ANALYSIS AND ALGORITHMS FOR PARTITIONING OF LARGE-SCALE ADAPTIVE MOBILE NETWORKS

Bin Xiao, Qingfeng Zhuge, Edwin H.-M. Sha *

Department of Computer Science
University of Texas at Dallas
Richardson, Texas 75083, USA
{bxiao, qfzhuge, edsha}@utdallas.edu

Chantana Chantrapornchai

Department of Mathematics
Silpakorn University
Nakorn Pathom, Thailand 73000
ctana@su.ac.th

ABSTRACT

In adaptive mobile network like battle fields or military applications, the base station configuration may change dynamically. Mobile nodes communicate via a base station and can move freely. This dynamic change together with the constraints make the assignment of mobile nodes to base stations difficult. In this paper, we propose the polynomial-time algorithms to find a good partitioning of mobile node assignment. We also show that the assignment of mobile nodes to base stations under bandwidth and communication constraints is NP-complete. We propose two graph models to represent mobile node communication requirement as well as base station configuration. Our simulation results show that our techniques can efficiently give a good performance close to the exhaustive approaches which minimize the frequency of a node that is not covered by a base station in the dynamic environment.
Keywords: Partitioning algorithms, Adaptive mobile network.

1. INTRODUCTION

A large scale of mobile communication units that may represent tanks, airplanes, commanding centers etc. need real-time communications among them. Some units may communicate with some others directly point-to-point. However, for most of cases, the communications need to be switched and routed by a network of base stations. A mobile communication unit may be inserted, deleted or moved to another geographical area. Therefore the connection requirement is dynamically changed. Moreover, the network of base stations is also likely to be changed dynamically. A base station may be added or deleted because of the nature of a battle field. The dynamic structure becomes a challenge for this type of adaptive network.

A similar wireless network can be found in mobile network where mobile consumers can move randomly and freely while base stations are fixed. However, the network also does not consider the possible change in base station configuration. On the other hand, in an ad hoc wireless network there is no base station. An ad hoc network consists of wireless mobile nodes which can act as a router [1]. Many routing algorithms [2–5] are proposed for a better system routing performance. The clustering algorithm [6–8] is one of the important basis, which dynamically selects a cluster head in a cluster, which behaves as a virtual base station. The cluster head may become the bottleneck of the cluster for system performance. Both system models cannot handle the dynamically change in wireless network we mentioned above well.

To represent the adaptive wireless network, we propose two graph models called *node connection graph* and *base station network graph*. The first one represents the connection requirements of mobile units while the second one is for the network of base stations. These graphs represent the actual geographical configurations. From the two models, finding an assignment of a node to a base station is reduced to partitioning problem.

In this paper, we propose efficient algorithms to assign mobile units to base stations base on the two graphs. A valid partition assignment should satisfy all constraints of each base station in terms of bandwidth requirement as well as the communication requirement of mobile nodes. An efficient partition should minimize the number of nodes that cannot satisfy its communication requirement. We also show that the partitioning problem in our case is NP-complete. We propose three algorithms, i.e. ANB, $GREEDY$, $IGREEDY$ and compare them with another two algorithms, i.e. $MIN\ CUT$ and $MAX(MIN\ cv\ left)$ which exhaustively find an optimal solution. After the initial partition, mobile nodes can move freely and randomly. Our simulation results show that while $MIN\ CUT$ algorithm has the best result initially, the result is closed to our approaches, $GREEDY$, $IGREEDY$ algorithms in a long run. Of all the three algorithms, $IGREEDY$ performs better than ANB which reduces the nodes not covered by any base station up to 42%.

The rest of this paper is organized as follows. In Section 2 we introduce graph models and definitions to be used in the paper. We show that our partitioning problem is NP-complete in Section 3. Efficient algorithms are presented in Section 4. Section 5 describes the simulation model and the simulation results are discussed in Section 6. Finally, conclusions are drawn in Section 7.

2. DEFINITIONS AND MODELS

We now define some notation to be used in the rest of the paper.

