Genetic Algorithm for Broadcast Scheduling in Packet Radio Networks

Goutam Chakraborty, Yuuske Hirano

Abstract— **Packet Radio (PR) networks are to provide data communication among a set of nodes distributed over a region. A time division multiple access (TDMA) protocol is adopted for conflict free communication. The goal is to find a conflict free transmission schedule for different nodes at different time slots of a fixed length time cycle, called TDMA cycle. The optimization criterion is primarily to minimize the TDMA cycle length, and then to maximize the number of transmissions. First classical Genetic Algorithm is tried to solve this NP-complete problem, which showed poor performance for bigger networks. Then we proposed some special crossover operators suitable for this kind of problem. This modified operator could deliver very good quality of results even for big networks and in few generations. Some study on the dependence of the result on population size etc. are studied. The results are empirically compared with other approaches, a greedy-heuristic algorithm and mean field annealing.**

Keywords— **Genetic algorithm, Evolutionary programming, Broadcast scheduling.**

I. Introduction

Packet Radio (PR) network [1] is a good option for wireless communication over a wide geographical range. Here each station can transmit or receive, controlled by its control unit. When a node transmits, all its neighbours (determined by distance) can receive. The neighbour node could absorb the packet, if it is so designated. Else it may store it to transmit it later. In that case it acts as an intermediate repeater, when there is no direct connectivity available between the source and the destination.

In PR network, a single wide band Radio channel is used by all nodes. A multiple access protocol, called time division multiple access (TDMA) protocol, is used [2]. The transmissions of packets are controlled by a single clock. The time is divided into distinct frames consisting of a number of time slots, equal to the transmission time required for single packet to be transmitted and received by the neighbouring nodes. Many nodes may transmit simultaneously without conflict if they are far apart. But all the nodes must be able to transmits at least once in a TDMA cycle. This is termed as *no-transmission* constraint.

The basic optimization objective is to get the smallest length TDMA cycle, where many nodes are allowed to transmit simultaneously in a slot. The secondary objective is to maximize the number of such transmissions for maximum utilization of the channel. There is a third possible objective so that the waiting time for all the nodes are more concentrated towards the average - a measure of fairness, which we will not consider in this work.

In addition to *no-transmission* constraint, there are other constraints, namely the *primary conflict* and the *secondary conflict*. *Primary conflict* says that a particular node can not transmit and receive in the same time slot i.e. two neighbouring nodes can not transmit simultaneously. A *secondary conflict* occurs when two or more packets arrive at a node in a single time slot. This will occur when two nodes at a distance of two hops are allowed to transmit simultaneously. Then the intermediate node will receive two different packets at the same time slot. The aim is to schedule the transmissions in the TDMA cycle such that the *primary* and the *secondary conflicts* are avoided.

This scheduling problem is proved to be NP-complete [3]. Several heuristics and other algorithms are proposed. We will use GA to solve this problem and will empirically compare some of our results with a distributed greedy algorithm proposed in [3] and one using mean field annealing [4].

A formal definition of this scheduling problem is done in section II. It is trivial to map this problem to GA. We first tried to solve this problem using classical GA [5] [6]. This approach is discussed in section III. It gave somewhat good results for very small networks (number of nodes < 50), but failed for larger nets. We proposed some special crossover operator suitable for this problem, explained in section IV. The details of simulation and results, discussions about parameters and empirical comparison with earlier works are in section V. Section VI is the conclusion.

II. The Problem

PR network can be represented by a graph $G = (V, E)$, where $V = \{v_1, v_2, ..v_i, ..v_N\}$ is the set of nodes and $E = \{e_1, e_2, ..e_P\}$ is the set of undirected edges. The existence of an edge between two nodes means that the two nodes can receive packets transmitted from each other. The neighbouring information i.e. the connectivity among network nodes are described by a $N \times N$ symmetric matrix C, where

$$c_{ij} = \begin{cases} 1, & \text{if } v_i \text{ and } v_j \text{ are connected} \\ 0, & \text{otherwise} \end{cases}$$

When two nodes are so connected, we say that they are one *hop* apart. We also assume slotted time and constant packet length i.e. one packet length = one slot. A time frame i.e. a TDMA cycle consists of a fixed number of such time slots. A number of packets can be transmitted in a time slot from different stations if that does not create any interference. We consider a fixed schedule of transmission for nodes in a TDMA cycle. Once the optimum transmission pattern for a frame is decided, the same

Both the authors are with the University of Aizu, Aizu Wakamatsu shi, Japan - 965-80. E-mail: goutam/s1021056@u-aizu.ac.jp .

