

A Genetic Algorithm for Power Aware Minimum Connected Dominating Set Problem in Wireless Ad-Hoc Networks

Shahin Kamali
Vahid Safarnourollah

15 December 2006
Concordia University

Index:

1. Introduction
2. Wireless AD HOC Networks
 - 2.1. Example and Usage
3. Broadcasting
4. Unit Disk Graph
5. Flooding
 - 5.1. Problem of Flooding
6. Minimum Connected Dominating Set for our problem
7. DS, MDS and MCDS:
 - 7.1. Dominating Set (DS)
 - 7.2. Minimum Dominating Set (MDS)
 - 7.2.1. Proof of NP-Completeness
 - 7.3. Minimum Connected Dominating Set (MCDS)
8. Genetic Algorithm
 - 8.1. Definition
 - 8.2. Steps
 - 8.3. Applications
9. A Genetic Algorithm for Power Aware Minimum Connected Dominating Set
 - 9.1. Introduction
 - 9.2. Problem Representation
 - 9.3. Fitness Function
 - 9.4. Crossover

9.5. Mutation

9.6. End Condition

10. References

1 Introduction:

Our project is about finding Minimum Connected Dominating Set (MCDS) in Unit Disk Graph (the graph representation of the Wireless AD HOC networks) with weight for vertices by using Genetic Algorithm.

By Finding MCDS in the Wireless AD HOC networks, we can make a virtual backbone infrastructure to solve our problem of broadcasting in these networks.

Through this report we first discuss more about the Wireless AD HOC network and they way that we represent it as Unit Disk Graph. Then the problem of broadcasting and the premier way of doing that by flooding will be explained and also the problems that would be made by this approach will be mentioned.

Afterward, the Concepts of Dominating Set, Minimum Dominating Set (with its NP-Completeness proof) and Minimum Connected Dominating Set will be discussed in detail.

Next, Genetic Algorithm, simple example of using this algorithm and what has been done so far for solving Minimum Connected Dominating Set by using Genetic Algorithm will be explained.

Finally, our approach for solving this problem by considering weight for the vertices is discussed in detail. This approach is an extension of on a Genetic Algorithm represented in [17] for finding Minimum Connected Dominating Set. We have modified Fitness Function and also Crossover operator to present a power aware Genetic Algorithm.

2 Wireless AD HOC Network:

An ad hoc wireless network is a special type of wireless network in which a collection of mobile hosts with wireless network interfaces may form a network on a temporary basis and unlike wired and cellular networks has no physical backbone infrastructure. If only two hosts, located closely together within each other's wireless transmission range, are involved in the ad hoc wireless network, no real routing protocol or decision is necessary. However, if two hosts that want to communicate are outside each other's wireless transmission ranges, they could communicate only if other hosts between them in the ad hoc wireless network are willing to forward packets for them.

2.1 Example and usage:

Wireless ad hoc networks can be flexibly and quickly deployed for many applications such as automated battlefield, search and rescue, and disaster relief. Unlike wired networks or cellular networks, no wired backbone infrastructure is installed in wireless ad hoc networks. In this paper, we assume that all nodes in a wireless ad hoc network are distributed in a two-dimensional plane and have an equal maximum transmission range of one unit.

Overview:

1. Limitations
 - a. No physical backbone infrastructure like wired network or cellular networks
 - b. Limited wireless bandwidth
 - c. Limited battery power
 - d. Multi-hop routing

2. Challenge
 - a. Mobility
 - b. Scalability
 - c. Power
 - i. Minimizing power consumption during the idle time
 - ii. Minimizing power consumption during communication
 - d. QOS
 - i. End to End delay
 - ii. Bandwidth management
 - iii. Probability of packet loss
3. Operation
 - a. Broadcasting
 - b. Routing
 - c. Multicasting

3 Broadcasting:

Broadcasting is a fundamental networking operation in wireless ad hoc networks. It is widely and frequently performed in many networking tasks such as paging a particular host, sending an alarm signal, and finding a route to a particular host [1,2,3]. A simple broadcasting mechanism, known as *flooding*, is to let every node retransmit the message to all its 1-hop neighbors when receiving the first copy of the message.

