1998 International Computer Symposium
Workshop on Computer Networks, Internet, and Multimedia
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

GENETIC ALGORITHMS FOR CHANNEL ASSIGNMENT IN PCS

Chih-Jen Chen, Ming-Feng Chang and Rong-Jaye Chen

Department of Computer Science and Information Engineering,
National Chiao Tung University, Hsinchu, Taiwan, R.O.C.
Email:{jrchen,mfchang,rjchen}@csie.nctu.edu.tw

ABSTRACT

The channel assignment problem is to assign channels to the radio base stations in a PCS in order to get better system performance. In this paper, we design a genetic algorithm to assign channels and then cite an iterative algorithm to derive the system blocking probability as the evaluation function used in the genetic algorithm. The computational result shows this approach can significantly reduce the system blocking probability.

1. INTRODUCTION

With the technology of mobile cellular telecommunication advancing rapidly, the population of mobile users will continue to grow at a tremendous speed. Because the wireless/mobile user population increases drastically, we need to know how to use scarce radio spectrum allocated to wireless/mobile communications more efficiently.

The focus of this paper is using genetic algorithms (GAs) to solve the channel assignment problem in personal communication service (PCS) so that the available frequency spectrum is employed more efficiently under the interference constraints. After using a genetic algorithm to assign channels to each cell, we need to calculate the evaluation function of the genetic algorithm. So we cite a PCS handoff model [6] and an iterative algorithm [4] to derive system blocking probability. And we use the blocking probability as the evaluation function in GA and the criteria to evaluate the system performance.

The rest of the paper is organized as follows. In section 2, we define the problem we want to solve and make some assumptions about it. Besides, we discuss the handoff scheme in our PCS model and an iterative algorithm to derive the system blocking probability. Section 3 describes the fundamentals of GAs. In section 4, we describe how to apply GAs to channel assignment problem. As for the program simulation and result comparison are reported in section 5. Finally, the conclusion is given in section 6.

2. HANDOFF MODELING

In this section, we precisely define our system topology, construct an analytic model [6] for our PCS, and show how to derive mathematical formulas of system properties.

First we list the notation used in this paper

1. m : the total number of channels for the whole system.
2. n : the system width because our system is composed of n^2 cells.
3. c_i : the total number of channels of a cell i.
4. λ_{0i} : new call arrival rate of cell i.
5. $\lambda_{in\,i}$: handoff call arrival rate of cell i.
6. $\lambda_{out\,i}$: handoff call departure rate from cell i.
7. $1/\eta_i$: the mean handset residence time.
8. p_{0i} : new call blocking probability of cell i.
9. p_{hi} : handoff call blocking probability of cell i.
10. P_0 : the new call blocking probability for the whole system.
11. P_f : the handoff call blocking probability for the whole system.
12. t_{on} : the channel occupation time of a new call.
13. t_{oh} : the channel occupation time of handoff call.

After introducing the notation, we go on to describe our system topology.

2.1 System Topology

We make three assumptions about our system topology. These assumptions are described as follows.

First assumption: We assume the topology in our system is square and is composed of n^2 small square cells. Each cell is numbered from 0 to n^2-1, and we call such a topology $n{\times}n$ system.

Second assumption: We use wraparound topology so that the handsets at boundary cells can move to their 4 neighboring cells in equal probability 0.25. The wraparound topology for a 5×5 system is depicted in Figure 1. The main purpose of this wraparound topology is to eliminate the boundary effect occurring in an unwrapped topology.

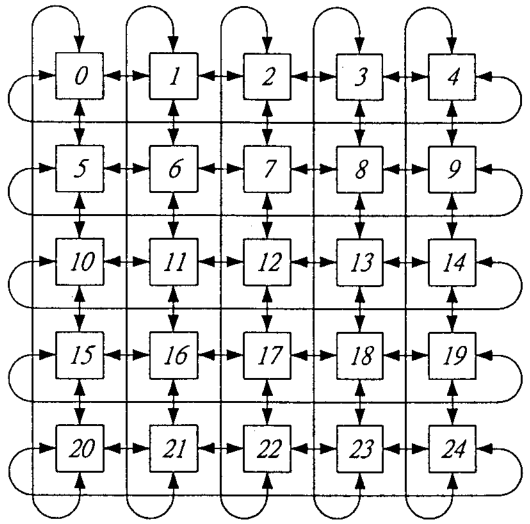

Figure 1 Wraparound topology for a 5×5 system

Third assumption: We assume that each channel used in one cell will interfere with eight adjacent cells. Disregarding whether the cell is boundary or not, there are always eight adjacent cells in our wraparound topology.