0-7803-4869-9/98 $10.00©1998 IEEE

frame is repeated over time. Let us denote such a TDMA frame by a $M \times N$ matrix T, where

$$t_{mj} = \begin{cases} 1, & \text{if } v_j \text{ transmits in time slot } m \\ 0, & \text{otherwise} \end{cases}$$

When node v_i transmits a packet, none of its neighbours i.e. nodes which are one *hop* away, are allowed to transmit in the same time slot, as this would give rise to *primary conflict*. All nodes, two *hops* away from v_i, should also be disabled to transmit simultaneously with v_i, as this would create *secondary conflict* of multiple reception at intermediate nodes. All these nodes which are one *hop* and two *hops* away from v_i form what is called the *broadcasting zone* [3] of v_i. The set of these nodes we denote by B_i. It is clear that non-interference requires that none of the nodes in B_i should be allowed to transmit simultaneously with v_i. From this concept we can generate a $N \times N$ matrix D, where

$$d_{ij} = \begin{cases} 1, & \text{if } v_j \in B_i \\ 0, & \text{otherwise} \end{cases}$$

The problem is to find optimum TDMA cycle, such that the following constraints must be satisfied.

- Every stations should be scheduled to transmit at least once i.e. *no-transmission* constraint is satisfied.

$$\sum_{m=1}^{M} t_{mi} \geq 1 \qquad \forall i \tag{1}$$

- A station can not transmit and receive packets in the same time slot to avoid *primary conflicts.*
- A station can not receive two or more transmissions simultaneously i.e. *secondary conflicts* are to be avoided. Formally,

$$if \;\; t_{mi} = 1, \; then \;\; \sum_{j=1}^{N} t_{mj} d_{ij} = 0 \;\; \forall m, i$$

i.e.,

$$\sum_{m=1}^{M} \sum_{i=1}^{N} \sum_{j=1}^{N} t_{mi} t_{mj} d_{ij} = 0 \tag{2}$$

The last two conflicts are avoided if T is scheduled satisfying Eq. (2). A trivial solution satisfying all the three constraints is a N-slot TDMA frame, where N different stations transmit in N different time slots.

The optimization criteria are as follows:

- The length of TDMA cycle should be as short as possible, i.e. M should be as small as possible.
- To maximize the total number of transmissions. This is represented by channel utilization index ρ, where

$$\rho = \frac{1}{M \times N} \sum_{m=1}^{M} \sum_{j=1}^{N} t_{mj} \tag{3}$$

It is trivial that if the maximum degree of a node in the net is X, i.e.

$$X(G) = \max_{v_i \in V} deg(v_i)$$

then the tight lower bound for M could be written as,

$$M \geq (X + 1) \tag{4}$$

As we do not know of any algorithm to find the optimum solution, we will use this tight lower bound as an empirical measure to judge the quality of our solution.

A sample network with B sets, C and D matrices are shown in Fig. 1. A trivial TDMA solution, which satisfies all the constraints but does not optimize any of the optimization criterion is shown, together with an optimum TDMA cycle. To find such an optimum TDMA schedule is a NP-complete problem. We will try solving this using genetic algorithm, which is discussed in the next section.

III. Traditional Genetic Algorithm Approach

As already mentioned, we first tried classical Genetic algorithm to solve the scheduling problem. It failed to give good results for any reasonable sized network. We then proposed new crossover operator appropriate to this problem. The first approach can be termed as a 'weak' method [7], where little assumptions about the problem domain is considered. It failed. The second approach is a 'stronger' one in the sense that we used some knowledge from the problem domain and exploited them to design our crossover operator. We had very good results. We describe these two algorithms in the next two subsections.

The basic difficulties to map a problem so as to be able to use genetic algorithm, is, first to represent the solution by binary strings and then to find proper evaluation function to judge the fitness value of different competing solutions. TDMA cycle is in binary. Therefore there is no coding problem. We need only to evaluate different solutions. Classical crossover operator will generate invalid solutions difficult to evaluate. We therefore used some penalty function for invalid individuals. The steps of the algorithm are as follows.