Overview:

1. Function:
 - a. paging a particular host
 - b. sending an alarm signal
 - c. finding a route to a particular host
2. Objective:
 - a. Reliability
 - i. (All nodes have received the broadcast packet)
 - b. Optimization

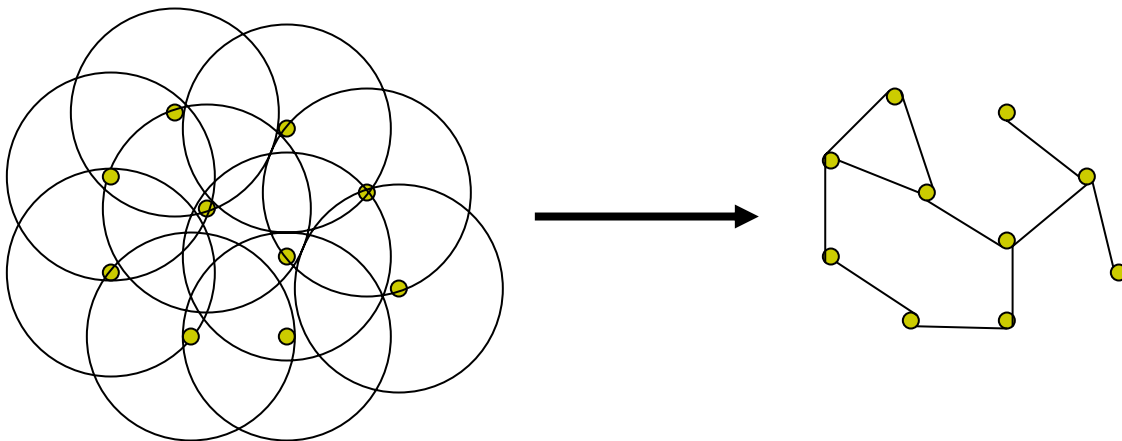
4 Unit Disk Graph:

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. Thus the corresponding graph will be an undirected graph [4].

Overview:

1. Unit Disk Graph

- a. All mobile hosts are homogeneous
 - i. The same transmission range
 - ii. Unidirectional link
- b. Vertices with weights (Remaining Power)



Example of the Unit Disk Graph

5 Flooding:

Flooding, or called blind flooding, was first discussed in [5, 6], where every node in the network retransmits the flooding message when it is its first time to receive it. This simple scheme guarantees that a flooding message can reach all nodes if there is no collision and the network is connected. However, it generates excessive amount of redundant network traffic, because all nodes in the network transmit the flooding message. This will consume a lot of energy resource of mobile nodes and cause the congestion of the network. Furthermore, due to the broadcast nature of radio transmissions, there is a very high probability of signal collisions when all nodes flood the message in the network at the same time, which would cause more re-transmissions or some nodes failing to receive the message. It is so called the broadcast storm problem [7]. Sinha et al claimed that “in moderately sparse graphs the expected number of nodes in the network that will receive a broadcast message was shown to be as low as 80%” in [8].

5.1 Problem of Flooding:

The problem addressed is the using flooding to propagate a broadcast message throughout a network. The “broadcast storm problem” refers to the problem associated with flooding. First flooding results in a large number of duplicate packets being sent in the network. Second, a high amount of contention will take place, because nodes in close proximate of each other will try to rebroadcast the

message. Third, collisions are likely to occur because the RTS/CTS are not applicable for broadcast messages.

6 Minimum Connected Dominating Set for our problem:

Dominating-set-based routing [11,12] is based on the concept of dominating set in graph theory [12]. This concept of routing is valid only for networks, which can be represented by connected graphs. So, in here we will consider only connected graphs. Main advantage of this approach is that searching space for a route is reduced to the nodes in the dominating set. As long as changes in network topology do not affect this sub network there is no need to re-calculate routing tables.

In ad hoc wireless networks, the limitation of power of each host poses a unique challenge for power-aware design [9]. There has been an increasing focus on low cost and reduced node power consumption in ad hoc wireless networks. Unfortunately, nodes in the dominating set in general consume more energy, since they are involved almost in every routing and broadcasting task. A consequence of this observation is that if the selection of the nodes in the dominating set remains fixed forever, these nodes will soon fall short of power and as a result the network will fail. That is why we use dynamic selection schemes, in which the dominating set is changed according to the energy level of all nodes in the network. In this way, we will try to improve lifespan of the whole network.

So in the following we have the definition of DS, MDS and MCDS and the rest will be how to solve this problem with genetic algorithm.

7 DS, MDS and MCDS:

7.1 Dominating Set (DS):

In Graph theory, a dominating set for a graph $G = (V, E)$ is a subset V' of V such that every vertex not in V' is joined to at least one member of V' by some edge.

