A GENETIC ALGORITHM FOR POWER AWARE MINIMUM CONNECTED DOMINATING SET PROBLEM ON WIRELESS AD HOC NETWORKS

COMP 6651 Project

Shahin Kamali 6023614
Vahid Safarnourollah 5671337

November 2006
Outline

• Wireless AD HOC Network
• Broadcasting Problem in Wireless ADHOC Networks
• Unit Disk Graph
• DS, MDS and MCDS
• Genetic Algorithm’s Function
• A Simple Example of GA
• A GA for MCDS
• A GA for Power Aware MCDS
Wireless ADHOC Network

• Definition
  Example: Automated battlefield, search and rescue, and disaster relief

• Limitations
  – No physical backbone infrastructure like wired network or cellular networks
  – Limited wireless bandwidth
  – Limited battery power
  – Multi-hop routing
Wireless ADHOC Networks (Cont’d)

• Challenge
  – Scalability
  – Power
    • Minimizing power consumption during the idle time
    • Minimizing power consumption during communication

• Operation
  – Broadcasting
  – Routing
  – Multicasting
Broadcasting

• Function:
  – paging a particular host
  – sending an alarm signal
  – finding a route to a particular host

• Objective:
  – Reliability
    (all nodes have received the broadcast packet)
  – Optimization
Broadcasting (Cont’d)

• Simple mechanism:
  – Flooding: Each node retransmits the message to its 1-hop neighbors

Drawbacks: redundant transmission, collisions, contention and inefficient (Limitation of Node’s Power)
Wireless ADHOC Network’s Graph Representation

• Unit Disk Graph
  • All mobile hosts are homogeneous
    – The same transmission range
    – Unidirectional link

-Vertices with weights (Remaining Power)-
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.

\[ DS \text{ of } G \]
Minimum Dominating Set (MDS)

- The optimization version of the algorithm, that is finding the smallest $|V'|$ such that $V'$ is a dominating set, has approximation algorithm.

- The dominating set problem is NP-complete. Can be proved by reduction from the Vertex Cover problem

  its approximation is within a factor of $1 + \log |V|$, but it doesn’t have approximation within $c\log |V|$ for some $c > 0$
Minimum Connected Dominating Set (MCDS)

- **Connected Dominating Set (CDS)** is a dominating set which is also a connected subgraph 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.
Genetic Algorithms (GAs)

• A particular class of evolutionary algorithms

• Use techniques inspired by biology such as inheritance, mutation, selection, and crossover (recombination)

• Used for finding true or approximate solutions to:
  – Optimization Problems
  – Search Problems (huge size)
GA’s function

• Trying to evolve a population of candidate solutions (chromosomes) to get “better and better” population.

• Candidate solutions (Chromosome)
  – Potential answers of the problem.
  – Initially generated randomly.

• What is a good chromosome?
  – A “Fitness Function” which is a criteria for evaluating chromosomes (potential answer) is defined.
  – “Evolution” means making the populations with higher fitness.
GA’s function (cont’d)

• To “evolve population” two operations are defined:
  – Crossover
    • Two chromosomes are randomly selected as parents and are combined to compose a new chromosome (offspring).
  – Mutation
    • A single chromosome is selected and a random change is applied to it to generate a new chromosome.
  – The size of population
    • a fixed number
    • After each stage the chromosomes with lower fitness are removed from population (a new Generation is created)

• Termination condition
  – Having a chromosome (solution) with a large enough fitness
  Or
  – The number of stages passes a certain number.
GA Procedure (Review)

• Initialising
  – start with a randomly generated population of chromosomes (candidate solutions).

