A GA-Based Effective Fault-Tolerant Model for Channel Allocation in Mobile Computing

Lutfi Mohammed Omer Khanbary and Deo Prakash Vidyarthi

***Abstract*—Efficient channel allocation to mobile hosts is of utmost importance in a cellular network. A genetic algorithm (GA), which is a useful tool in solving optimization problems, is explored to design a fault-tolerant cellular channel allocation model that allows a cell to continue communicating with its mobile hosts, even if there are insufficient channels available in the cell. Sometimes, the load over a cell may increase to the extent that it needs more channels than it actually has in order to handle the traffic. On the other hand, it is quite possible that the load in some other cell is less than its channel capacity, resulting in underutilization of the channels. This problem is solved by temporarily taking unutilized channels from cells that have lesser load and allocating them to the cells that are overloaded. We propose a model that reuses available channels more efficiently. The model also considers the handoff by using the reserved channel technique. A reserved pool of channels makes the model fault tolerant. Thus, the proposed work uses GA for fault-tolerant dynamic channel allocation to minimize the average number of blocked hosts and handoff failures in the mobile computing network. Simulation experiments evaluate the performance of the proposed model. Comparison of the results with the two recent earlier models reveals that the proposed model works better in serving mobile hosts.**

***Index Terms*—Channel reuse, fault-tolerant model, genetic algorithm, handoff, mobile communication.**

I. Introduction

THE AREA covered by a cellular network is divided into smaller regions called cells, which are usually hexagonal. An ideal model of the cellular radio system consists of an array of hexagonal cells with a base station (BS) located at the center of each cell.

The available spectrum in a cell is used for uplink channels for mobile terminals (MTs), which are also called mobile stations (MSs), to communicate with the BS and for downlink channels for the BS to communicate with MTs. All users in the cell are served by the BS [1], [2]. Fig. 1 depicts a brief architecture of a mobile cellular system. Traffic from MSs is routed through a mobile switching center (MSC) to other networks, such as a public switched telephone network (PSTN).

Wireless channels are scant resources and need to be reused to its maximum possible extent. Hence, there is a need to properly manage these resources. This paper proposes an approach to managing these resources in an effective manner.

Manuscript received November 10, 2006; revised April 10, 2007, July 3, 2007, and August 7, 2007. The review of this paper was coordinated by Prof. C. Lin.

L. M. O. Khanbary is with the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India, on leave from the Department of Computer Science and Engineering, Aden University, Aden, Yemen (e-mail: llkhanbari@yahoo.com).

D. P. Vidyarthi is with the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi 110067, India (e-mail: dpv@mail.jnu.ac.in; dpvidyarthi2002@yahoo.com).

Digital Object Identifier 10.1109/TVT.2007.907311

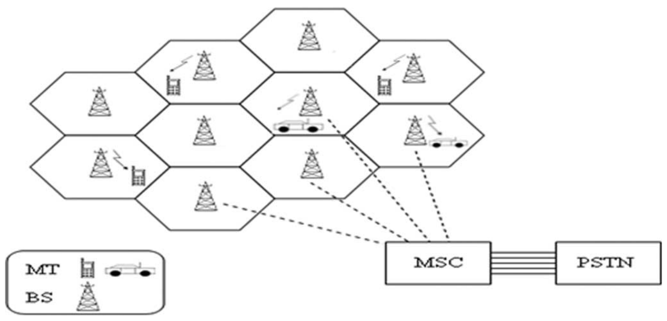

Fig. 1. Cellular system.

Mobile computing deals with the execution of an application from a mobile host over other servers or hosts. The emphasis in mobile computing is on the computation that utilizes the communication infrastructure; thus, handoff is to be taken care of more seriously than call initiation. The proposed model emphasizes handoff management using the reserved channel technique and, at the same time, manages the new calls well.

A genetic algorithm (GA), which is often used to solve optimization problems, is based on Darwin's theory of "survival of the fittest." Individuals from the population of potential solutions reproduce, and solutions are refined successively over a number of generations. GA is intended to provide the better solution, as it is based on the natural theory of evolution. The taxonomy in the GA is borrowed from genetic engineering. GA has recently gained importance across many disciplines as a problem-solving tool [3]–[5]. The potential of a GA is exploited to solve the cellular resource allocation problem presented here.

Fault tolerance is the ability of a system to respond gracefully to an unexpected hardware or software failure. For our model of channel allocation in mobile computing, fault tolerance is the ability of a cell to continue communicating with its mobile hosts, even if there is an insufficient number of channels available. A fault-tolerant model for channel allocation is also proposed in [7]. This paper presents a GA-based fault-tolerant model, which is better in managing resources.

The rest of the paper is organized as follows. In Section II, channel allocation problem in mobile computing is elaborated upon. In Section III, a brief discussion on genetic algorithms and a few GA-based channel allocation models are presented. The proposed GA-based fault-tolerant channel allocation (FTCA) model is described in Section IV. In Section V, the performance of the FTCA model is evaluated and compared with that of the IGA model proposed in [6] and the fault-tolerant model proposed in [7]. Observations, which were based on the results of the experiment on the model, and the comparative study appear in Section VI, followed by concluding remarks in Section VII.

II. Channel Allocation in Mobile Computing

The channel allocation problem deals with the allocation of frequency channels to mobile hosts. Two important concepts in channel allocation are cellular reuse of channels and handoff.

The fundamental and elegant concept of cells relies on the channel or frequency reuse, that is, the usage of the same frequency by different mobile users separated by a minimum distance [7], without interfering with each other (cochannel interference).

Another important concept in cellular networks is handoff, which is also called handover. Handoff occurs when a user moves from the coverage area of one BS to the adjacent one while still involved in communication. A new channel is to be assigned to continue the call. The new channel may be within the same cell (intracell handoff) or in a different cell (intercell handoff). These are important issues in microcellular systems where the cell radius is small [1], [8].

Handoff techniques are significant in any cellular system, as it is irritating for a mobile user to break the connection while moving from one cell to another. It is preferable to block a new initiated call rather than to break an existing one. Handling handoff is of the utmost importance in mobile computing, as breaking a connection may sometimes result in restarting the application from scratch. Several techniques are used to manage handoffs that cope well with the traffic variation while maintaining a high level of utilization.

A. Channel Allocation Problem

With the significant increase in the number of the mobile users, the number of mobile devices (hosts) has increased. Due to this increasing load, the number of mobile hosts that could not connect to the destination (called blocked hosts) has also increased. There are two ways to solve this problem. The first is by increasing the number of channels, but that incurs certain costs. The other way is to utilize the current infrastructure efficiently and try to maximize the uses of the available infrastructure so that the best performance can be achieved. The latter is obviously preferable, as it is always cost effective to utilize the available resources effectively than to add more bandwidth. The channel allocation problem addresses the effective channel utilization in a cellular network.

B. Channel Allocation Scheme

The channel allocation schemes are described as follows. In fixed channel allocation (FCA), assignment of frequencies to a cell is fixed, and there is no channel reuse. This is inefficient if the traffic load varies time to time and dynamic channel allocation (DCA) is used in its place. As the name specifies, DCA allocates the channel dynamically. A better method, in the case of a heavy load in one cell and light load in the neighboring cell, is to borrow channels from the neighbor cells. Cells with more traffic are dynamically assigned more channels. This scheme is known as the borrowing channel allocation (BCA) and is common in global systems for mobile communication [6]. However, BCA requires careful traffic analysis. There are several other ways to deal with the excess load in mobile networks in addition to channel borrowing, such as channel sharing and cell splitting [13].