Definition 2.1. *A node connection graph* $G_N = \langle V_N, E_N, w, S\rangle$ *is an edge-weighted undirected graph where* V_N *is a set of nodes represent mobile units,* E_N *is a set of edges represent communication requirement between mobile nodes,* $w(e_n)$ *represents bandwidth requirement of each edge, and* $S(v_n)$ *represents a set of base station candidates for each node.*

This work is partially supported by TI University Program, NSF EIA-0103709 and Texas ARP 009741-0028-2001, USA.

This work is partially supported by NECTEC, NT-B-06-4D-16-509, Thailand.

A mobile node may actually contain one mobile unit or a group of mobile units that can communicate with each other directly without any channel involvement of a base station. Different nodes can only communicate with each other by transmission of a base station or a relay of base stations. Because of the geographical constraint, a node cannot connect to every base station. A set $S(v)$ represents which base stations that node v is able to connect to. The bandwidth requirement of an edge e $(u \rightarrow v)$ is represented by $w(e)$. If u and v are handled by the same base station, we should make sure the internal base station capacity is large enough to handle $w(e)$. If u and v are mapped to different base station, we should make sure that the bandwidth capacity between these two base stations is sufficiently large.

Definition 2.2. *A base station network graph $G_B = \langle V_B, E_B, ce, cv\rangle$ is a node-weighted edge-weighted undirected graph where V_B is a set of nodes represent base stations, E_B is a set of edges represent communication links between base stations, the edge attribute $ce(e_b)$ for $e_b \in E_B$ represents bandwidth capacity of each edge, and the node attribute $cv(v_b)$ for $v_b \in V_B$ represents bandwidth capacity of a base station for communicating for a set of mapped mobile nodes in G_N. These mapped mobile nodes are communicating inside base station v_b.*

The challenge of mapping a set of mobile nodes to a set of base stations arises because of the mobility of both base stations and mobile nodes. The topology of base station network can be changed, and so can the topology of node connection. The mobility of mobile nodes leads to the change of $S(v)$. The challenge may also come from dynamically insertion or deletion of both base stations and mobile nodes.

We define a legal mapping as follows:

Definition 2.3. *A legal mapping is a mapping from the node set in node connection graph G_N to the node set in base station graph G_B, such that it satisfies the following constraints:*

- *$cv(b_i) \geq \sum w(e_i)$, where $cv(b_i)$ is the bandwidth capacity of a base station b_i, and for all $e_i = u \rightarrow v$ in G_N such that both u and v are mapped to the base station b_i.*
- *Let $W_{i,j}$ represent the total communication requirement between base station b_i and b_j after the mapping. Then, the minimum cut of all the possible paths from b_i to b_j with respect to the edge capacity of G_B must be greater than or equal to $W_{i,j}$.*
- *If v is mapped to b_i, b_i must be in $S(v)$.*

Based on the G_N, G_B and legal mapping definition, our problem can be described as follows:

Establish a mapping from a set of nodes in G_N to a set of nodes in G_B, whose mapping result satisfies the bandwidth capacity constraints for any two connected base station in G_B and the communication as well as bandwidth requirements for any two connected mobile nodes in G_N.

Given $G_N = \langle V_N, E_N, w, S\rangle$ and $G_B = \langle V_B, E_B, ce, cv\rangle$, after finding a legal mapping, we define:

- $E_{inside}(b_i)$ for $b_i \in V_B$ is a set of edges in E_N whose corresponding nodes are assigned to base station b_i.
- $E_{across}(b_i, b_j)$ for $b_i, b_j \in V_B$ is a set of edges $x - y$, where the communication between node x and y goes through base station b_i and b_j.
- $cv'(b_i)$ for $b_i \in V_B$, as $\sum w(e)$ for $e \in E_{inside}(b_i)$.

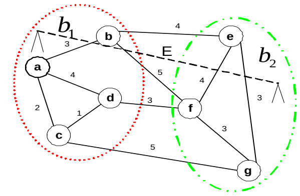

Figure 1: A partition for two base stations.

- $ce'(b_i, b_j)$ for $b_i, b_j \in V_B$, as $\sum w(e)$ for $e \in E_{across}(b_i, b_j)$. Let $ce'_{base}(b_i)$ be $\sum_{j \neq i} ce'(b_i, b_j)$.
- $ce_{base}(b_i)$ for $b_i \in V_B$, as $\sum ce(e_b)$ for $\forall e_b, e_b \in E_B$ where $e_b : x - b_i$ for some x.