7.2 Minimum Dominating Set (MDS):

The dominating set problem is defined as minimum Dominating Set that is an NP-complete problem in graph theory. The problem is to determine whether there is a dominating set of size K or less for G . In other words, we want to know if there is a subset D of V of size less than or equal to K such that every vertex not in D is joined to at least one member of D by an edge in E .

The optimization version of the problem, that is finding the smallest $|V'|$ such that V' is a dominating set, has the approximation algorithm. To be more precise, it has approximation algorithm within a factor of $1 + \log |V|$, but can not be within $c \log |V|$ for some $c > 0$.

7.2.1 Proof of NP-completeness:

The dominating set problem has been proven to be NP-complete by a reduction from the vertex cover problem [13,14].

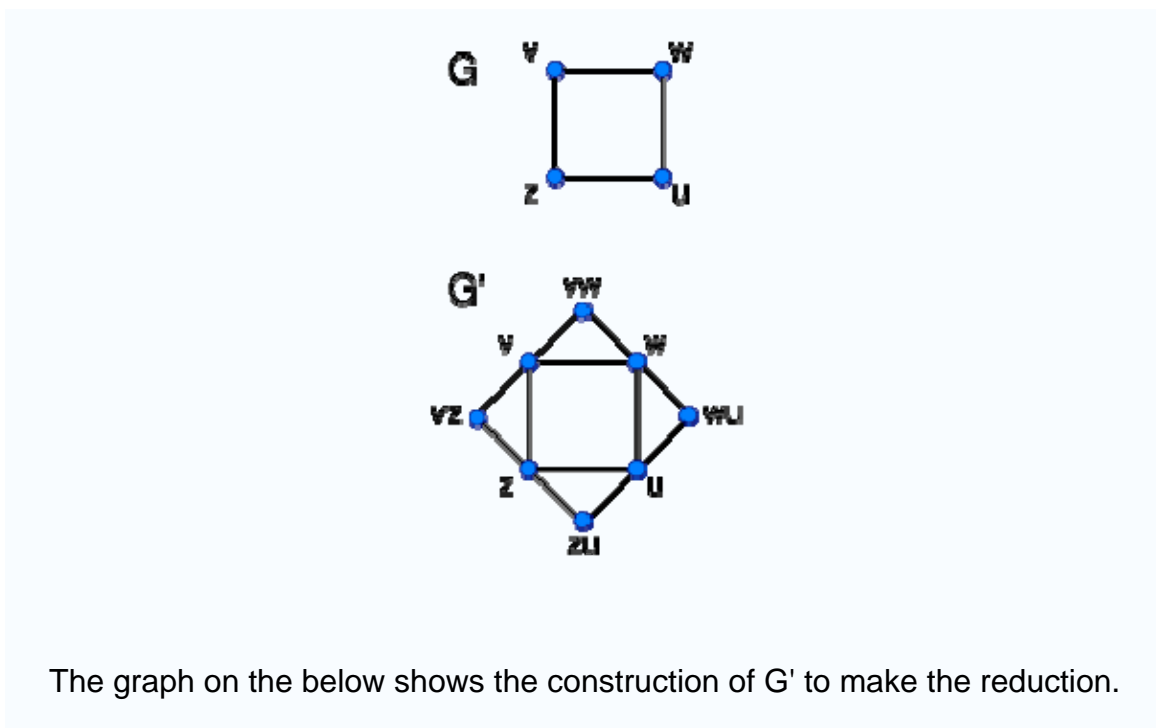
Vertex cover and dominating set has the same problem format; the difference is that a dominating set covers vertices, while a vertex cover covers edges. So, find a way to build a graph using vertices to represent the edges from the original

graph. Let's show how to build the graph to make the reduction from vertex cover to dominating set: Let $\langle G, k \rangle$ be an instance of the vertex cover problem. Build a new graph G' adding new vertices and edges to the graph G . Specifically, for each edge $\langle v, w \rangle$ of G , add a vertex vw and the edges $\langle v, vw \rangle$ and $\langle w, vw \rangle$. The new graph obtained is denoted by G' .

Now, the proof: G' has a dominating set D of size k if and only if G has a vertex cover C of size k .

(\Rightarrow) D is a dominating set of size k in G' . So, every edge hits some vertex in D . D is a vertex cover in G of size k .

(\Leftarrow) C is a vertex cover in G with size k , so new and old vertices dominated by k vertices.