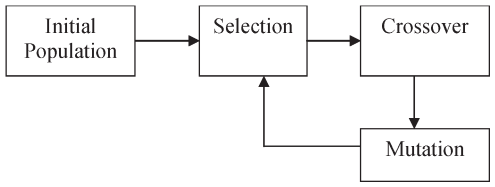

Fig. 2. Operations in GA.

There are two important approaches in channel allocation strategies. These are as follows.

1) Centralized Approach: In this approach [7], [9], [10], a request for channel is sent to and processed by a central controller called a mobile switching center (MSC). MSC is the only one that has access to systemwide channel usage information. It is responsible for allocating channels to cells in such a way that no cochannel interference arises and that channels are used in an efficient manner. The functioning of the whole system depends solely on the MSC. Because it is a centralized approach, MSC suffers from single-point failure. In addition, this approach is not scalable, as the MSC may become a bottleneck when the traffic load of the system increases.

2) Distributed Approach: In the distributed approach [11], [12], there is no central controller such as MSC; instead, a BS exists in each cell. BSs share the responsibility for allocating channels. Each BS makes this decision independently, based on its local information. BSs exchange information whenever necessary. The BS in the cell that wants to borrow a channel and the BS in the cell that grants the channel work in cooperation to ensure that no cochannel interference arises. The distributed approach requires much less signaling because the BS in each cell keeps information about the status of currently available channels in its neighborhood, and every change is communicated between the involved BSs.

III. GA and Its Related Models

The GA is a search procedure based on the principle of evolution and natural genetics. GA combines the exploitation of past results with the exploration of the new areas of the search space by using the survival-of-the-fittest technique combined with a structured, yet randomized, information exchange. In each new generation, a set of strings is created using information from previous ones. Occasionally, a new part is tried from good measure. GAs are randomized, but they are not simple random walks. GAs efficiently exploit historic information to speculate on new search points with expected improvement [3], [4], [14].

In GA, we start with an initial population derived from the solution space and then apply the genetic operators on it to appropriately mix exploitation and exploration. A simple genetic algorithm consists of an initial population followed by selection, crossover, and mutation operations [4], as shown in Fig. 2.

Selection: Selection operation selects good results among chromosomes through some objective function (fitness function) used to rank the quality of the chromosomes. A fitness value is assigned to the chromosome using the fitness function, and the chromosome is evaluated with this value for its survival. Thus, the fitness of the chromosome depends on how well that chromosome solves the problem at hand. Strings with a higher value have a higher probability of contributing to one or more offspring in the next generation.

Crossover: The idea of crossover is to swap part of the information between a pair of chromosomes to obtain the new chromosome. A simple crossover may proceed in two steps. First, members of the newly reproduced strings in the mating pool are mated at random. Second, each pair of strings undergoes crossing over as follows: An integer position k along the string is selected uniformly at random between 1 and the string length minus one $[1, l-1]$. Two new strings are created by swapping all characters between positions $k+1$ and l inclusively [3].

Mutation: In mutation, a chromosome is altered a little bit randomly to get a new chromosome. The mutation operator is used to introduce new genetic material (e.g., 0 or 1). As a result of its generality, mutation is an insurance policy against the premature loss of important notions. The probability of applying mutation is often very low. Mutation rates are normally small in natural populations.

A. Related Models

A DCA model using GA for a broadband fixed wireless access network was proposed by Wong and Wassell [15]. In this model, the aim is to allocate the channel to reduce the signal-to-noise ratio (SNR) and meet traffic demands at the same time. Their GA-based model is compared to least interference and channel segregation models and shows that the GA-based method achieves an SNR gain, as compared to the other two methods [15].

A hybrid genetic algorithm (HGA) for channel reuse in multiple access telecommunication networks was proposed by Kassotakis *et al.* [16]. They combined GA with a local search algorithm to ensure a reliability property of the GA with the accuracy of hill-climbing methods. The performance of the proposed HGA is compared via simulation to that of the graph coloring algorithm (GCA) proposed in [17], and it is concluded that the GCA performance is comparable to that of HGA at a light/medium load, whereas HGA solutions massively outperform the GCA at heavy network loads.

An evolutionary genetic DCA for resource management in mobile systems is proposed by Asvial *et al.* [18]. The chromosome structure proposed by them is the combination of traffic load and interference-limited DCS. The constraints in interference used in the algorithm include cosite interference, cospotbeam interference, and adjacent cospotbeam interference. Total interference is proposed to be minimized.

Zomaya and Wright [14] have proposed a GA-based DCA model in which they have modified the genetic operator mutation. They have compared their model with that of the FCA and greedy borrowing heuristics for the average number of blocked channels metric and are able to show that their model works better than FCA and has a slight edge over the heuristic model.

A heuristic fault-tolerant distributed channel allocation scheme is proposed by Yang *et al.* [7]. They assumed a three-cell cluster model, which belongs to the nonresource planning model, where a cell may borrow a channel based on partial channel usage information it receives from some of its neighbors. A cell can lend a channel to multiple borrowers (three at most), as long as any two are not neighbors. They proved that their scheme outperforms the algorithm proposed by Prakash *et al.* [11].

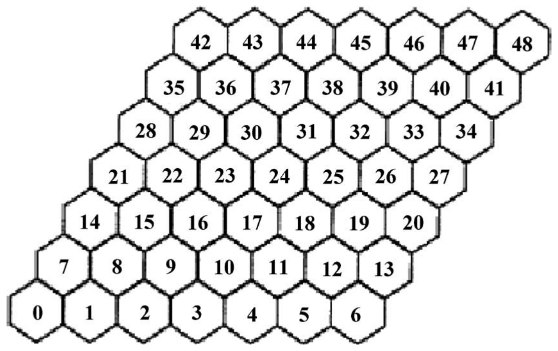

Fig. 3. Cellular network.

An improved GA model is proposed by Mahapatra *et al.* [6]. In their work, GA is applied for channel allocation in DCA with channel borrowing. A new genetic operator called "pluck" is introduced for improving the simple GA, and the proposed model is called the improved genetic algorithm (IGA).The idea in pluck operation is to add some knowledge by selecting the chromosomes that may not be giving good results right now but may lead the system to a stabilized state for a better result in the future. The model is compared to Zomaya and Wright's [14] model and shows that the IGA has an edge for improving the channel utilization.

IV. Proposed Model

We propose a GA-based fault-tolerant model for channel allocation in this work. The proposed algorithm is under the resource planning model [7], i.e., primary channels are initially allocated to each cell. Furthermore, secondary (borrowed) channels must be returned to the cell from which they have been borrowed as soon as the call is over.

Other assumptions made in the model are listed as follows.

A. Assumptions

- Cells are assumed to be hexagonal.
- The number of channels per cell is distributed according to the initial demand in each cell (based on the past experience of the channel usage pattern in cells).
- For experimental purposes, mobile hosts are distributed randomly among cells in proportion to the number of channels per cell. It is assumed that mobile host movement across the cells is stochastic.
- The cells have been numbered in the increasing order of enumeration, as shown in Fig. 3.
- Each cell has a set of reserved channels (in proportion to primary channels) that will immediately be given to a crossing over mobile host (to handle handoff). At the same time, however, the cell will search for a new channel.

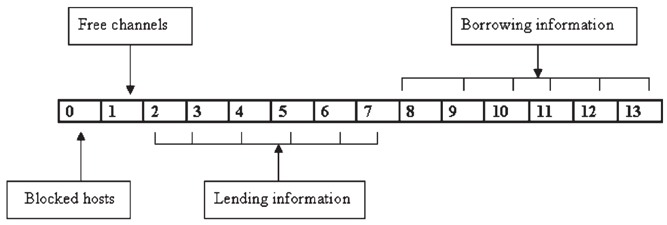

Fig. 4. Chromosome structure.

