Chapter 7

CLUSTERING

Mobile Computing
Summer 2002

Overview

- Motivation
- Dominating Set
- Connected Dominating Set
- The “Tree Growing” Algorithm
- The “Marking” Algorithm
- An algorithm for the unit disk graph

Clustering (Trick 7 revisited)

• Situations where many mobile nodes are close-by. In other words, in situations where it is usually the case that two neighbors are also neighboring. Example: conferences or this classroom.

• Graph to the right has \( \text{diameter}^* = 2 \). But what happens when we do flooding (for a first routing step, or a broadcast)? There will be much more than 2 transmissions.

*\( \text{diameter}^* \) = longest shortest path

Backbone

• Idea: Some nodes become backbone nodes (gateways). Each node can access and be accessed by at least one backbone node.

• Routing:
  1. If source is not a gateway, transmit message to gateway
  2. Gateway acts as proxy source and routes message on backbone to gateway of destination.
  3. Transmission gateway to destination.
(Connected) Dominating Set

- A **Dominating Set** DS is a subset of nodes such that each node is either in DS or has a neighbor in DS.

- A **Connected Dominating Set** CDS is a connected DS, that is, there is a path between any two nodes in CDS that only uses nodes that are in CDS.

- A CDS is a good choice for a backbone.

- It might be favorable to have few nodes in the CDS. This is known as the Minimum CDS problem.

An MCDS Algorithm

- Input: We are given an undirected graph. The nodes in the graph are the mobile stations; there is an edge between two nodes if the nodes are within transmission range of each other.

- Note that the graph is undirected, thus transmission is symmetric. Also note that the graph is not Euclidean.

- Output: Find a Minimum Connected Dominating Set, that is, a CDS with a minimum number of nodes.

- Problem: MCDS is NP-hard.

- Solution: Can we find a CDS that is “close” to minimum?

The “too simple tree growing” algorithm

- Idea: Start with the root and then greedily choose a neighbor of the tree that dominates as many new nodes as possible.

- Black nodes are in the CDS
- Grey nodes are neighbors of nodes in the CDS
- White nodes are not yet dominated, initially all nodes are white.

- Start: Choose a node of maximum degree, and make it the root of the CDS, that is, color it black (and its white neighbors grey).

- Step: Choose a grey node with maximum number of white neighbors and color it black (and its white neighbors grey).

Example of the “too simple tree growing” algorithm

Graph with $2n+2$ nodes; tree growing: $|\text{CDS}|=n+2$; Minimum $|\text{CDS}|=4$
Tree Growing Algorithm

- Idea: Don’t scan one but two nodes!
- Alternative step: Choose a grey node and its white neighbor node with a maximum sum of white neighbors and color both black (and their white neighbors grey).

Analysis of the tree growing algorithm

- Theorem: The tree growing algorithm finds a connected set of size $|\text{CDS}| \leq 2(1+H(\Delta)) \cdot |\text{DS}_{\text{OPT}}|$.
- $\text{DS}_{\text{OPT}}$ is a (not connected) minimum dominating set
- $\Delta$ is the maximum node degree in the graph
- $H$ is the harmonic function with $H(n) \approx \log(n)+0.7$
- In other words, the connected dominating set of the tree growing algorithm is at most a $O(\log(\Delta))$ factor worse than an optimum minimum dominating set (which is NP-hard to compute).
- With a lower bound argument (reduction to set cover) one can show that a better approximation factor is impossible, unless $P=NP$.

Proof Sketch

- The proof is done with amortized analysis.
- Let $S_u$ be the set of nodes dominated by $u \in \text{DS}_{\text{OPT}}$, or $u$ itself. If a node is dominated by more than one node, we put it in one of the sets.
- We charge the nodes in the graph for each node we color black. In particular we charge all the newly colored grey nodes. Since we color a node grey at most once, it is charged at most once.
- We show that the total charge on the vertices in an $S_u$ is at most $2(1+H(\Delta))$, for any $u$.

Charge on $S_u$

- Initially $|S_u| = u_0$.
- Whenever we color some nodes of $S_u$, we call this a step.
- The number of white nodes in $S_u$ after step $i$ is $u_i$.
- After step $k$ there are no more white nodes in $S_u$.
- In the first step $u_0 - u_1$ nodes are colored (grey or black). Each vertex gets a charge of at most $2/(u_0 - u_1)$.
- After the first step, node $u$ becomes eligible to be colored (as part of a pair with one of the grey nodes in $S_u$). If $u$ is not chosen in step $i$ (with a potential to paint $u_i$ nodes grey), then we have found a better (pair of) node(s). That is, the charge to any of the new grey nodes in step $i$ in $S_u$ is at most $2/u_i$. 
Adding up the charges in $S_u$

$$C \leq \frac{2}{u_0 - u_1}(u_0 - u_1) + \sum_{i=1}^{k-1} \frac{2}{u_i}(u_i - u_{i+1})$$

$$= 2 + 2\sum_{i=1}^{k-1} \frac{u_i - u_{i+1}}{u_i}$$

$$\leq 2 + 2\sum_{i=1}^{k-1} \left(H(u_i) - H(u_{i+1})\right)$$

$$= 2 + 2(H(u_1) - H(u_k)) = 2(1 + H(u_1)) \leq 2(1 + H(\Delta))$$

Discussion of the tree growing algorithm

- We have an extremely simple algorithm that is asymptotically optimal unless $P=NP$. And even the constants are small.