Let us first consider a simple case with base stations number is two. Given a graph G_N, the problem is to find a good partition of nodes in V_N for each base station in G_B, such that the summation of edge weights inside the partition conformed with a bandwidth capacity of an associated base station and the summation of the edge weights across the partition conformed the bandwidth capacity between two base stations. An edge of G_N may be a partition across two base stations for a legal mapping. Note that the edge inside the partition will consume a base station's bandwidth capacity cv, while the edge across the two partitions will consume the bandwidth capacity ce of the two connected base stations.

In Figure 1, there are two base stations, b_1 and b_2. For b_1, $E_{inside} = \{(a,b),(a,d),(a,c),(c,d)\}$, $cv' = 3 + 4 + 2 + 1 = 10$. For b_2, cv' is 10 and ce' is 17. $E_{across} = \{(b,e),(b,f),(d,f),(c,g)\}$ and $ce'_{base}(b_1) = ce'_{base}(b_2) = ce'(b_1, b_2) = 4+5+3+5 = 17$. If this is a legal mapping by satisfying Definition 2.3, the G_B must have $cv(b_1) \geq 10$, $cv(b_2) \geq 11$, $ce(E) \geq 17$.

3. PARTITION PROBLEMS

To solve the problem defined in Section 2 we consider different minimization criteria as follows:

INSTANCE: Graph $G_N = \langle V_N, E_N, w, S\rangle$ for node connection, graph $G_B = \langle V_B, E_B, ce, cv\rangle$ for base station, n for the number of nodes in V_B, nonnegative integer K.

- **Question 1** Can we partition V_N into n disjoint sets $V_{B_{b_1}} \ldots V_{B_{b_n}}$ such that for each base station b_i, $cv(b_i) - cv'(b_i)$ is at least K and for $\forall e_i \in E_{across}$ we can find available bandwidth from $e_j \in E_B$ such that $ce(e_j) \geq ce'(e_i)$.
- **Question 2** Can we partition V_N into n disjoint sets $V_{B_{b_1}} \ldots V_{B_{b_n}}$ such that for all base stations, $\frac{1}{n} \cdot \sum_{i=1}^{n} cv'(b_i)$ is at most K and $cv(b_i) \geq cv'(b_i)$, for $\forall e_i \in E_{across}$ we can find available bandwidth from $e_j \in E_B$ such that $ce(e_j) \geq ce'(e_i)$.
- **Question 3** Can we partition V_N into n disjoint sets $V_{B_{b_1}} \ldots V_{B_{b_n}}$ such that for $\frac{1}{n} \sum_{i=1}^{n} ce'_{base}(b_i)$ is at most K and $cv(b_i) \geq cv'(b_i)$, for $\forall e_i \in E_{across}$ we can find available bandwidth from $e_j \in E_B$ such that $ce(e_j) \geq ce'(e_i)$.

In practical, Question 1 is to maintain an available bandwidth k for future burst node communication requirement of every base station. Question 2 tries to minimize cv' and Question 3 tries to minimize ce'. We will show that to find an optimal algorithm for above questions are *NP-complete*.

Theorem 3.1. *The problem for Question 1 is* NP-complete.

Proof.

1. The problem is NP. We can simply test the condition in polynomial time.
2. We know the *PARTITION* [9] problem is a NP-complete problem, which is:

   INSTANCE: A finite set A and a "size" $s(a) \in Z^+$ for each $a \in A$.

   QUESTION: Is there a subset $A' \subseteq A$ such that

   $$\sum_{a \in A'} s(a) = \sum_{a \in A - A'} s(a)? \quad (1)$$

3. For a given instance of *PARTITION* problem, we can follow the below process to transform it to an instance of problem for *Question 1*:

   Let the base station number $n = 2$ and $K = 0$. For $\forall a \in A$, let $cv(b_1) = cv(b_2) = \frac{1}{2}\sum s(a)$ and the weight of ce for every pair of base stations be a very large number.