As soon as it gets that channel, it will be allocated to the crossed-over mobile host so that the reserved channel pool is intact.

- The probability of applying mutation is often very low. The main weakness of mutation in a channel allocation problem is taking/borrowing decisions ahead of time that may result in nonoptimality for two reasons. First, their effectiveness is not measured in the fitness function, and second, these decisions degrade future quality of service [14]. Thus, the probability of mutation is assumed to be zero in the proposed model.

The performance of the algorithm is evaluated by measuring the average number of blocked hosts and handoff failures in each generation.

B. Aim of the Algorithm

The algorithm exploits the potential of the genetic algorithm to design a fault-tolerant cellular resource allocation model.

Sometimes, the load over a single cell is increased, and thus, it needs more channels than it actually has in order to handle communication traffic and to minimize the number of blocked hosts. On the other hand, it is quite possible that the load on a cell is less than its channel capacity, and therefore, channels are being underutilized. This problem is addressed by borrowing the extra channels from the cells that have a lesser load and temporarily allocating them to the cells that are overloaded. The algorithm also considers the handoff using the reserved channel technique. Thus, we implement DCA, using a genetic algorithm to minimize the number of blocked hosts and the handoff failures (as a part of the blocked hosts) in the mobile computing network system.

C. Encoding Used

- Each cell is represented by a chromosome.
- A chromosome is an array of length 14, as shown in Fig. 4.
- The first location of the chromosome array represents the number of blocked hosts.
- The second location of the chromosome array is for the number of free channels.
- The next six locations contain the information about the channel lending to six neighbor cells.
- The last six locations contain the information about the channel borrowing from six neighbor cells.
- The chromosome of a cell and the chromosomes of its six neighboring cells form a matrix of 7 *14 called a superchromosome.

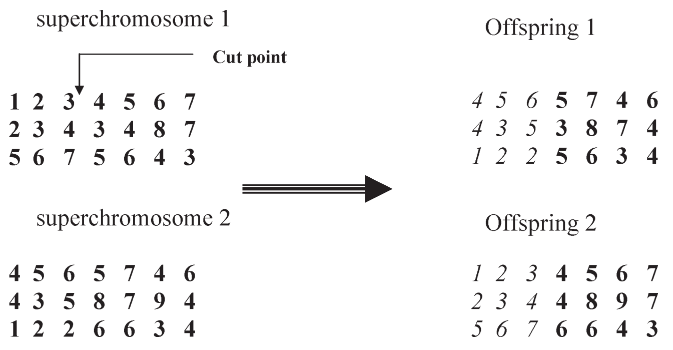

Fig. 5. Crossover operation.

- Chromosomes are combined into a superchromosome, and all the superchromosomes together give the information about the whole network.
- All GA operations are performed on the superchromosome.

D. Functions Used in the Algorithm

The functions used in the algorithm have been described in the following section.

1) Lend_borrow: This function performs the following tasks.

- It permits a cell to lend the same channel to any of its neighbors for its concurrent use, provided the two borrowing cells C_i and C_j satisfy
  - $C_i - C_j \neq 1$;
  - $C_i - C_j \neq$ number of cells in a row;
  - $C_i - C_j \neq$ number of neighbor cells in cell pattern.

  This is to avoid cochannel interference. Here, C_i and C_j represent the cells as shown in Fig. 3.
- It handles the handoff problem using the channel from the reserved pool of channels.
- It searches for a new channel for the crossing-over mobile host.

2) Crossover: The crossover operation occurs between two superchromosomes (two matrices) and generates two offspring from them, i.e., two new matrices [6]. After this operation, we get two new different chromosomes. In Fig. 5, two example superchromosomes are taken, and the crossover operation is illustrated. The crossover site is the cut point in the figure. Note that the example superchromosomes are the shorter ones and are not the one used in the model.

3) Update: This function is used to estimate the cell information and update it after the crossover operation. If the number of hosts is greater than the available free channels, we then have the following.

- The free channel number is updated to zero.
- The lending part of the chromosome is updated to zero.
- The borrowing part of the related neighbors is updated.

If the number of hosts is less than the available free channels, we then have the following.

- The blocked hosts number is updated to zero.
- The borrowing part of the chromosome is updated to zero.
- The lending part of the related neighbors is updated.

The update operation is necessary to carry forward the correct information.

4) Fitness Function: The fitness function is used to measure the fitness value of each chromosome. The chromosome with the good fitness value will be selected with the objective of minimizing the number of blocked hosts and handoff failures.

The fitness function used in this model is as follows:

$$\text{Fitness} = \text{blocked\_hosts} - \text{reserved\_channels} - \text{prime\_channels}. \quad (1)$$

The fittest chromosome is the one with the lowest fitness value.

The fitness function in (1) is simple and easy to evaluate. We found this to be effective, although any other fitness function may be designed and chosen for the experiment.

E. Algorithm

The algorithm used in the model is as follows.

```
1) Input the total number of channels and the mobile hosts.
2) Assign channels to each cell based on the initial demand.
   // based on a priory knowledge.
3) Input generation_no. // for how many generations to carry
   out experiment.
4) Initialize generation_index = 0. // used as index.
5) Initialize total_blocked_hosts = 0.
6) Distribute the hosts among the cells in proportion to each
   cell capacity. // host assignment depending on channel
   assignment as in Step 2.
7) Create an initial population.
8) Calculate the number of free_channels and blocked_hosts
   of each cell.
9) Repeat steps 10–18 until the generation_index =
   generation_no.
10) Perform Lend_Borrow ( ).
11) Perform Crossover ( ).
12) Perform Update ( ).
13) Calculate Fitness ( ). // based on (1).
14) Select the best chromosome as the current chromosome.
15) Calculate the free_channels and blocked_hosts of each
    cell again.
16) Output the number of blocked_hosts and handoff failures
    resulted in the current generation.
17) Increment generation_index.
18) Total_blocked_hosts = Total_blocked_hosts +
    blocked_hosts.
19) Average_blocked_hosts =
    Total_blocked_hosts/generation_no.
20) Output Average_blocked_hosts.
```

The algorithm is designed to be executed for the number of generations, which is the termination criterion used for the simulation experiment. In actual implementation, the termination criterion may be the convergence of solution.

Steps 1–8 are taking constant time. The complexity of the algorithm depends on the number of iterations and the operations performed within the iteration. The Lend_Borrow function takes constant time as it depends on the number of neighbor cells, which is fixed. The time for a crossover depends on the size of the chromosome and the population size. If the size of the population is n it will be of the order $\Theta(n)$. The update and fitness calculation is about some elementary addition and subtraction; thus, the time is constant. Thus, complexity of the algorithm will be of the order $\Theta(nN)$, where N is the number of iterations. It is observed in Section V that, in most cases, the solution converges by the 20th generation.

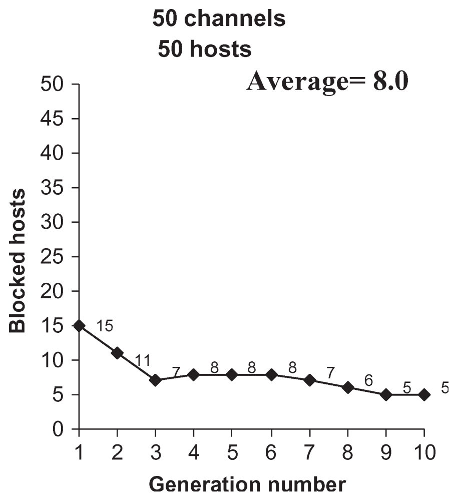

Fig. 6. Fifty channels and 50 hosts.