A. Initial Population

For a N-node PR network, as already indicated, a trivial solution is a TDMA cycle with N time slots. Different nodes are allocated in different time slots, so that there is no interference. This solution is far from optimal though. A number of such trivial solutions could be generated by randomly selecting different time slots for different nodes i.e. no two nodes are allotted in the same time slot. While creating such chromosomes, we do not bother multiple identical copies of solutions. In all our experiments population size remains fixed through all generations. Experiments were done with different population sizes. Let us denote the population size by P.

B. Evaluation and Selection of Chromosomes

While we use the traditional crossover, it is possible that the offsprings created violate the constraints. To penalize such infeasible solutions we used a penalty function, $\boldsymbol{Pen}(\boldsymbol{x})$, where $\boldsymbol{x}$ denotes the chromosome. Instead of actually evaluating different chromosomes, we rank them according to their goodness. The best solution is assigned

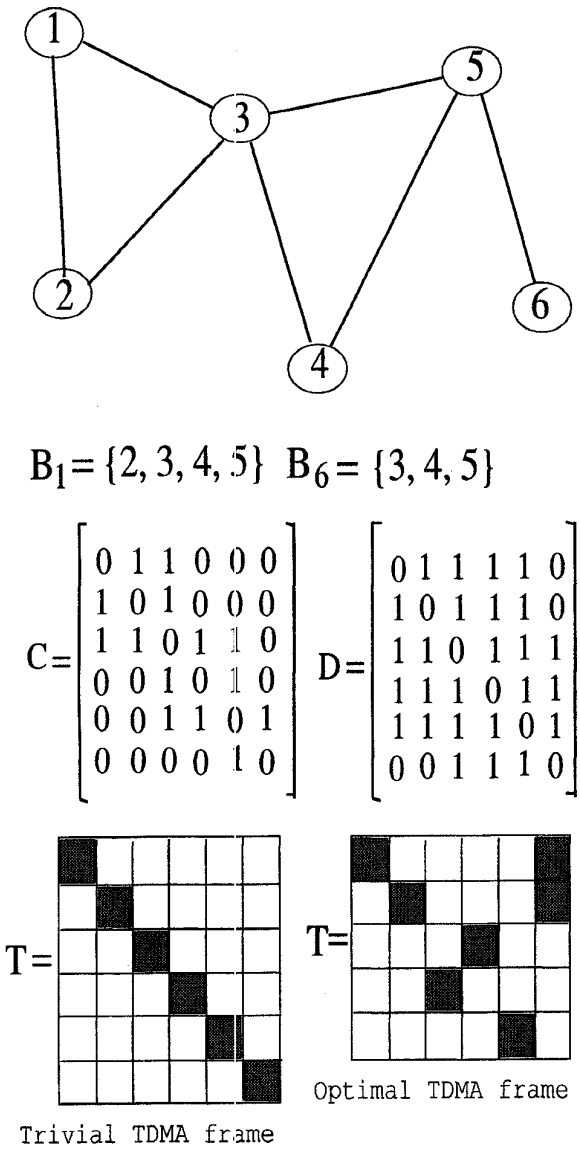

Fig. 1. Example of C and D matrix

rank 1, and the worst rank P, the population size. The ranking of the different chromosomes are done using the following algorithm.

step 1 First we check if the chromosome is a feasible solution or not?

Step 2 If it is a feasible solution:
The chromosome with lesser number of time slots will have lower rank. For two chromosomes with equal number of time slots, the one with more transmissions will have lower rank.

Step 3 If it is an infeasible solution:
Calculate a quadratic penalty function $\boldsymbol{Pen}(\boldsymbol{x})$ for the chromosome $\boldsymbol{x}$ as follows.

$$Pen(x) = (P1 \times (ntr + cf/(P2 \times N)))^2 \quad (5)$$

where,
ntr = the number of nodes failed to transmit
cf = number of *primary* and *secondary* conflicts
N = number of network nodes
$P1, P2$ = parameters to tune $\boldsymbol{Pen}(\boldsymbol{x})$
Rank of $\boldsymbol{x}$ increases with its $\boldsymbol{Pen}(\boldsymbol{x})$ value.

Once all the chromosomes are serially arranged using the above algorithm, lower rank i.e. better chromosomes first, they are given the final ranking with rank = 1 to the best solution and rank = P (population size) to the worst solution.