2.2 Handset Timing Diagram

Now, we describe the handoff mathematical model and the timing diagram as shown in Figure 2. We use this model to derive some important results.

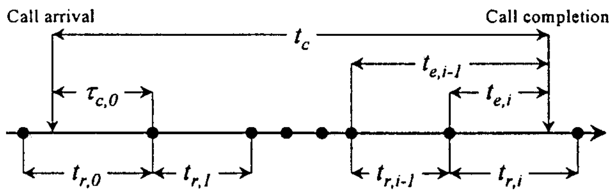

Figure 2 Handset timing diagram

The call arrivals for a handset are assumed to be a Poisson process with arrival rate λ_{0i} in cell i and each notation listed in Figure 2 is elaborated as follows.

1. t_c in Figure 2 is the call holding time of a handset and is assumed to be exponentially distributed with mean $1/\mu$.
2. $t_{r,i}$ in Figure 2 is the handset residence time of cell i and $t_{r,i}$ is assumed to be exponentially distributed with mean $1/\eta_i$.
3. $\tau_{c,0}$ is the period between call arrival and the time when the handset moves out of cell 0. Note that $\tau_{c,0} \leqq t_{r,0}$.
4. $t_{e,i}$ is the *excess life* of t_c between the time when the handset moves into cell i and the time when the call is completed. From [6, 7] we have $f_{e,i}(t)=f_c(t)$.

2.3 Derived Measures

After constructing the timing diagram, we can derive some important results. These results are listed as follows:

1. The probability $\Pr[t_c > \tau_{c,0}]$ that a new call still continues after the user leaves the cell i can be represented as [8]

$$\Pr[t_c > \tau_{c,0}] = \frac{\eta_i}{\mu + \eta_i}$$

2. The probability $\Pr[t_{e,i} > t_{r,i}]$ that a handoff call still continues after the user leaves the cell can be represented as [8]

$$\Pr[t_{e,i} > t_{r,i}] = \frac{\eta_i}{\mu + \eta_i}$$

3. The expected channel occupation time of a new call $E[t_{on}]$ can be represented as [8]

$$E[t_{on}] = \frac{1}{\mu + \eta_i}$$

4. The expected channel occupation time of handoff call $E[t_{oh}]$can be represented as [8]

$$E[t_{oh}] = \frac{1}{\mu + \eta_i}$$

2.3.1 Handoff Call Arrival and Departure Rate

From the results in the previous sub-section, we can get handoff call arrival rate and departure rate. Now we assume that we have already known p_{0i} and p_{fi}. For our system, if the adjacent cells of cell i are cell j, cell k, cell l and cell m, $\lambda_{in\,i}$ and $\lambda_{out\,i}$ can be expressed as follows:

$$\lambda_{out\,i} = \lambda_{in\,i}\,(1-p_{fi})\Pr[t_{c,i} > t_{r,i}] + \lambda_{in\,i}\,(1-p_{0i})\Pr[t_c > \tau_{c,0}]$$

$$\lambda_{in\,i} = 1/4\,(\lambda_{out\,j} + \lambda_{out\,k} + \lambda_{out\,l} + \lambda_{out\,m})$$

For initial values, we set $\lambda_{in\,i}$ and $\lambda_{out\,i}$ as follows:

$$\lambda_{out\,i} = \lambda_{in\,i}\,(1-p_{0i})\Pr[t_c > \tau_{c,0}]$$

$$\lambda_{in\,i} = 0$$

2.3.2 Blocking Probability

Besides handoff call arrival rate and departure rate, we can get blocking probability for each cell. If we adopt a non-prioritized handoff scheme where the handoff calls and the new calls have the same priority to access free channels, we get $p_{0i}=p_{fi}$ for each cell i. Now we assume each cell under study is a c-server blocking system with general service times (*M/G/c/c* system). According to the queueing theorem [5], we get the net traffic to a cell is

$$\rho_n = \lambda_{0i} E[t_{on}] + \lambda_{hi} E[t_{oh}]$$

So the blocking probability is

$$p_{oi} = p_{fi} = B(\rho_{ni}, c_i) = \frac{\left(\rho_{ni}^{c_i}/c_i!\right)}{\sum_{j=0}^{c_i}\left(\rho_{ni}^{j}/j!\right)}$$