V. Simulation Experiments

In this section, the performance of the proposed algorithm is evaluated. We have performed the experiment up to 10 and 20 generations to see the effect of solution convergence. The results of the algorithm have also been compared to some results of the channel allocation algorithm proposed in [6] and [7]. The experiments have been designed by writing programs in C++.

Simulation Parameters: The simulation parameters used in the experiment are as follows.

1) The simulated cellular network consists of 20 cells.
2) The crossover probability is 1.
3) The mutation probability is 0.
4) The total number of channels and hosts in the network are varying.
5) The reserved channels for all the experiments are 30% of the total number of channels and are distributed among the cells in proportion to the distribution of the mobile hosts. For example, in the experiments with 50, 100, 150, and 200 channels, the reserved channels are 15, 30, 45, and 60, respectively.
6) The handoff probability is considered to be 30%, which is in conformity with that of the reserved channels.

The results are represented in the performance graphs, where the x-axis is the number of generations for all the experiments. The y-axes are blocked hosts for some experiments and handoff failures for some other experiments.

The experiment is done first for ten generations and then extending it to 20 generations to have clear observation of the convergence of the result.

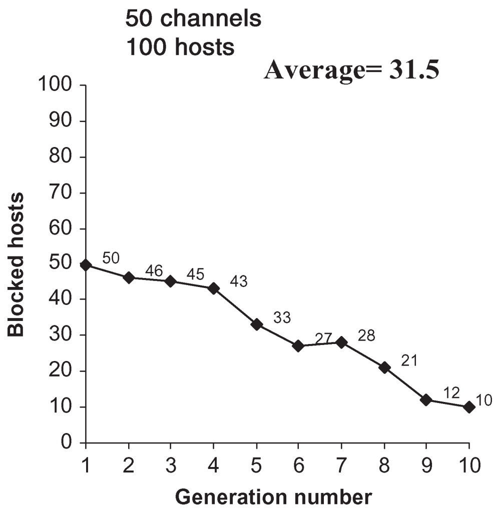

Fig. 7. Fifty channels and 100 hosts.

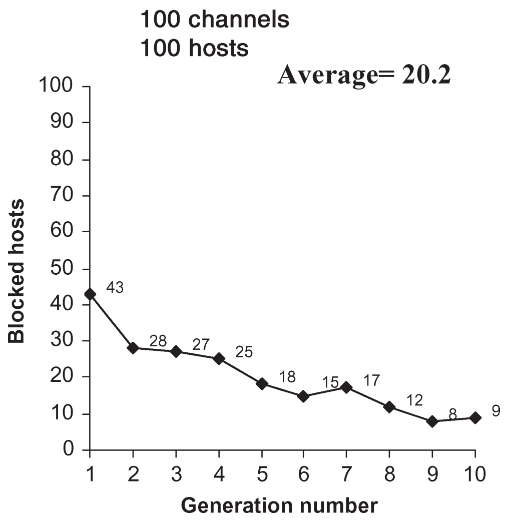

Fig. 9. One hundred channels and 100 hosts.

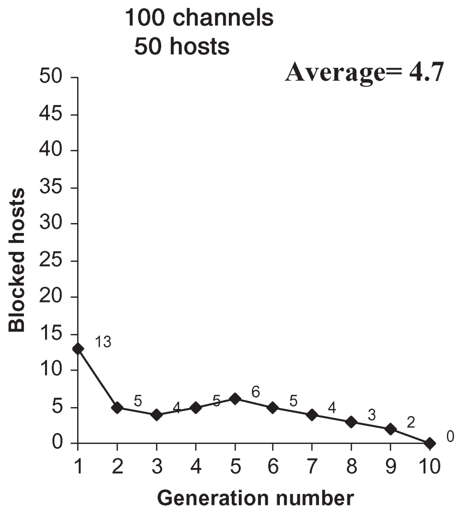

Fig. 8. One hundred channels and 50 hosts.

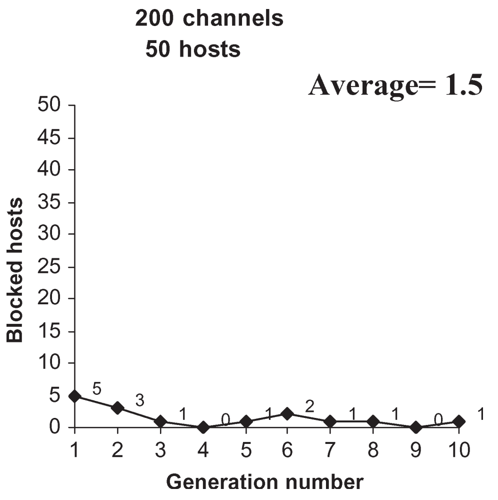

Fig. 10. Two hundred channels and 50 hosts.

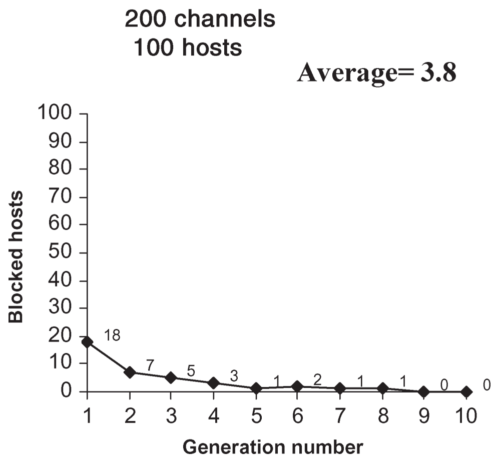

Fig. 11. Two hundred channels and 100 hosts.

A. Blocked Hosts Experiment

We performed the experiment to calculate the average number of blocked hosts for generations 10 and 20.

The input values for the channels and hosts are as follows:

- number of channels: 50, 100, 150, 200;
- number of hosts: 50, 100, 150, 200.

The graphs for the experiment are shown in Figs. 6–17.

1) Simulation Results for 10 Generations: The experiment is conducted on the given input data, and the result is observed for 10 generations. The graphs shown in Figs. 6–11 reveal the performance.

Conspicuous observation from the graph is that if there are more channels and fewer number of hosts, the rate of blocked host drops sharply.

2) Simulation Results for 20 Generations: We conducted the same experiment with the same set of data as in Section V-A1, but we observed the results up to 20 generations to see the convergence of the result. The experiment shows that the result is stabilized by 20 generations. The graphs are shown in Figs. 12–17.

Results obtained in Sections V-A1 and 2, for average blocked hosts, are summarized in the graphs shown in Figs. 18 and 19, respectively. The average value is chosen to indicate the aggregate performance.

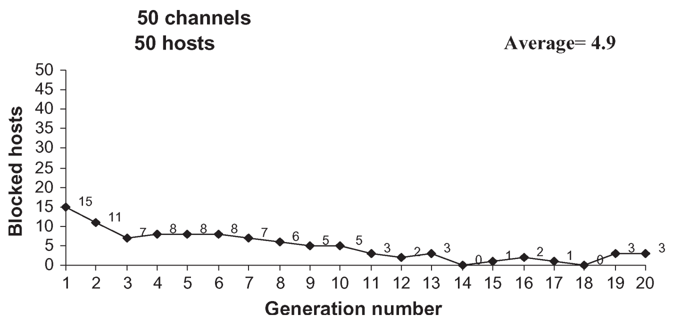

Fig. 12. Fifty channels and 50 hosts.