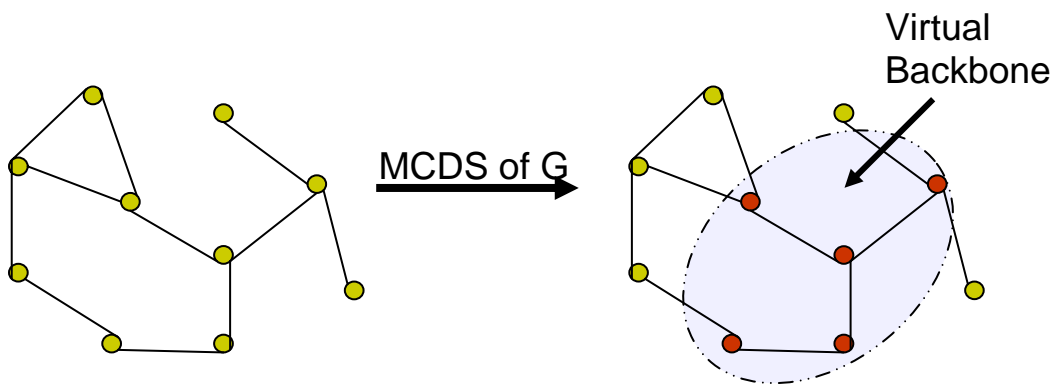


7.3 Minimum Connected Dominating Set (MCDS):

Connected Dominating Set (CDS) is a dominating set which is also a connected sub graph of the original graph G .

Minimum connected dominating set is a connected dominating set such that removal of any node from that set makes it a Non-connected dominating set.

This is also NP-complete problem



Example of the Minimum Connected Dominating Set that provides the virtual backbone structure for our network

8 Genetic Algorithms

8.1 Definition

A Genetic Algorithm (GA) is a stochastic search method which is inspired by natural biological evolution. A GA operates on a population of potential solutions applying the principle of survival of the fittest to produce (hopefully) better and better approximations to a solution. 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.

8.2 Steps

In the first step, individuals, or current approximations, are encoded as strings, called *chromosomes*, composed over some alphabet(s). The most commonly used representation in GAs is the binary alphabet {0, 1}, although other representations can be used, e.g. ternary, integer, real-valued etc.

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, each individual is assigned a fitness value given by the fitness function. 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.

Consider the two parent binary strings:

P1 = 1 0 0 1 0 1 1 0

P2 = 1 0 1 1 1 0 0 0

If an integer position, i , is selected uniformly at random between 1 and the string length, l , minus one $[1, l-1]$, and the genetic information exchanged between the individuals about this point, then two new offspring strings are produced. The two offspring below are produced when the crossover point $i = 5$ is selected,

O1 = 1 0 0 1 0 0 0 0

O2 = 1 0 1 1 1 1 1 0

This crossover operation is not necessarily performed on all strings in the population. Instead, it is applied with a probability P_x when the pairs are chosen for breeding. A further genetic operator, called mutation, is then applied to the new chromosomes, again with a set probability, P_m . Mutation causes the individual genetic representation to be changed according to some probabilistic rule. In the binary string representation, mutation will cause a single bit to change its state,

0 → 1 or 1 → 0. So, for example, mutating the fourth bit of O1 leads to the new string,

O1_m = 1 0 0 0 0 0 0 0

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.

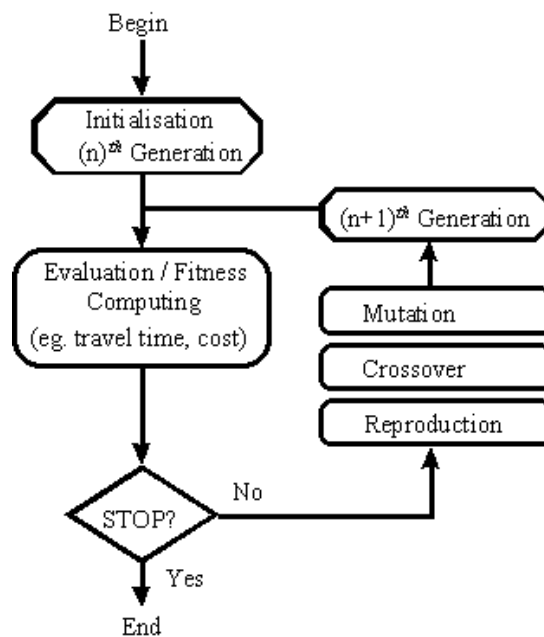


Figure 1 Genetic Algorithms' function¹

8.3 Applications of Genetic Algorithms:

¹ <http://www.its.leeds.ac.uk/projects/smartest/d3f5p9.gif>

Nearly everyone can gain benefits from Genetic Algorithms, once he can encode solutions of a given problem to chromosomes in GA, and compare the fitness of solutions.