• Evolution process
  – crossover (applied in a random number of times)
  – mutation (applied in a random number of times)
  – removing bad chromosomes (low fitness) to have a new generation

• Termination
  – Maintaining a good solution
  – After n generation
A simple application of GA

- Light colors shows the higher pixels
- The goal is to find the highest pixel (lightest color)
- Not very easy!
• A population of random points (chromosomes) are generated as the initial population.
Crossover

- Two chromosomes (points) are selected randomly (parents)
- The offspring (child) chromosome is a point in the rectangle bounded by parents
Removing

- Fitness of each chromosome (point) is its height.
- Chromosomes (points) with lower height (fitness) are removed.
A GA for MCDS

• **Representation**
  
  – Each chromosome is a simply a “Set” of vertices

  • No Duplicate vertices!
  
  • No ordering among vertices.
A GA for MCDS

• **Fitness Function (Not Power Aware)**
  - for each chromosome $ch$ we define

  $$fit(ch) = 0.8 \times X(ch) + 0.2 \times Y(ch)$$

  • $X(ch)$ is the number of vertices covered by $ch$ (contained or dominated by $ch$).

  • $Y(ch)$ is the max size of connected components of $ch$. 

  **Example**
  
  - $ch1 = \{3, 4, 7, 8, 9\}$
    - $X = 10$, $Y = 2$
    - $fit(ch1) = 8.4$

  - $ch2 = \{3, 5, 6, 7, 9\}$
    - $X = 10$, $Y = 4$
    - $fit(ch2) = 8.8$
A GA for MCDS

• **Fitness function for Power Aware MCDS**
  
  – Each vertex has a weight (battery)
  
  – We prefer the vertices which have higher battery.
  
  – Remember fitness function without power was:
    \[
    \text{fit}(ch) = 0.8 \times X + 0.2 \times Y
    \]
  
  • A fitness function with power applied:
    \[
    \text{fitP1}(ch) = \sum_{v_i \in ch} \text{batt}(v_i) \times \text{fit}(ch)
    \]
  
  • A better fitness function with power
    \[
    \text{fitP2}(ch) = \left( \mu(\text{batt}(v \in ch))^2 \right. \left/ \text{var}(\text{batt}(v \in ch)) \right) \times \text{fit}(ch)
    \]
A GA for MCDS (Evolution)

• Crossover

  – Select two chromosomes $p_1$, $p_2$ randomly as parents.
  
  – Define Exchange Vectors
    
    \[
    evP_1 = P_1 - P_1 \cap P_2 \\
    evP_2 = P_2 - P_1 \cap P_2
    \]
  
  – generate random number $r$
  
  – swap $r$ vertices of $evP_1$ with $evP_2$.  
    (randomly select vertices)

\[
\begin{align*}
G(23,180) & \\
p_1 & = \{1,2,3,4,5,6,7\} \\
p_2 & = \{2,5,7,9,10,12,20\} \\
evp_1 & = \{1,3,4,6\} \\
evp_2 & = \{9,10,12,20\} \\
r & = 3 \rightarrow \{1 \leq r \leq 4\} \\
evp_1' & = \{9,20,4,10\} \\
evp_2' & = \{1,6,12,3\} \\
off_1 & = evp_1' \cup (p_1 \cap p_2) = \{9,2,20,4,5,10,7\} \\
off_2 & = evp_2' \cup (p_1 \cap p_2) = \{2,5,7,1,6,12,3\}
\end{align*}
\]
A GA for MCDS

• **Mutation**
  
  – Lots of techniques
  
  – “Simple” Mutation
    
    • Select a chromosome randomly.
    
    • A random vertex of that chromosome is replaced by another random vertex
    
    • Ex:
      
      – ch1 = \{2,5,6,3,15,16,12\}

      – ch2 = \{2,14,6,3,15,16,12\}. 
A GA for MCDS

• **Termination**

  – The algorithm terminates if
    • A chromosome represents a solution (for the decision problem)
    • A connected subset of vertices that dominate all vertices \((X=n)\) is find.
  – Or if
    • GA finds no solution after a particular number of generations.
    • Probably there is no solution for decision problem
    • There are approximate results.
Thanks for your patience

Any Questions ?