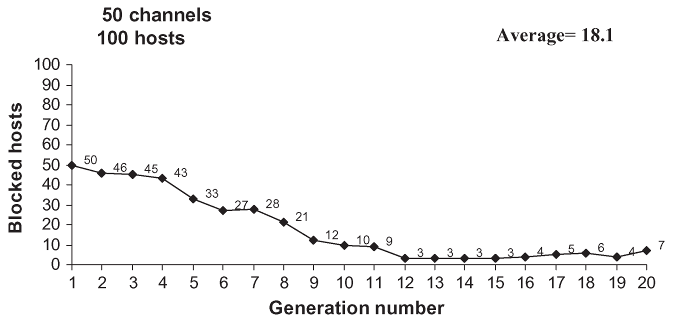

Fig. 13. Fifty channels and 100 hosts.

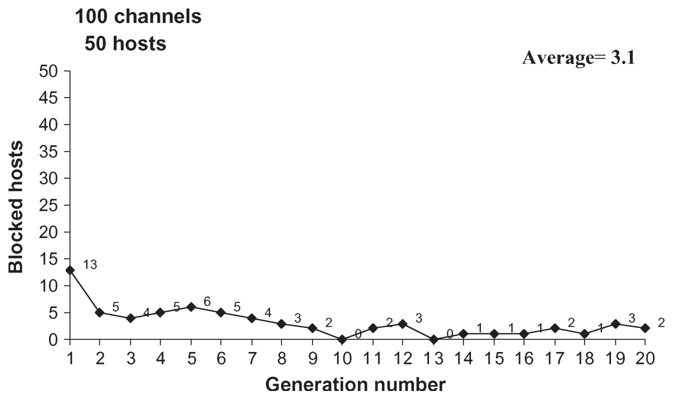

Fig. 14. One hundred channels and 50 hosts.

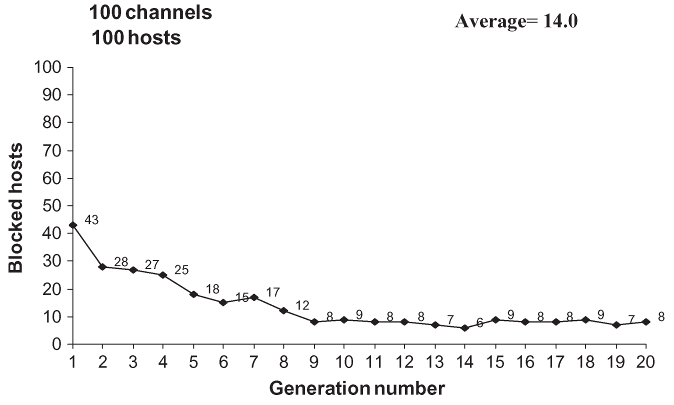

Fig. 15. One hundred channels and 100 hosts.

B. Handoff Failure Experiment

The proposed model works well in managing handoff. To observe the effect of handoff exclusively, the experiment is conducted for handoff.

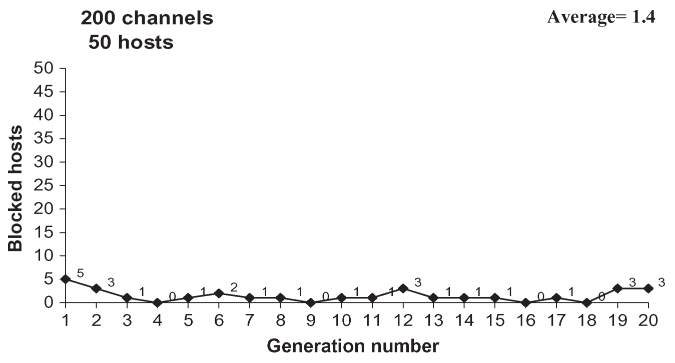

Fig. 16. Two hundred channels and 50 hosts.

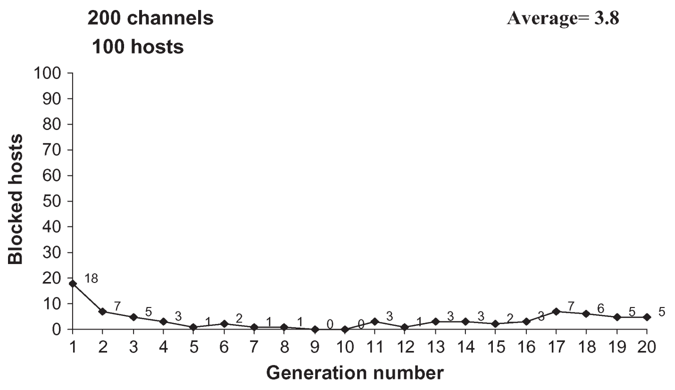

Fig. 17. Two hundred channels and 100 hosts.

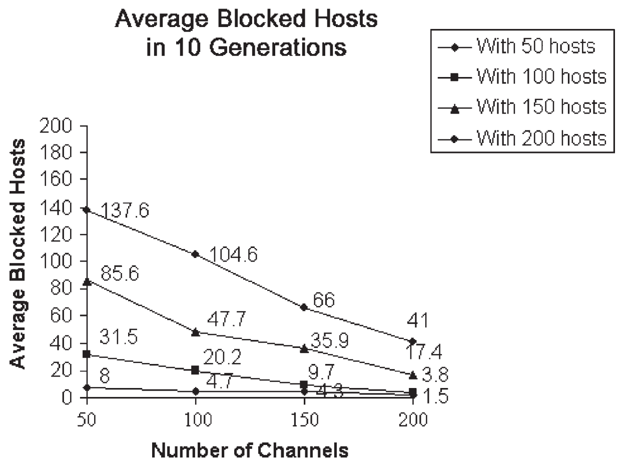

Fig. 18. Results for 10 generations.

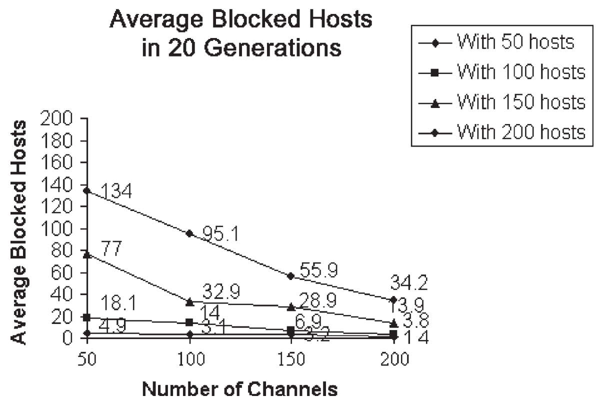

Fig. 19. Results for 20 generations.

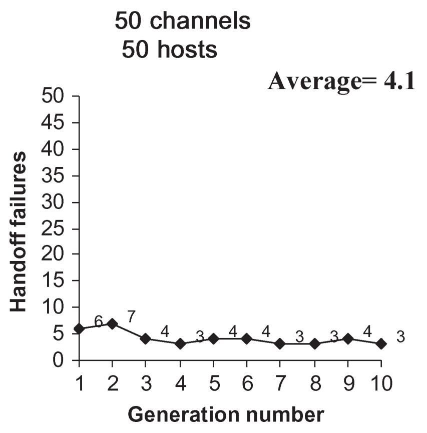

Fig. 20. Fifty channels and hosts.