Followings are the most common applications of GAs:

When ...

- The search space is large, complex or poorly understood
- Domain knowledge is scarce or expert knowledge is difficult to encode to narrow the search space.
- No mathematical analysis is available.
- Traditional search methods fail.²

NP_Complete problems are instances of the last case in which a traditional search method takes exponential time (consequently fails)

8 A Genetic Algorithm for Power Aware Minimum Connected Dominating Set Problem:

9.1 Introduction

GAs has been very successful in Graph Theory. For example genetic algorithms are widely used as good approximations of TSP. Also there are numerous 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 simple and effective. MCDS and also

² http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/tcw2/report.html#WhoGain

Power Aware MCDS (which is the main point here) are samples of these problems.

We represent a GA for MCDS and also Power Aware MCDS we start with defining a fitness function:

Problem representation:

The goal 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). Each chromosome is interpreted 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 vertexes 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 some how 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 a Boolean value, it can be interoperated as the gender (male or female) of each chromosome. This would be very useful for generating better offsets during crossover operation. (This can be helpful just in power aware MCDS and not in classic MCDS)

8.3 Fitness Function

The fitness evaluation for the connected dominating set problem needs to differentiate between any two chromosomes based 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:

Equation 1

$$fit(ch) = \alpha \times X(ch) + \beta \times Y(ch)$$

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

The fitness function of equation 1 does not consider the weight of each vertex (battery of each node). So it can not be applied for Power Aware MCDS and a new fitness function is needed. This function is defined in equation 2:

Equation 2

$$fitPAware(ch) = (\mu(batt(v \in ch))^2 / \text{var}(batt(v \in ch))) \times fit(ch)$$

In which $fit(ch)$ has the same definition of equation 1.

Also in power aware system we prefer the answers (chromosomes) which contain nodes (vertexes) with higher battery (since they are going to be the backbone of broadcasting). As a result the factor $(\mu(batt(v \in ch))^2 / \text{var}(batt(v \in ch)))$ is multiplied to the initial formula. As it shows the answers (chromosomes) with higher expected values of battery and lower variance are preferred. Variance is also considered since we do not like answers in which one vertex has a high battery while the other one has a low battery, Hence the light vertex (low battery) can disconnect broadcasting after some little steps (it gets out of battery), In this case the remained battery of the high battery vertex would be useless.

The fitness function is not different for male or female chromosomes at all. (This is necessary for having the same number of males and females in different generations)

8.4 Crossover:

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

Non-power aware:

For each mutation operation two paren chromosomes are needed. These parents are selected randomly; however chromosomes with higher fitness have more chance of being selected.

After two parents (chromosomes) have been selected for crossover, the GA computes two exchange vectors, one for each parent, as follows:

$$evP1 = P1 - P1 \cap P2$$

$$evP2 = P2 - P1 \cap P2$$

For example if we have two parent p_1 and p_2 as:

$$p1 = \{1,2,3,4,5,6,7\}$$

$$p2 = \{2,5,7,9,10,12,20\}$$

The exchange vector would be:

$$evP1 = \{1,3,4,6\}$$


$$evP2 = \{9,10,12,20\}$$


Figure 2 swapping elements of exchange vectors

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:

$$evP1' = \{9,20,4,10\}$$

$$evP2' = \{1,6,12,3\}$$

Figure 3 exchange vectors after swapping

Then two children would be retrieved as:

$$off1 = evP1' \cup (p1 \cap p2) = \{9,2,20,4,5,10,7\}$$

$$off2 = evP2' \cup (p1 \cap p2) = \{2,5,7,1,6,12,3\}$$

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

Power aware:³

Crossover in power aware GA is a little different. First we need to select two parents which are not from the same gender. Also a probability for being swapped is defined each vertex based on its remaining battery. In male chromosomes the vertexes with higher battery would have more probability of being swapped, while in female chromosomes the vertexes with lower battery have more chance of being swapped. As a result after swapping one of the exchange vectors contains high battery vertexes with higher fitness. This one would be selected to construct the final output of crossover.

Also a gender is randomly assigned to the new offset. (To have generations with fixed percentage of male and female chromosomes)

8.5 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 replaced by another random vertex is not effective in graphs with large size (>500).

³ The crossover method suggested here is not implemented yet and we are not sure if it helps to increase convergence rate of algorithm or not.