Once the chromosomes are ranked, their selection to the next generation is done using a nonlinear probability function (chap.4 of [6]),

$$prob(rank) = q(1-q)^{rank-1}$$

For any chromosome $\boldsymbol{x}$, if a random number generated between 0 to 1 is less than $\boldsymbol{prob(rank(x))}$, $\boldsymbol{x}$ is selected to the next generation.

C. Crossover and Mutation

As time slots are the time units of transmission, we did crossover on the rows of a chromosome, not on the whole TDMA frame. The total number of time-slots in the whole population is $P \times N$ initially. We select rows from different chromosomes (TDMA frames) using predetermined crossover probability, and gather them in the mating pool. For crossover between two strings from the pool, a cut position is determined randomly and the classical crossover operation is done. If in a valid offspring, all entries for a row becomes 0, that row is erased from that chromosome. During crossover violation of constraints are not examined. If an individual becomes an invalid solution, it is penalized as discussed in the previous section.

Simple mutation is used where bits are selected using mutation probability.

It will be seen that the classical GA failed to solve complex problem with large number of nodes. By allowing generation of invalid solutions using simple crossover, the search space explodes. It was impossible to find good solutions.

IV. Description of Our Algorithm

We proposed new modified crossover operator to solve this problem. We used the knowledge of the problem i.e. the constraints to design the crossover operation. Very good results even for complex problems were obtained.

A. The Proposed Algorithm

The reason for failure of the traditional genetic algorithm for TDMA scheduling is to allow crossover and mutation operators to generate invalid solutions. We modified crossover operator so that invalid solutions are prevented. The new crossover operator is designed using knowledge of

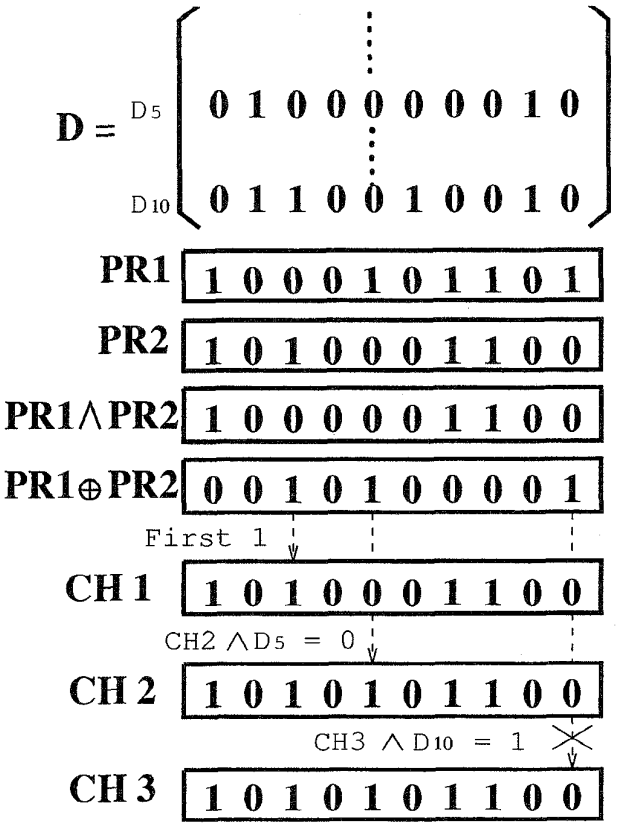

Fig. 2. An example of crossover operation, for a 10-node network

constraints of the problem. The details of the algorithm is described below.

A.1 Initialization

The initialization is done exactly the same way as for the traditional genetic algorithm.

A.2 Evaluation and Selection of Chromosomes

In this case crossover is done such that the solutions always satisfy all the required constraints. The comparison of fitness among solutions is now easier. We use the following criterion.

- The chromosome with lesser number of time slots is a better solution and will have lower rank. For two chromosomes with equal number of time slots, the one with more transmissions will have lower rank.

We use *tournament selection* [8] in this case. Here, some predefined number (say k, the tournament size) of chromosomes are randomly chosen from the whole population. The best among them goes to the next generation. This process is repeated P (population size) number of times. For $k = 2$ the selection pressure is very low, and increases with the value of k. We tried with different values of k and the results will be discussed in section V.