- Are we happy?

- Not really. How do we implement this algorithm in a real mobile network? How do we figure out where the best grey/white pair of nodes is? How slow is this algorithm in a distributed setting?

- We need a fully distributed algorithm. Nodes should only consider local information.

---

The Marking Algorithm

- Idea: The connected dominating set CDS consists of the nodes that have two neighbors that are not neighboring.

1. Each node $u$ compiles the set of neighbors $N(u)$
2. Each node $u$ transmits $N(u)$ and receives $N(v)$ from all its neighbors
3. If node $u$ has two neighbors $v, w$ and $w$ is not in $N(v)$ (and since the graph is undirected $v$ is not in $N(w)$), then $u$ marks itself being in the set CDS.

- Completely local; only exchange $N(u)$ with all neighbors
- Each node sends only 1 message, and receives at most $\Delta$
- Messages have size $O(\Delta)$

- Is the marking algorithm really producing a connected dominating set? How good is the set?

---

Example for the Marking Algorithm

- We have an extremely simple algorithm that is asymptotically optimal unless $P=NP$. And even the constants are small.

- Are we happy?

- Not really. How do we implement this algorithm in a real mobile network? How do we figure out where the best grey/white pair of nodes is? How slow is this algorithm in a distributed setting?

- We need a fully distributed algorithm. Nodes should only consider local information.
Correctness of Marking Algorithm

- We assume that the input graph G is connected but not complete.

- Note: If G was complete then constructing a CDS would not make sense. Note that in a complete graph no node would be marked.

- We show:
  The set of marked nodes CDS is
  a) a dominating set
  b) connected
  c) a shortest path in G between two nodes of the CDS is in CDS

Proof of a) dominating set

- Proof: Assume for the sake of contradiction that node u is a node that is not in the dominating set, and also not dominated. Since no neighbor of u is in the dominating set, the nodes N'(u) := u ∪ N(u) form:
  a complete graph
  - if there are two nodes in N(u) that are not connected, u must be in the dominating set by definition
  no node v ∈ N(u) has a neighbor outside N(u)
  - or, also by definition, the node v is in the dominating set

- Since the graph G is connected it only consists of the complete graph N'(u). We precluded this in the assumptions, therefore we have a contradiction

Proof of b) connected, c) shortest path in CDS

- Proof: Let p be any shortest path between the two nodes u and v, with u,v ∈ CDS.

- Assume for the sake of contradiction that there is a node w on this shortest path that is not in the connected dominating set.

- Then the two neighbors of w must be connected, which gives us a shorter path. This is a contradiction.

Improving the Marker Algorithm

- We give each node u a unique id(u).

- Rule 1: If N'(v) ⊆ N'(u) and id(v) < id(u), then do not include node v into the CDS.

- Rule 2: Let u,w ∈ N(v). If N(v) ⊆ N(u) ∪ N(w) and id(v) < id(u) and id(v) < id(w), then do not include v into the CDS.

  (Rule 2+: You can do the same with more than 2 covering neighbors, but it gets a little more intricate.)

  …for a quiet minute: Why are the identifiers necessary?
Example for improved Marking Algorithm

- Node 17 is removed with rule 1
- Node 8 is removed with rule 2

Quality of the Marking Algorithm

- Given a Euclidean chain of $n$ homogeneous nodes
- The transmission range of each node is such that it is connected to the $k$ left and right neighbors, the id’s of the nodes are ascending.

Example

- This gives a dominating set. But it is not connected.
Euclidean Unit Disk Graph Continued

3. If a node $w$ is dominated by at least two dominators $u$ and $v$, and node $w$ has not yet received a message "I am dominated by $u$ and $v"$, then node $w$ transmits "I am dominated by $u$ and $v" and enters the CDS.

- And since this is still not quite enough…

4. If a neighboring pair of nodes $w_1$ and $w_2$ is dominated by dominators $u$ and $v$, respectively, and have not yet received a message "I am dominated by $u$ and $v", or "We are dominated by $u$ and $v", then nodes $w_1$ and $w_2$ both transmit "We are dominated by $u$ and $v" and enter the CDS.

Results

- The algorithm for the Euclidean Unit Disk Graph produces a connected dominating set.
- The algorithm is completely local
- Each node only has to transmit one or two messages of constant size.
- The connected dominating set is asymptotically optimal, that is, $|CDS| = O(|CDS_{OPT}|)$
- If nodes in the CDS calculate the Gabriel Graph GG(UDG(CDS)), the graph is also planar
- The routes in GG(UDG(CDS)) are “competitive”.

- But: is the UDG Euclidean assumption realistic?