2.4 An Iterative Algorithm

After deriving these results, we cite an iterative algorithm [4] to find each value of these results. This algorithm is described below:

Step 1: Compute the initial values of $\lambda_{in\,i}$ and $\lambda_{out\,i}$ for each cell i:

Step 2: Compute the new call blocking probability p_{0i} and handoff call blocking probability p_{fi} for each cell *i*:

Step 3: Compute the new handoff call arrival rate $\lambda'_{in\,i}$ and new handoff call departure rate $\lambda'_{out\,i}$:

Step 4: If for all cell i $|\lambda'_{out\,i}-\lambda_{out\,i}|>\delta\lambda_{out\,i}$, we set

$$\lambda_{out\,i}=\lambda'_{out\,i}$$

$$\lambda_{in\,i}=\lambda'_{in\,i}$$

Then go to Step 2. Otherwise, go to Step 5.

Step 5: Compute the system blocking probability:

$$P_0 = P_f = \frac{\sum_{i=1}^{n} \lambda_{0i} P_{0i}}{\sum_{i=1}^{n} \lambda_{0i}}$$

After the algorithm converges, we get the system blocking probability P_0 and P_f in step 5. We use the value P_f as our evaluation function of GA and use the value of $10*(1-P_f)$ as the fitness function.

3. GENETIC ALGORITHM

In this section, we introduce the conventional GAs [3] and its primary genetic steps. We introduce these steps by turns.

3.1 Initialize the Population

First, we must decide the *population size* to form the first generation. Generally speaking, population sizes depend upon the problem and the computer resource. Larger populations ensure greater diversity whereas smaller populations help to simplify examination of the GA performance from one generation to the next. Once the population size is chosen, then the initial population must be randomly generated.

3.2 Evaluate Fitness

Now that we have a population of potential problem solutions, we need to judge how good they are. In GA, we use the *evaluation function* as well as the *fitness function* for the judgement. The distinction between the two functions is that the evaluation function provides a measure of an individual's performance, while the fitness functions provides a measure of an individual's reproduction opportunities.

3.3 Reproduction

Reproduction is a process, which actually reproduces copies of individuals to form the intermediate population according to their fitness value. The individuals are reproduced by a roulette wheel strategy, which means that individuals with a higher fitness value have a higher probability of contributing one or more offspring in the next generation.

3.4 Crossover

The crossover operator has been described as the key to GA's power [2], as it promotes structured yet randomized information exchange between individuals. A simple crossover may proceed in two steps. First, members of the newly reproduced individuals in the mating pool are coupled at random. Second, each couple exchanges either portion to produce two children.

3.5 Mutation

In simple GAs, mutation is an occasional operation with small probability random alteration of the value of a string position which simply means changing a 1 to a 0 and vice versa and mutation rates are similarly small in natural populations.

3.6 Survival Policy

A survival policy determinates which individuals populate the new generation. There are many policies [1] have been derived, such as pure, keep-incumbent, no-worse, keep-incumbent and no-worse, and best-two policy. Some of these policies can guarantee that the best fitness value or the worst fitness values in the generations are non-decreasing.

4. GA FOR CHANNEL ASSIGNMENT

We formulate the channel assignment problem as a discrete optimization problem and apply GAs to it in order to minimize the system blocking probability. So we define the corresponding populations, discrete individuals, operators, survival policy, fitness function and evaluation function of our GAs.

4.1 Individual and Population Representation

Now, we illustrate how the channel assignment problem can be manipulated by GAs. We have assumed that a mobile radio network is a $n\times n$ system composed of n^2 radio cells, each of them capable to carry any of the m channels that are available for the whole system. We suppose that each individual represents a state of the channel assignment for the whole system. So, an intuitive choice is given by a binary matrix (s_{ij}) of dimension $m\times n$ with the following interpretation of solution entries:

$$s_{ij} = \begin{cases} 0 \\ 1 \end{cases} \text{if channel } i \text{ is} \begin{Bmatrix} \text{not used} \\ \text{used} \end{Bmatrix} \text{at radio cell } j$$

After defining the representation of individuals, we can simply gather an apposite number of individuals, say 20, to compose a population. Apposite number here will provide enough diversity of individuals and will not require heavy computer resource.