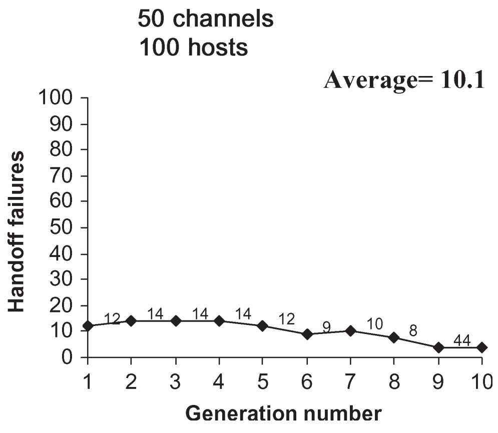

Fig. 21. Fifty channels and 100 hosts.

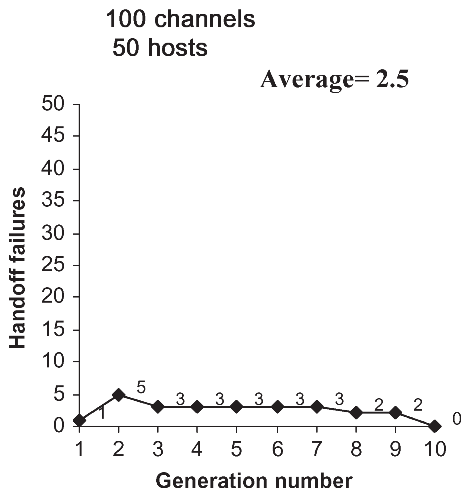

Fig. 22. One hundred channels and 50 hosts.

In this section, we performed the experiment to calculate the average number of handoff failures up to 10 and 20 generations. The other parameters are same as in the earlier experiment.

1) Simulation Results for 10 Generations: The experiment is conducted on the given input data, and the result is observed for 10 generations. The graphs shown in Figs. 20–23 indicate the performance.

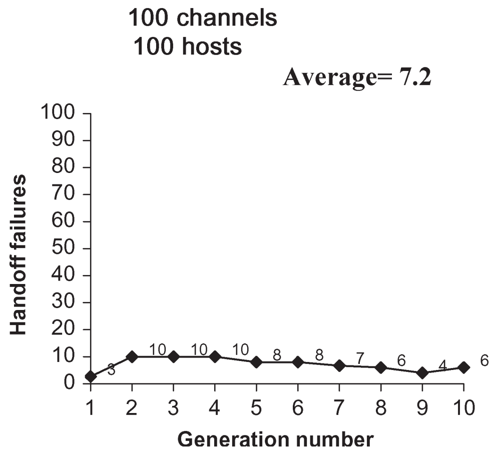

Fig. 23. One hundred channels and 100 hosts.

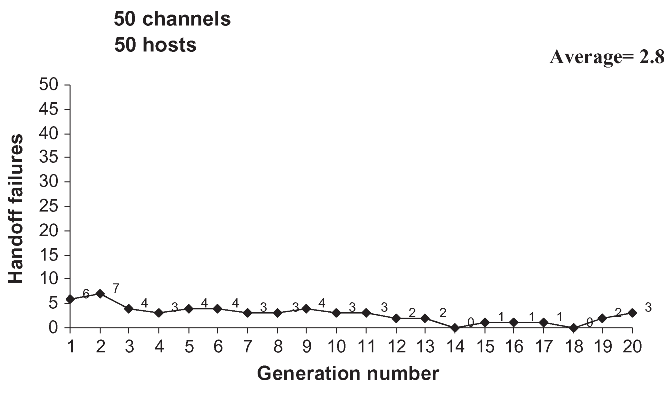

Fig. 24. Fifty channels and 50 hosts.

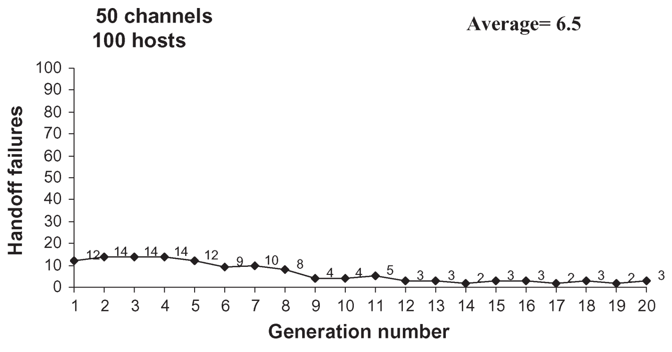

Fig. 25. Fifty channels and 100 hosts.

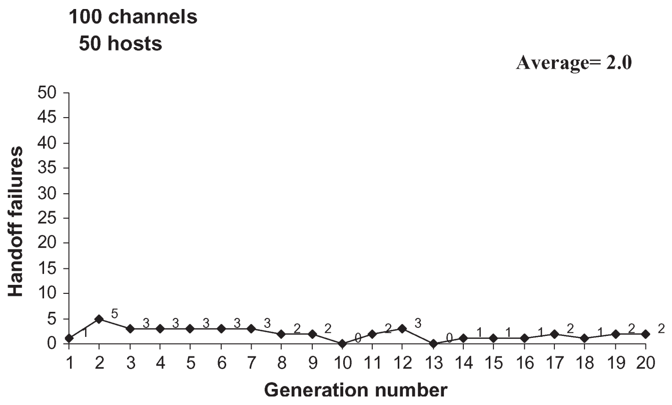

Fig. 26. One hundred channels and 50 hosts.

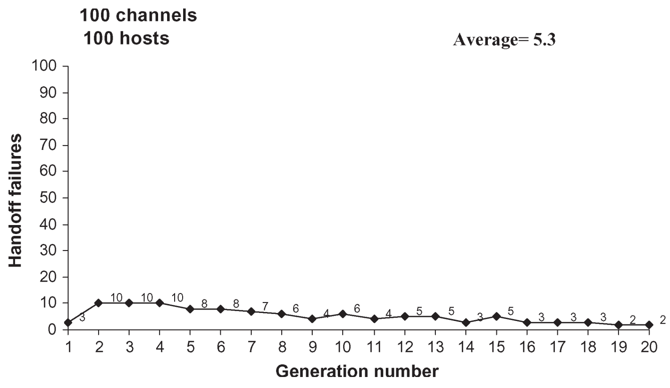

Fig. 27. One hundred channels and 100 hosts.

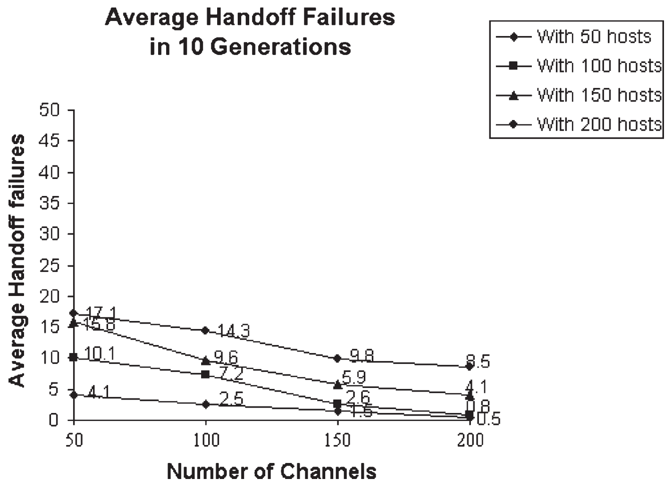

Fig. 28. Results for 10 generations.