What we used for here is N4N mutation which is suggested for finding MCDS and some other problems in [17]. This type of mutation is mainly based on Hypermutation operator which is a classical mutation in graph GAs.

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: $(X - \{i\}) \cup \{j\}$

Calculate the fitness of Y

If $\text{fitness}(Y) < \text{fitness}(\text{BEST})$ then $\text{BEST} = Y$

END FOR

if $\text{fitness}(\text{BEST}) < \text{fitness}(X)$ then $X = \text{BEST}$

END FOR

Insert the new X into the population replacing the old X

END FOR

We have not applied any modification for power aware genetic algorithm for mutation yet.

8-6 End condition:

While checking 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 found 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 deduce 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.

References:

- [1] J. Broch, D.B. Johnson and D.A. Maltz, The Dynamic Source Routing Protocol for mobile ad hoc networks, IETF Internet Draft, draft-ietfmanet-dsr-05.txt (March 2001).
- [2] Z.J. Haas, M.R. Pearlman and P. Samar, The Interzone Routing Protocol (IERP) for ad hoc networks, IETF Internet Draft, draft-ietf-manetzone-ierp-00.txt (January 2001).
- [3] C.E. Perkins, E.M. Royer and S. Das, Ad Hoc On Demand Distance Vector (AODV) routing, IETF Internet Draft, draft-ietf-manet-aodv-08.txt (March 2001).
- [4] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Commun., 1999, pp. 7–14.
- [5] C. Ho, K. Obraczka, G. Tsudik, and K. Viswanath, "Flooding for Reliable Multicast in Multi-hop Ad Hoc Networks," in Proc. of the Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Communication, 1999, pp. 64–71.
- [6] J. Jetcheva, Y. Hu, D. Maltz, and D. Johnson, "A Simple Protocol for Multicast and Broadcast in Mobile Ad Hoc Networks," Internet Draft: draft-ietf-manet-simple-mbcast-01.txt, July 2001.
- [7] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, "The broadcast storm problem in a mobile ad hoc network," Proc. of ACM/IEEE MOBICOM'99, pp. 151-162, Aug. 1999.

- [8] P. Sinha, R. Sivakumar and V. Bharghavan, "Enhancing ad hoc routing with dynamic virtual infrastructures", IEEE INFOCOM 2001, pp. 1763-1772.
- [9] J.Wu, F. Dai, M. Gao, and I. Stojmenovic, "On Calculating Power-Aware Connected Dominating Sets for Efficient Routing in Ad Hoc Wireless Networks", IEEE International Conference, Parallel Processing ,Sept.2001.
- [10] S. Butenko, X. Cheng, C.A.S Oliveira, and P.M. Pardalos, "A New Heuristic for the Minimum Connected Dominating Set Problem on Ad Hoc Wireless Network", to appear in *Cooperative Control and Optimization*, Kluwer Academic Publisher, pp. 61-73, 2004.
- [11] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in Proc. of the 3rd Int'l Workshop on Discrete Algorithms and Methods for Mobile Computing and Commun., 1999, pp. 7-14.
- [12] B. Das and V. Bhargavan. Routing in ad-hoc networks using minimum connected dominating sets. IEEE International Conference on Communications (ICC '97), June 1997.
- [13]Michael R. Garey and David S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman. ISBN 0-7167-1045-5. A1.1: GT2, pg.190.
- [14]Mitchell, S., and S. Hedetniemi [1977], "Edge domination in trees", Proceedings of the 8th Southeastern Conference on Combinatorics, Graph Theory, and Computing, Utilitas Mathematica Publishing, Winnipeg, 489-509.

- [15] GRUIA CA˘ LINESCU, ION I. MA˘ NDOIU, PENG-JUN WAN and ALEXANDER Z. ZELIKOVSKY, Selecting Forwarding Neighbors in Wireless Ad Hoc Networks, Mobile Networks and Applications 9, 101–111, 2004
- [16] Tamaghna Acharya and Rajarshi Roy, Distributed Algorithm for Power Aware Minimum Connected Dominating Set for Routing in Wireless Ad Hoc Network, Proceedings of the 2005 International Conference on Parallel Processing Workshops (ICPPW'05) 2005
- [17] Yaser Alkhalifah, Roger L. Wainwright “A Genetic Algorithm Applied to Graph Problems Involving Subsets of Vertices” IEEE International Conference, IEEE Congress on Evolutionary Computation 2004 (CEC'04).
- [18] Eric Krevise Prebys. The Genetic Algorithm in Computer Science.