4.2 Fitness Function and Evaluation Function

We use the system blocking probability P_f derived before as the evaluation function. The fitness function in our method is set to be $10*(1-P_f)$. We use the fitness function to perform reproduction operation and use evaluation function to evaluate the performance of each individual.

4.3 Initializing Population

Before initializing, in order to prevent some cells from getting no channels, we previously assign $m/2$ channels to the whole system and let each cell get $m/8$ *basic channels*. This can be achieved by using four compact assignments. For example, there are four kinds of compact assignment for an 8×8 system shown in Figure 3.

Now, we will describe how to assign one channel to the whole system in order to initialize population. The steps of the algorithm are described below:

Step 1: Evaluate the blocking probability of each cell and then sort all cells according to their blocking probability. Rank 0 represents the cell with the worst blocking probability.

Step 2: If we assign channel j, we repeat step 2 until k times. k is a positive small integer, which depends on the system size. k means the times that we force to assign channel to the cell according to blocking probability. In each time, we assign this channel directly to a cell according to the rank of step 1. For example, if we choose cell i to be assigned, we set $(s_{ji})=1$. After choosing cell i, we should find out all the cells interfered by cell i. We must mark these interfered cells in order to prevent choosing them in the next time.

Step 3: We repeat step 3 until no cell can be chosen (i.e., all cells are marked). After assigning k cells with higher blocking probability, we construct a roulette wheel with slots sized in proportion to the blocking probability of each cell. We randomly assign this channel to cells using the roulette wheel.

We can repeat the algorithm m times to assign total m channels to our system and get an individual composed of m rows. Then we can gather enough individuals to organize the population for our model.

4.4 Reproduction Representation

We adapt conventional reproduction operators in our method. While conventional reproduction uses a roulette wheel strategy to derive the next generation according to their fitness function, we use the system $10*(1-P_f)$ as the fitness function.

4.5 Crossover Representation

Because of the individual structure, we can exchange rows arbitrarily between two individuals without breaking the

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

1st compact assignment

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

2nd compact assignment

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

3rd compact assignment

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

4th compact assignment

0	1	2	3	4	5	6	7
8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23
24	25	26	27	28	29	30	31
32	33	34	35	36	37	38	39
40	41	42	43	44	45	46	47
48	49	50	51	52	53	54	55
56	57	58	59	60	61	62	63

Merge the four compact Assignment

Figure 3 Four compact assignments for an 8×8 system co-channel reuse constraint. And in every crossover, we must make the evaluation function have chances to be improved. We implement crossover operation described as follows:

Step 1: If our population size is 100, we randomly divide 100 individuals into 50 couples. Each couple is ready to perform crossover.

Step 2: For all couples (*M, F*) in the mating pool, we want to perform crossover over each couple. We repeat Step 2 until we process all couples.

Step 2.1: For couple (*M, F*), we randomly choose a number k ($1 \leq k \leq m$). Then we want to exchange k rows between individual *M* and individual *F*.

Step 2.2: Each time we exchange one row, we sort each cell in individual *M* according to their blocking probability first. If we find the worst cell x of individual *M* without using channel l but the cell x of individual *F* using channel m, we exchange row l of individual *M* for row m of individual *F*. We repeat step 2.2 until we exchange k rows.

4.6 Mutation Representation

We know that mutation is needed because reproduction and crossover may occasionally become over zealous and lose some potentially useful genetic material. We set the mutation rate equal to 0.02, and in every generation, there are approximately $0.02*N*m$ (N is the population size, m is the total number of channels) mutations occurring. Our mutation is described below:

Step 1: Calculate the value of k=$0.02*N*m$ and we perform mutation k times.

Step 2: Repeat Step 2 k times.

Step 2.1: Randomly choose one individual i from population and randomly choose one row r form individual i.

Step 2.2: Remove row r from individual i, calculate the blocking probability for each cell. Then sort these cells according to their blocking probability.

Step 2.3: Using the rank of blocking probability to reconstruct row r of individual i.

4.7 Survival Policy

Our survival policy is keep-incumbent. In every generation, we always keep a position for the best individual. So we can guarantee our best evaluation function value in the generations is non-decreasing.

5. COMPUTATIONAL RESULT

We consider an 8×8 system with 80 channels needed to be assigned. 40 channels are assigned by GA and the other 40 channels will be assigned by four compact assignments to each cell. We use uniform distribution in [2/minutes, 7/minutes] to generate the new call arrival rate for each cell. The mean handset residence time for each cell is uniform distribution in [1 minutes, 4 minutes]. The call holding time is exponential distribution with 3 minutes for all cells.