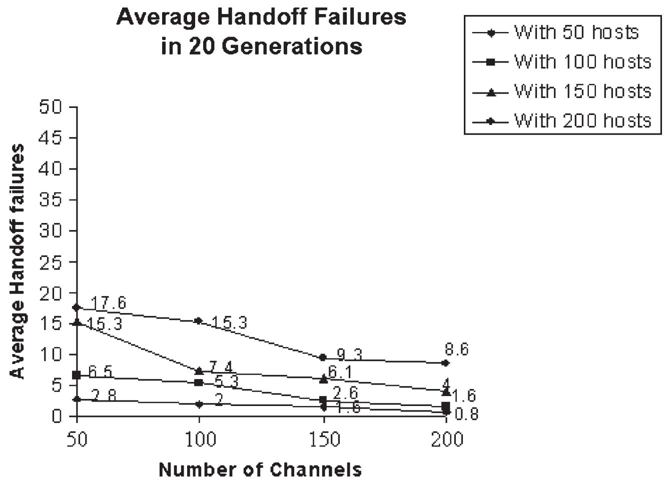

Fig. 29. Results for 20 generations.

2) Simulation Results for 20 Generations: Again, we conducted the same experiment, with the same set of data as in Section V-B1, and observed the result up to 20 generations to see the convergence of the result. The graphs are shown in Figs. 24–27.

Results for average handoff failure, which are discussed in Sections V-B1 and 2, are summarized in Figs. 28 and 29, respectively.

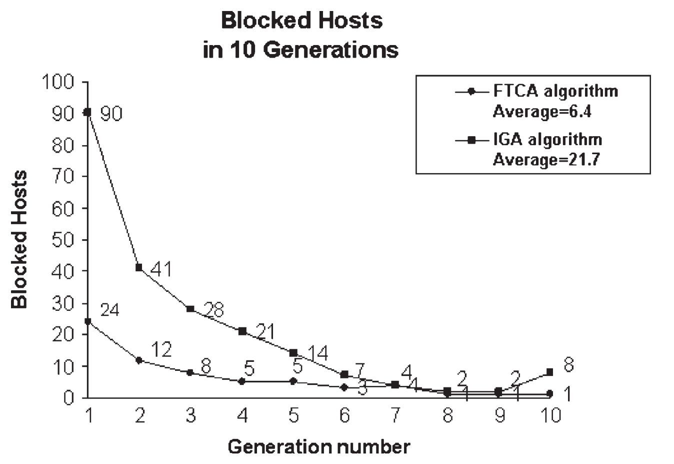

Fig. 30. FTCA and IGA with 100 hosts in 10 generations.

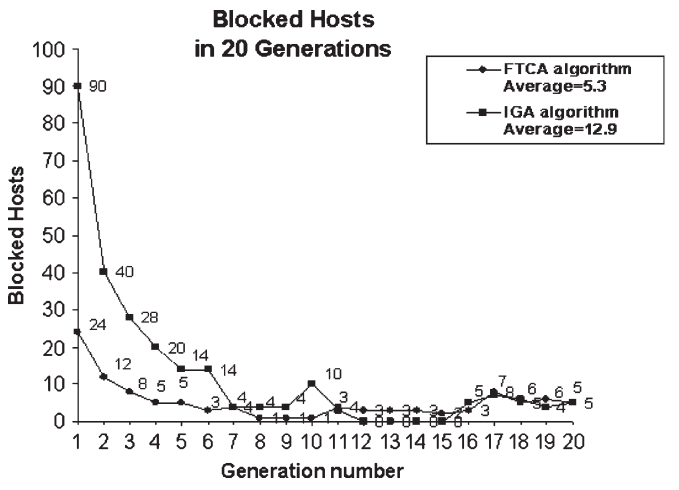

Fig. 31. FTCA and IGA with 100 hosts in 20 generations.

C. Comparative Study

To show the effectiveness of our model in allocating channels, we have compared it to the two recent channel allocation models: one GA-based model that is not fault tolerant and the other a heuristic fault-tolerant model.

1) Comparison With the IGA Model: We compared our algorithm to the results of the blocked hosts obtained by the IGA algorithm in [6], where the genetic algorithm is improved by introducing the pluck operation to reduce the blocked hosts.

Because there are 190 channels in the algorithm [6], we performed the experiment with the same number of channels. There were 50 and 100 hosts. Graphs for 10 and 20 generations are shown in Figs. 30–32.

2) Comparison With the Fault-Tolerant Model: Because the proposed model is fault tolerant, we compared it to another fault-tolerant model of Yang *et al.* [7]. The comparison is made for the call failure rate and the handoff drop rate. The comparison was made with the cell failure only, as it is more realistic.

To compare the models exactly, most of the parameters and environment are kept same to that of [7]. There were a total number of 36 cells used and a total number of 300 channels. The experimental data were collected after 10 000 calls were processed; the simulation ends after 100 000 calls are

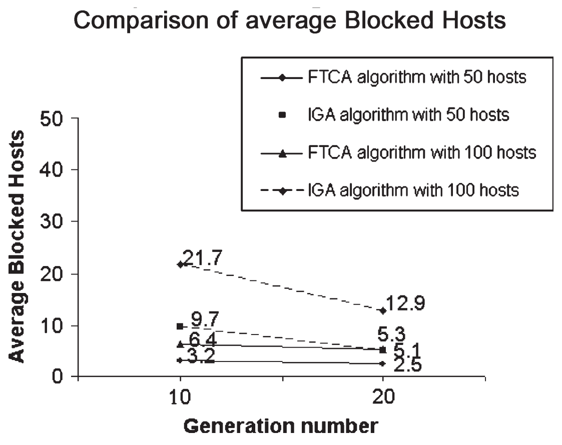

Fig. 32. FTCA versus IGA for varying hosts and generations.

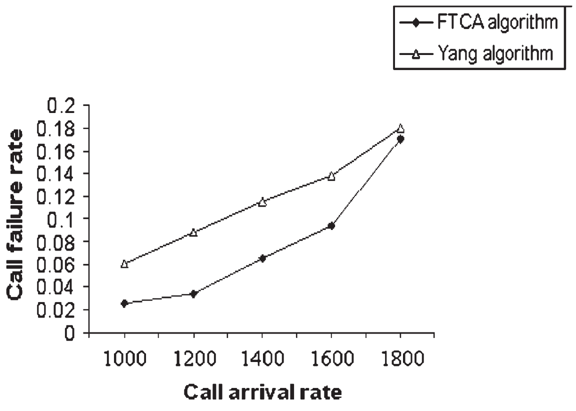

Fig. 33. Call failure rates for FTCA and Yang algorithm.

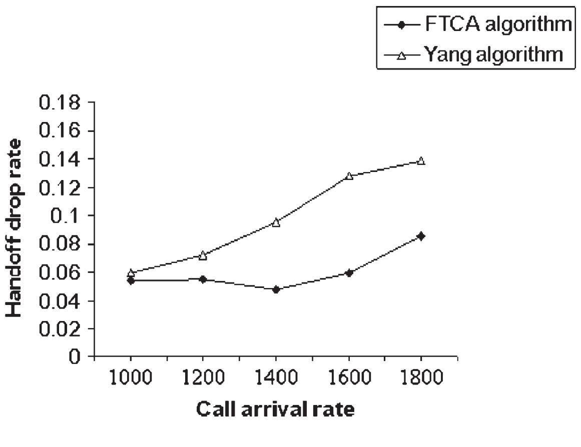

Fig. 34. Handoff drop rates for FTCA and Yang algorithm.

processed. The simulation is conducted under the nonuniform traffic pattern so that it conforms with that of [7].

The range of call arrival rate is from 1000 to 1800 call arrivals per hour per cell. The 1000 call arrival rate represents a situation where a cell is lightly loaded, whereas the 1800 call arrival rate indicates that a cell suffers a heavy load [7]. As the proposed FTCA model takes cell failure randomly, for comparison, we took the average failure of one to five of Yang's cells and compared it to the FTCA call failure under various load conditions. The graphs for call failure and handoff failure are shown in Figs. 33 and 34, respectively.

