

[6] Y. S., Chen, Y. C. Tseng, J. P. Sheu and P. H. Kuo, "An On-Demand, Link-State, Multi-Path QoS Routing in a Wireless Mobile Ad Hoc Network", *Computer Communications*, vol. 27, no. 1, pp. 27–40, 2002.

[7] H. C. Lin, P. C.Fung, "Finding available bandwidth in multihop mobile wireless networks", in *Proc. of the IEEE VTS 51st Vehicular Technology Conference (VTC 2000 Spring),* vol. 2, 2000, P. 912–916.

[8] P. R. Sheu, K. L Chang, "A genetic algorithm for the bandwidth calculation problem in mobile ad hoc networks", in *Proc. of the 2002 International Computer Symposium* (*ICS 2002*), Taiwan, R.O.C., 2002, pp. I-331–I-338.

[9] G. Chakraborty, "Genetic algorithm to solve optimum TDMA transmission schedule in broadcast packet radio networks", *IEEE Transactions on Communications*, vol. 52, no. 5, pp.765–777, 2004. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2004.826234

[10] D. E. Goldberg*, Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, NY, 1989.

[11] Z. Michalewicz, "Heuristic methods for evolutionary computation techniques", *Journal of Heuristics*, vol. 1, pp. 177–206, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00127077

[12] C. Y. Ngo, V. O. K. Li, "Centralized broadcast scheduling in packet radio networks via genetic-fix algorithms", *IEEE Transactions on Communications*, vol. 51, no. 9, pp.1439–1441, 2003. [Online]. Available: http://dx.doi.org/10.1109/TCOMM.2003.816950

[13] M. Li, Y. Lu, "A refined genetic algorithm for accurate and reliable DOA estimation with a sensor array", *Wireless Personal Communications*, vol. 43, no. 2, pp. 533–547, 2007. [Online]. Available: http://dx.doi.org/10.1007/s11277-007-9248-5

[14] M. Wall, "GAlib: A C++ Library of Genetic Algorithm Components", 1996. [Online]. Available: http://lancet.mit.edu/ga

[15] A. Hamann, R. Ernst, "TDMA time slot and turn optimization with evolutionary search techniques", in *Proc. of Design, Automation and Test in Europe,* vol. 1, 2005, pp. 312–317. [Online]. Available: http://dx.doi.org/10.1109/DATE.2005.299