A.3 Crossover and Mutation

As before crossover is done on rows of TDMA cycle. Rows from different chromosomes (TDMA frames) are selected using the predetermined crossover probability, and are marked to form the mating pool. A pair of rows are selected randomly for crossover. These two parents are named as $PR1$ and $PR2$. The child is named CH which grows in number of 1s, forming a better time-slot with more transmissions. The maximum possibility is to have a child which is logical OR of $PR1$ and $PR2$. But it may violate conflicts described by matrix D. The final child string CH is created using the following algorithm, illustrated in Fig. 2.

step 1 First $PR1$ is logically ANDed with $PR2$ and written on CH i.e. $CH \Leftarrow PR1 \wedge PR2$, where $\wedge$ denotes logical AND operation. It is trivial that CH does not violate any constraint.

Step 2 $PR1$ is logically exclusive-ored with $PR2$ and $PR1 \oplus PR2$ is stored. This XOR string is used for possible addition of 1s (which means more transmissions to that slot) to CH string.

Step 3 The $PR1 \oplus PR2$ string is scanned from left to right. The first 1 encountered is copied to the same position of CH string replacing 0. This is shown as $CH1$ in Fig. 2. This addition of 1 would not violate any constraints, as CH would still be a subset of $PR1$ or $PR2$, both of which do not violate *primary* or *secondary conflicts.*

Step 4 We go further right on the $PR1 \oplus PR2$ string till we encounter the next 1, say at i^{th} bit position. This 1 is temporarily copied to CH string, and logically ANDed with the i^{th} row of D matrix. If the resulting string is all 0, it is easy to see that there will be no *primary* or *secondary conflict* by adding this 1 to the i^{th} position of the CH string. This 1 is then added to CH, which becomes the current CH. If the ANDing results in a non-zero string, this 1 at i^{th} position is not added, as it would violate *primary or secondary conflict.* The XOR string is scanned further to the right and the above procedure is repeated, till the end of the string is reached.

Step 5 In strings $PR1$, $PR2$ and CH, a 1 denotes transmission allowed for a particular station. Thus corresponding to a string, say $PR1$, there is a set of stations for which transmissions are allowed. We denote it by $\{PR1\}$. After step 4 is completed, there are three possible outcomes (See Fig. 3).
(1) $\{CH\} \supset \{PR1\}$ and $\{CH\} \supset \{PR2\}$.
In this case (Fig. 3(b)) CH replaces both the rows corresponding to $PR1$ and $PR2$.
(2) $\{CH\}$ is superset of either $\{PR1\}$ or $\{PR2\}$.
In that case (Fig. 3(c)) CH replaces only that parent which is its proper subset. The other parent remains unchanged.
(3) Neither of the above two cases are true (Fig. 3(d)). Do nothing.

Step 5 When a parent PR is replaced by the offspring CH, all rows of that TDMA frame, which are subsets of $\{CH\}$ are erased.

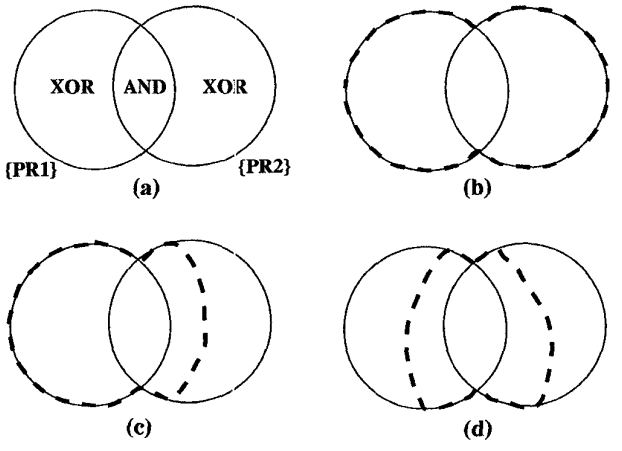

Fig. 3. Three possible cases for CH set

Following parameters are common for both TABLE I & TABLE II
Population Size = 100, Maximum Generation = 1000
Probability of crossover = 0.3, Prob. of mutation = 0.001
Avg node degree = 3, Tournament size = 3. X is maximum degree.
a: avg. number of time slots. ρ = Channel utilization index

# of Nodes	10	20	30	40	50	60
$X+1$	6	5	6	6	10	7
a	6.0	9.6	21.1	28.0	49.2	59.9
ρ	0.183	0.208	0.208	0.169	0.023	0.017

TABLE I

RESULTS OBTAINED USING CLASSICAL GA

# of Nodes	10	20	30	40	50
$X+1$	6	5	6	6	10
a	6.4	7.2	9.1	10.0	12.6
ρ	0.183	0.199	0.194	0.198	0.199

# of Nodes	60	70	80	90	100	200
$X+1$	7	7	8	8	8	8
a	12.9	13.5	15.3	16.7	18.0	29.9
ρ	0.188	0.192	0.196	0.178	0.198	0.193

TABLE II

RESULTS OBTAINED USING MODIFIED CROSSOVER

Simple mutation is done based on mutation probability, but every time the broadcast matrix is checked. If the mutation created an invalid frame, it is discarded. **Theorem**: All chromosomes in every generations will satisfy all the constraints.

