A Genetic Algorithm for Solving Connected Dominating Set Problem in Wireless ADHOC Network

Vikas More  
2nd Year M.Tech.  
Dept of PG Studies  
VTU, Belgaum  
vikasmore08@gmail.com

Dr. S.R.Mangalwede  
Professor,  
Dept. of Information Science,  
G.I.T, Belgaum.  
mangalwede@yahoo.com

Abstract—In this paper we propose Genetic Algorithm for reducing connected dominating set in Wireless Adhoc networks. Virtual backbone will be formed to count connected dominating set. It shows that the connected dominating size is reduced in our virtual backbone. It solves the problem of flooding. We use concept of Unit Disk Graph from graph theory to solve our example. We finally prove that the Minimum Connected Dominating Set (MCDS) is NP-Complete. Our approach for solving this problem by considering weight for the vertices.

Keywords: Genetic Algorithm, Connected Dominating Set (CDS)

1. INTRODUCTION

Genetic Algorithms

1.1 Definition

A Genetic Algorithm (GA) is a stochastic search method which is inspired by natural biological evolution. A better and better approximation to a solution is given by GA which operates on a population of potential solutions applying the principle of survival of the fittest. At each generation, a new set of approximations is created by the process of selecting individuals according to their level of fitness in the problem domain and breeding them together using operators borrowed from natural genetics. This process leads to the evolution of populations of individuals that are better suited to their environment than the individuals that they were created from, just as in natural adaptation.

1.2 Steps

In the first step, individuals, or current approximations, are encoded as strings, called chromosomes, composed over some alphabet(s). In GA the representation used are binary, ternary, integer, real-valued. The most commonly used representation in GAs is the binary alphabet \{0, 1\}. Having decoded the chromosome representation into the decision variable domain, it is possible to assess the performance, or fitness, of individual members of a population. This is done through a fitness function that characterizes an individual’s performance in the problem domain. In the natural world, this would be an individual’s ability to survive in its present environment. Thus, the objective function establishes the basis for selection of pairs of individuals that will be mated together during reproduction.

During the reproduction phase, it calculates fitness value with fitness function and is assigned to individuals. This value is used in the selection to bias towards more fit individuals.

Highly fit individuals, relative to the whole population, have a high probability of being selected for mating whereas less fit individuals have a correspondingly low probability of being selected. Once the individuals have been assigned a fitness value, they can be chosen from the population, with a probability according to their relative fitness, and recombined to produce the next generation. Genetic operators manipulate the characters (genes) of the chromosomes directly, using the assumption that certain individual’s gene codes, on average, produce fitter individuals. The recombination operator is used to exchange genetic information between pairs, or larger groups, of individuals. The simplest recombination operator is that of single-point crossover.

Mutation is generally considered to be a background operator that ensures that the probability of searching a particular subspace of the problem space is never zero. This has the effect of tending to inhibit the possibility of converging to a local optimum, rather than the global optimum.

After recombination and mutation, the individual strings are then, if necessary, decoded, the fitness function evaluated, a fitness value assigned to each individual and individuals selected for mating according to their fitness, and so the process continues through subsequent generations. In this way, the average performance of individuals in a population is expected to increase, as good individuals are preserved and bred with one another and the less fit individuals die out. The GA is terminated when some criteria are satisfied, e.g. a certain number of generations, a mean deviation in the population, or when a particular point in the search space is encountered.

Unit Disk Graph:

We can use a simple graph \( G = (V; E) \) to represent an ad hoc wireless network, where \( V \) represents a set of wireless mobile hosts and \( E \) represents a set of edges. An edge between host pairs \( \{v, u\} \) indicates that both hosts \( v \) and \( u \) are within each other’s wireless transmitter ranges. To simplify our discussion, we assume all mobile hosts are homogeneous, i.e., their transmitter ranges are the same. Due to homogeneous the corresponding graph will be an undirected graph.
In this algorithm, the existing node of a graph to minimize or maximize some objective function. Presenting a GA for such problems is very easy and better.

Problem representation:
The objective of the GA algorithm presented here is just to solve a decision problem which is “is there any Power Aware Connected Dominating Set with size k?”. The chromosome used for this problem is simply a list of integers in the chromosome, and as mentioned before every chromosome has the same length k. Consider a graph with size in range [500...1000], we can start with a genetic algorithm with k=100 (Connected Dominating Set with size k). In this case the algorithm probably converges to an answer. (Since 100 is rather high as the size of CDS of a graph with size in range [500...1000]). Then we can try k=50 and run algorithm again. Probably algorithm fails in finding a CDS of size 50 (fail condition would be explained in the next parts). In this way we can use somehow binary search to finding the minimum value for k in which the algorithm converges to an answer. For finding power aware

III. PROPOSED A GENETIC ALGORITHM FOR MINIMUM CONNECTED DOMINATING SET PROBLEM

Genetic Algorithm has been very successful in Graph Theory. For example genetic algorithms are mostly used as good approximations of TSP. Also there are many graph problems where the search for the optimal solution involves obtaining a subset of the vertices of a graph to minimize or maximize some objective function. Presenting a GA for such problems is very easy and better.

Problem representation:
The objective of the GA algorithm presented here is just to solve a decision problem which is “is there any Power Aware Connected Dominating Set with size k?”.