We write a C program and run it on Sun SPARC system 600. The CPU time spent is 344 minutes. Figure 4 shows the system performance. The system blocking probability decreases from 2.95% to 1.44% in 400 generations.

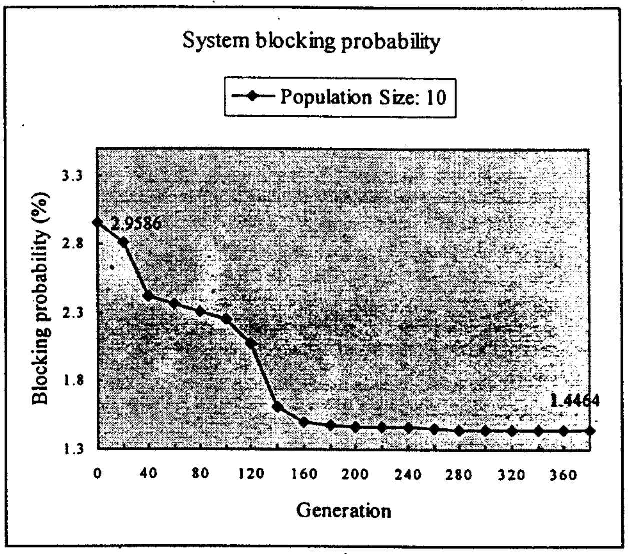

Figure 4 Performance of GA

Now we compare the two simple methods with our GA. The first method is average assignment that assigns 80 channels to 64 cells and makes each cell receive 20 channels. The second method is called always-assign-worst that assigns channel to the cell with the worst blocking probability. The system blocking probabilities for these two methods are 2.99% and 2.36% respectively, which are much higher than our minimized results 1.44%.

6. CONCLUSION

GA as a general algorithm for combinatorial optimization has been systematically applied to some forms of the channel assignment task of radio network planning. In this paper, we apply GA to solve the channel assignment problem and cite a handoff model to calculate the system blocking probability. Finally we find that GAs are very flexible because they can be modified in many ways to improve the final result. There are a lot of new rules we can add to adjust our algorithm without imposing the extra computational burden of checking for their consistency with the existing rules.

There is much work to be done in the future. The first is to modify the three primary operators: reproduction, crossover, and mutation to improve performance. In addition to modifying three operators, we can adjust crossover rate, mutation rate, and other parameters used in our GA to make GA converge faster and reach a better optimum. It is also significant to substitute hexagonal cell for square cell, and may face some difficulties trying to solve it. There are some other problems happening when writing the simulation program, and they need to be addressed on later. After all, GAs appear to be a valuable approach for practical radio cellular network design, and thus worthwhile to be studied further on.

7. REFERENCES

[1] R. J. Chen, Robert R. Meyer, and Jonathan Yackel, "A Genetic Algorithm for Diversity Minimization and Its Parallel Implementation," 5th International Conference on Genetic Algorithms, UI-Urbana-Champaign, July 1993.

[2] L. Davis and M. Steenstrup, "Genetic Algorithms and Simulated Annealing: an Overview," in Genetic Algorithms and Simulated Annealing, L. Davis, Ed. Morgan Kaufmann, pp. 1-11, 1987.

[3] D. E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Press, 1989.

[4] D. Hong and S. S. Rappaport, "Traffic Model and Performance Analysis for Cellular Mobile Radio Telephone Systems with Prioritized and Non-Protection Handoff Procedure," IEEE Trans. Vehicular Tech., Vol. VT-35(3), pp. 77-92, 1996.

[5] L. Kleinrock, Queueing Systems: Volume I - Theory, New York: Wiley, 1976.

[6] Y. B. Lin, Introduction to Mobile Network Management, Weikeg Press, 1997.

[7] Y. B. Lin, Seshadri Mohan, and Anthony Noerpel, "Queueing Priority Channel Assignment Strategies for PCS Hand-off and Initial Access," IEEE Trans. Vehicular Tech., Vol. 43, pp. 704-712, 1994.

[8] Y. B. Lin, "Performance Modeling for Mobile Telephone Networks," IEEE Network, Vol. VT-11(6): pp. 63-68, 1997.