Proof: It is trivial that the TDMA frames, after crossover and mutation operation, do not create any chromosome that would violate either the *no-transmission constraint* or *primary* or *secondary conflict constraints*. We have started with an initial population of which all the members satisfy all the constraints. Therefore, the solutions after each generations will also satisfy them.

V. SIMULATION AND RESULTS

A. Creating the Network

We created random fully connected networks of different average node degrees. A network of node N and average degree d is created as follows:

1. Create an area of $\sqrt{N} \times \sqrt{N}$. Put N points on it randomly.
2. All points are connected, so that the degree of every node is exactly 2. The probability of connection between two points decreases exponentially with the distance and is given by the following equation.

$$Prob = exp\left(-\frac{distance\ \ between\ \ nodes}{L_{max} \times \alpha}\right)$$

   L_{max} the diagonal distance of the area, α a constant.
3. The rest $N(d-2)/2$ links are again added using the same probability function as in the previous step.

B. Results with Traditional Genetic Algorithm

Several experiments are done to solve the TDMA scheduling problem of different sizes with many variations of population size, the penalty function, penalty function parameters, and probability function for the selection from ranked chromosomes. In Table 1 the average of results from 30 runs are shown, with fixed parameter values. For a network with number of nodes $N \leq 50$, traditional genetic algorithm could find somewhat good solutions. But when the size of the network is more than 50 nodes, the length of the TDMA cycle for valid solutions is hardly changed i.e. classical GA fails for big networks.

There are several possible reasons for this failure. The most important reason is that by allowing infeasible solutions the search space explodes. Thus a crossover that create infeasible solutions should be avoided. Secondly, by uniform ranking we lose the information about the relative goodness of different solutions, and the selection pressure becomes weak. Considering these aspects we look for some new crossover operator that would generate feasible solutions only.

C. General Results with the New Crossover Operator

Results, as shown in Table 2, are much improved when the proposed crossover is used. It is evident that, (1) it could deliver good solutions even for very big networks, and (2) the results obtained are very stable i.e. the variance of TDMA cycle length over the mean is very low. All these experiments are done with the same population size.

A typical run is shown in Fig. 4 for a network with 400 nodes and with a node average degree of 5. This is quite complex problem. We could see in Fig. 4 that the TDMA cycle reach its stable value quite soon, in less than 300 generations. Also the quality of the result is also very good with cycle length less than 15% of N. But the transmission efficiency still increases evern after 600 generations. Thus, depending on the time available, we could run the algorithm for some time and still have a pretty good result.

In [4] mean field annealing method was used to solve the same problem, with number of nodes up to 40 only. The quality of results are poor compared to our results,