The chromosome used for this problem is simply a list of vertices with size k. Consequently the size of all chromosomes would be k. Every chromosome is represented as a set of vertices, in the mathematical sense of a set. That is, there are no duplicated vertices and there is no ordering among the vertices. The vertices are represented as integers in the chromosome, and as mentioned before every chromosome has the same length k. Consider a graph with size in range [500...1000], we can start with a genetic algorithm with k=100 (Connected Dominating Set with size k). In this case the algorithm probably converges to an answer. (Since 100 is rather high as the size of CDS of a graph with size in range [500...1000]). Then we can try k=50 and run algorithm again. Probably algorithm fails in finding a CDS of size 50 (fail condition would be explained in the next parts). In this way we can use somehow binary search to finding the minimum value for k in which the algorithm converges to an answer. For finding power aware
MCDS also a new characteristic is applied to the chromosome which is Boolean value, it can be interooperated as the gender (male or female) of each chromosome. This would be very useful for generating better offsets during crossover operation. (This can be helpfulroller in power aware MCDS and not in classic MCDS).

Fitness Function

The fitness evaluation for the connected dominating set problem required to differentiate between any two chromosomes depends on the number of points that the subset covers, and also the number of points in the subset which are connected. This is because it is possible to find a solution that covers all of the points, but perhaps not all of the subset points are connected.

Therefore, the fitness function for this problem can be defined as:

\[ fit (c h ) = \alpha X ( c h ) + \beta Y ( c h ) \]

Where \( X \) is the number of points covered in the solution, and \( Y \) is the size of the maximum connected subset. Also \( \alpha \) and \( \beta \) are weight parameters that can be changed based on the importance of domination or connectivity. [It is suggested to have \( \alpha = .8 \) and \( \beta = .2 \)]

Crossover:

In each step of algorithm a random number of mutations are applied to the generation of chromosomes.

For each mutation operation two parent chromosomes are required. These parents are choosing randomly; however chromosomes with higher fitness have more chance of being chosen. After two parents (chromosomes) have been selected for crossover, the GA computes two exchange vectors, one for each parent, as follows:

\[ evP1 = P1 \cap P2 \cap P1 \]
\[ evP2 = P2 \cap P1 \cap P2 \]

For example if we have two parent \( p1 \) and \( p2 \) as:

\[ P1 = \{1, 2, 3, 4, 5, 6, 7\} \]
\[ P2 = \{2, 5, 7, 9, 10, 12, 20\} \]

The exchange vector would be:

\[ evP1 = \{1, 3, 4, 6\} \]
\[ evP2 = \{9, 10, 12, 20\} \]

Since all chromosomes have the same length (k), the size of exchange vectors would be the same too. Swapping a random number of elements in exchange vectors, results in exchange vectors of two offspring:

\[ evP1' = \{9, 20, 4, 10\} \]
\[ evP2' = \{1, 6, 12, 3\} \]

Then two children would be retrieved as:

\[ off1 = evP1 \cup ( p1 \cap p2 ) = \{9, 20, 4, 5, 10, 7\} \]
\[ off2 = evP2 \cup ( p1 \cap p2 ) = \{2, 5, 7, 1, 6, 12, 3\} \]

The expected value of the fitness of these twoVERTEXES is the same and one of them is selected randomly as the output of crossover.

Mutation:

As mentioned before, mutation is very important for escaping from the local optimums. A bad mutation technique like simple mutation in which a single vertex in one random chromosome is swapped by another random vertex is not effective in graphs with big size (>500). What we used for here is N4N mutation which is suggested for finding MCDS and some other problems in. This type of mutation is mainly based on Hyper mutation operator which is a classical mutation in graph GAs.

End condition:

While verifying the fitness of each new generation, if there exist a chromosome \( ch \) for which in Equation 1 we have \( X(ch) = n \) (the set is dominating) and \( Y(ch)=n \) (the set is connected), then we report this chromosome as the answer of the decision problem. It means we have got a subset of vertexes with length k which is dominating and connected. (the GA stops here). If we did not find such a chromosome after s generations (s steps of GA), then we can declare there is no answer for the decision problem, the GA stops and the fittest chromosomes of the last population (s’th population) would be reported as the approximate answers for the decision problem.

Algorithm:

Procedure N4N

Step1:
Randomly select a subset of 10% of the chromosomes from the entire population

Step2:
FOR EACH chromosome X selected in Step1
DO
FOR EACH node “i” included in set X
DO

BEST = X
Let H be the set (of up to four) of the neighbors of node i that are not currently present in chromosome X
FOR EACH node index “j” that is currently present in the set H
DO

Let Y be a new chromosome with the set of nodes given by: (X – {i}) U {j}
Calculate the fitness of Y
If fitness(Y)<fitness(BEST) then BEST = Y
END FOR
If fitness(BEST)<fitness(X) then
X = BEST
END FOR
Insert the new X into the population replacing the old X
END FOR

IV. RESULTS

Experiment No 1:

In this experiment we study the impact of applying genetic algorithm for calculating connected dominating set in wireless adhoc network. We received following result as shown in table using Genetic Algorithm.

<table>
<thead>
<tr>
<th>Sr. No</th>
<th>Number of Nodes</th>
<th>CDS Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>60</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>80</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>100</td>
<td>19</td>
</tr>
<tr>
<td>4</td>
<td>120</td>
<td>20</td>
</tr>
</tbody>
</table>

Experiment No 2:

In this experiment, we compare the results obtained from our proposed algorithm with the results of Butenko et al.’s algorithm [11], Li et al.’s algorithm [14], Xie et al.’s algorithm [15] and Misra’s algorithm [17]. In terms of
induced CDS size. From figure it is clear that the CDS size of genetic algorithm is low as compare to others.

Figure 2: Comparison of the backbone size for the CDS-based backbone formation algorithms.

V. CONCLUSION

In this paper, we presented a method based on genetic algorithms for solving the MCDS problem in wireless adhoc network. Most of the components of proposed GA are comparable to those used in a standard GA. We developed a heuristic called N4N heuristic for mutation operator. Experimental results show that the connected dominating set size is low as compare to Misra, Li, Butenko & Xie algorithms.

REFERENCES