VI. Observations

From the performance graphs of the blocked hosts and the handoff failures, we made the following observations.

- The average number of blocked hosts and handoff failures is less, in comparison with the input values of hosts and channels.
- There is an obvious minimization in the number of blocked hosts and handoff failures over the successive generations, i.e., there is an improvement in channel utilization. For example, in 10 generations and 100 channels, if the hosts are 50 and 100, then there are 4.7 and 20.2 average blocked hosts, respectively, whereas for 20 generations, the corresponding results are 3.1 and 14.0, respectively.
- In the handoff failure graphs, use of the reserved channels to provide free channels to the crossing-over mobile hosts is quite observable, as it keeps the initial values of the handoff failure very low and, in many cases, minimizes it to zero.
- Efficient use of the search function for the free channels to maintain the reserved channel pool prevented the increment in the handoff failures beyond a certain maximum value.
- Although the number of channels is small compared to the number of hosts in some cases, efficient reuse of the available channels still produced a good result.
- The convergence in the values of the blocked hosts and the handoff failures is remarkable only when the number of the mobile hosts is very high compared to the number of the available channels.
- Also notable is the blocking of the channels and handoff failure, even when the number of channels is more than the mobile hosts. This is because of the channel and mobile distribution among the cells. Some cells may have fewer channels and more mobile hosts than others.

A. *Observations of Comparative Study*

- From the graphs in Figs. 30–32, it is obvious that the proposed FTCA algorithm is better than the algorithm proposed in [6] in two ways. First, there is an improvement in channel utilization, and second, the FTCA uses a fault-tolerant approach, whereas the earlier one does not.
- In the FTCA algorithm, the initial values of the blocked hosts are significantly minimized compared to the other algorithm by distributing the channels according to the initial demand of each cell.
- From the performance graphs of Figs. 33 and 34, it is obvious that the proposed FTCA model performs better than the model in [7], even in the case of a heavy load.
- Also conspicuous is the performance of the FTCA model in managing handoff (Fig. 34). The FTCA model performs much better than Yang's model for the handoff drop.

VII. Conclusion

In this paper, we proposed a GA-based fault-tolerant channel allocation model to optimize channel usage in a mobile computing network under a resource planning model. The proposed FTCA algorithm is an effective approach to maintain the network connections of wireless mobile hosts without being affected by the failures. We observed that the initial distribution of channels per cell, according to the initial demand of each cell based on the past experience and statistics, greatly reduces the number of blocked hosts. Furthermore, well-managed and efficient usage of reserved channels to handle crossover mobile hosts minimizes the number of handoff failures. Performance of the proposed model was evaluated by conducting experiments to observe the number of blocked hosts and handoff failures. It is found that over the generations, both the average number of blocked hosts and handoff failure decreases, and the result converges after certain generations. It was also shown that the FTCA algorithm performs better in terms of minimizing blocked hosts and handoff failure than the one proposed in [6] and [7] for the same purposes.

ACKNOWLEDGMENT

The authors would like to thank all the reviewers, whose valuable comments and suggestions have resulted in the quality improvement of this paper.

REFERENCES

[1] C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc Wireless Networks Architectures and Protocols*. Upper Saddle River, NJ: Pearson, 2004.

[2] D. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*. Pacific Grove, CA: Brooks/Cole, 2003.

[3] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Upper Saddle River, NJ: Pearson, 2005.

[4] M. Mitchell, *An Introduction to Genetic Algorithms*. London, U.K.: MIT Press, 1999.

[5] A. K. Tripathi, D. P. Vidyarthi, and A. N. Mantri, “A genetic task allocation algorithm for distributed computing system incorporating problem specific knowledge,” *Int. J. High Speed Comput.*, vol. 8, no. 4, pp. 363–370, 1996.

[6] S. S. Mahapatra, K. Roy, S. Banerjee, and D. P. Vidyarthi, “Improved genetic algorithm for channel allocation with channel borrowing in mobile computing,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 7, pp. 884–892, Jul. 2006.

[7] J. Yang, Q. Jiang, D. Manivannan, and M. Singhal, “A fault-tolerant distributed channel allocation scheme for cellular networks,” *IEEE Trans. Comput.*, vol. 54, no. 5, pp. 616–629, May 2005.

[8] L. Ortigoza-Guerrero and A. Hamid Aghvami, *Resource Allocation in Hierarchical Cellular Systems*. Norwood, MA: Artech House, 2000.

[9] J. C.-I. Chuang, “Performance issues and algorithms for dynamic channel assignment,” *IEEE J. Sel. Areas Commun.*, vol. 11, no. 6, pp. 955–963, Aug. 1993.

[10] R. Mathar and J. Mattfeldt, “Channel assignment in cellular radio networks,” *IEEE Trans. Veh. Technol.*, vol. 42, no. 4, pp. 647–656, Nov. 1993.

[11] R. Prakash, N. G. Shivaratri, and M. Singhal, “Distributed dynamic fault-tolerant channel allocation for cellular networks,” *IEEE Trans. Veh. Technol.*, vol. 48, no. 6, pp. 1874–1888, Nov. 1999.

[12] G. Cao and M. Singhal, “Distributed fault-tolerant channel allocation for cellular networks,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 7, pp. 1326–1337, Jul. 2000.

[13] K. Y. Lim, M. Kumar, and S. K. Das, “Message ring-based channel reallocation scheme for cellular networks,” in *Proc. Int. Symp. Parallel Architectures, Algorithms, Netw.*, 1999, pp. 426–431.

[14] A. Y. Zomaya and M. Wright, “Observation on using genetic algorithms for channel allocation in mobile computing,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 9, pp. 948–962, Sep. 2002.

[15] S. H. Wong and I. Wassell, *Dynamic Channel Allocation Using a Genetic Algorithm for a TDD Broadband Fixed Wireless Access Network*. Cambridge, U.K.: Lab. Commun. Eng., Univ. Cambridge.

[16] I. E. Kassotakis, M. E. Markaki, and A. V. Vasilakos, “A hybrid genetic approach for channel reuse in multiple access telecommunication networks,” *IEEE J. Sel. Areas Commun.*, vol. 18, no. 2, pp. 234–243, Feb. 2000.

[17] N. F. Huang and H. I. Liu, “A study of isochronous channel reuse in DQDB metropolitan area networks,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 4, pp. 475–484, Aug. 1998.

[18] M. Asvial, B. G. Evans, and R. Tafazolli, “Evolutionary genetic DCA for resource management in mobile satellite systems,” *Electron. Lett.*, vol. 38, no. 20, pp. 1213–1214, Sep. 2002.

Lutfi Mohammed Omer Khanbary received the B.E.E. degree in 1995 from Aden University, Aden, Yemen, and the M.Tech. degree in 2006 from Jawaharlal Nehru University (JNU), New Delhi, India, where he is currently working toward the Ph.D. degree in computer science with the School of Computer and Systems Sciences, on study leave from the Department of Computer Science and Engineering, Aden University.

His research interests include mobile computing, network management, and genetic algorithms.

Deo Prakash Vidyarthi received the Master degree in computer applications from M. M. M. Engineering College, Gorakhpur, India, in 1991 and the Ph.D. in computer science from Jabalpur University, Jabalpur, India (work done at Banaras Hindu University, Varanasi, India) in 2002.

He taught undergraduate and postgraduate students at the Department of Computer Science, Banaras Hindu University, for over 12 years. He is currently an Associate Professor with the School of Computer and Systems Sciences, Jawaharlal Nehru University, New Delhi, India. His research interests include parallel and distributed system, grid computing, and mobile computing.