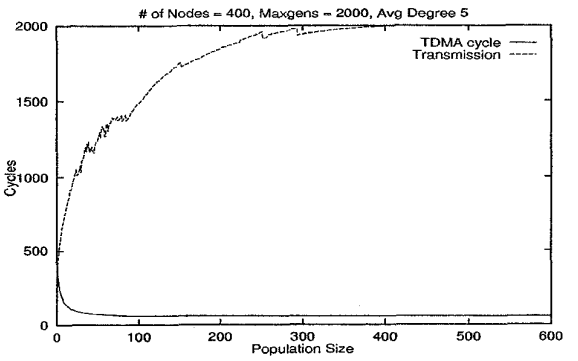

Fig. 4. Typical example of a run

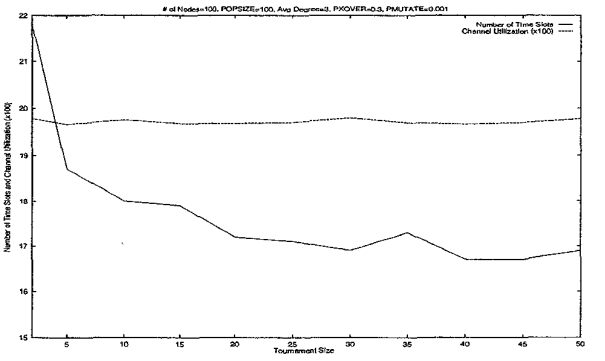

Fig. 6. Result with varying tournament size

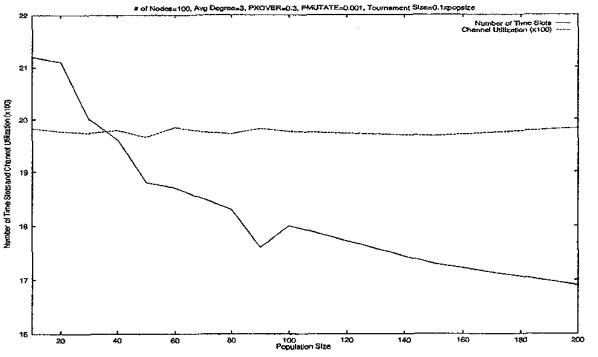

Fig. 5. Result with varying population size

with respect to both optimization criteria. We could not simulate the algorithm due to lack of informations about implementation.

For distributed greedy algorithm [3], the TDMA cycle lenght is alway equal to N, whereas we could always find much smaller TDMA cycle, as shown in Table. 2. Also, the average transmission efficiency for the greedy algorithm is only 3% (for big networks) to 4% (for smaller nets). Our results show transmission efficiencies of 8% to 11%.

D. Population Size, Tournament Size and the Result

The effect of changing population size (tournament size is kept at 10% of the population size) and tournament size is shown in Fig. 5 and Fig. 6 respectively. As is obvious, the quality of results improves with population size. With bigger tournament size the selection pressure is high, and we get better results.

VI. Conclusions

The problem of transmission scheduling for PR network is NP-complete. We tried traditional genetic algorithm to solve the problem, using classical crossover operator. Though it gave good results for small problems, it failed for any reasonable size network. We then proposed a new crossover operator, using some problem specific knowledge to avoid infeasible solutions. Excellent results are obtained for big networks with complex connectivities. We did not use any evaluation function which gives quantitative values for the fitness. Instead we evaluated only relative quality of the solutions, which forced us to use ranks and tournament selection. We would further like to propose some quantitative evaluation function for the solutions to be able to experiment with some other selection methods. Also we would like to experiment adding mutation operator. The main message of this work is that, with proper modification of the operators, evolutionary programming can be successfully applied to a wide variety of problems.

A further optimization criterion could be added to this problem. An undesirable schedule may result, when a small subset of nodes are allowed to transmit many times while others transmit only once or twice in a TDMA frame. Then the average waiting time for nodes to transmit will be high. So another optimization index could be lowering of the average waiting time. We will next extend our work adding this optimization criterion.

References

[1] B. M. Leiner, D. L. Nielson, and F. A. Tobagi, "Issues in packet radio network design," Proc. IEEE, vol. 75, no. 1, pp. 6-20, Jan. 1987.

[2] R. Nelson and L. Kleinrock, "Apatial TDMA: a collision-free multihop channel access protocol," IEEE Trans. Communication, vol. COM-33, pp. 934-944, Sep. 1985.

[3] A. Ephremides, and T. V. Truong, "Scheduling broadcasts in multiple radio networks," IEEE Trans. Communication, vol. COM-38, no. 4, pp. 456-460, April, 1985.

[4] G. Wang, and N. Ansari, "Optimal broadcast scheduling in packet radio networks using mean filed annealing," IEEE journal on selected areas in communication, vol. 15, no. 2, pp. 250-260, Feb, 1985.

[5] David E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addospm-Wesley, 1989.

[6] Zbigniew Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs.* Springer-Verlag, 1995.

[7] Kenneth A. De Jong, William M. Spears, "Using Genetic Algorithms to Solve NP-Complete Problems," Proceedings of the Third International Conference on Genetic Algorithms, pp. 124-132, Morgan Kaufmann Publishers, CA, 1989.

[8] David E. Goldberg, K. Deb, and B. Korb, "Do not Worry, be Messy," Proceedings of the Fourth International Conference on Genetic Algorithms, pp. 24-30, Morgan Kaufmann Publishers, CA, 1991.