   For each $a_i \in A$, we will have an edge with the same weight as $s(a_i)$ for G_N. The two nodes connected by the edge can be named as v_{2i} and v_{2i+1}, i.e., $V_N = \{v_{2i}, v_{2i+1} | \forall a_i \in A\}$ and $|V_N| = 2 \cdot |A|$. We connect node v_{2i} to every other nodes (except node v_{2i+1}) with a weight $\frac{1}{2}\sum s(a) + 1$. Thus graph G_N is a complete graph and for every edge, its weight is either in the set of A (if the edge is between nodes v_{2i} and v_{2i+1}) or $\frac{1}{2}\sum s(a) + 1$. Each node in V_N can be covered by any base station. This transformation can be done in polynomial time.

   According to the bandwidth constraint for legal mapping, the edges with weight $\frac{1}{2}\sum s(a) + 1$ can not be in neither base station b_1 nor b_2. Then a partition of node set V_N in graph G_N to two node sets $V_{B_{b_1}}$ and $V_{B_{b_2}}$ guarantees that all edges with weight in the set A is either in base station b_1 or in b_2. With a legal mapping, only those edges can be inside one base station. Otherwise, some nodes may not be included by base stations. Because we define $K = 0$ and $cv(b_1) = cv(b_2) = \frac{1}{2}\sum s(a)$, if we have a solution to *Question 1*, we must have the same solution to *PARTITION* instance of NP-complete problem.

□

Theorem 3.2. *The problem for Question 2 is* NP-complete.

Proof.

1. The problem is NP. We can simply test the condition in polynomial time.
2. The *MAX CUT* problem in [9] is NP-complete, which is described as:

   INSTANCE: Graph $G = (V', E')$, weight $u(e) \in Z^+$ for each $e \in E$, positive integer M.

   QUESTION: Is there a partition of V into disjoint sets V_1 and V_2 such that the sum of the weights of the edges from E that have one endpoint in V_1 and one endpoint in V_2 is at least M ?

3. For a given instance of *MAX CUT* problem, we can follow the below process to transform it to an instance of problem for *Question 2*:

   Let the base station number be $n = 2$ and two base stations be b_1 and b_2. For graph G, let $J = \sum w(e), \forall e \in E'$. Graph G_N has the same node and edge set as graph G, which means $E = E'$, $V_N = V'$ and $\forall e, w(e) = u(e)$. Let $K = \frac{1}{2}(J - M)$ and the weight of cv, ce for every base station, i.e. $cv(b_i), ce_{base}(b_i)$, be J. Every node can be covered by either b_1 or b_2, which defines S for graph G_N. The formation of graph G_N and G_B can be done in polynomial time.

   We can prove that if we can solve the problem for Question 2, we can find the solution for the *MAX CUT* problem. A cut makes graph G_N into two parts. Each part is covered by one base station. That means the edge between the two connected base stations will have weight ce', which is the value of cut. Since $J = cv'(b_1) + cv'(b_2) + ce'(b_1, b_2)$, graph G_N, J is a fixed value. If $ce'(b_1, b_2)$ is at least M, $\frac{1}{2}(cv'(b_1) + cv'(b_2))$ is at most $\frac{1}{2}(J - M)$, which is K. Thus given that each node can be covered by b_1 or b_2, if we have a solution for question 2 for graph G_N, $\frac{1}{2}\sum_{i=1}^{2} cv'(b_i)$ is at most K. That means the cut for partition is at least M, which is the solution for the *MAX CUT* NP-complete problem for graph G.

□

Theorem 3.3. *The problem for Question 3 is* NP-complete.

Proof.

1. The problem is NP. We can simply test the condition in polynomial time.
2. The *MINIMUM CUT INTO BOUNDED SETS* problem in [9] is NP-complete. The problem is:

   INSTANCE: Graph $G = (V', E')$, weight $u(e) \in Z^+$ for each $e \in E$, specified vertices $s, t \in V$, positive integer $B \leq |V|$, positive integer M.

   QUESTION: Is there a partition of V into disjoint sets V_1 and V_2 such that such that $s \in V_1, t \in V_2$, $|V_1| \leq B$, $|V_2| \leq B$, and such that the sum of the weights of the edges from E that have one endpoint in V_1 and one endpoint in V_2 is no more than M ?

3. For a given instance of *MINIMUM CUT INTO BOUNDED SETS* problem, we can follow the below process to transform it to an instance of problem for *Question 3*:

   Let the base station number be $n = 2$, and we have b_1 and b_2. Graph G_N has the same node and edge set as graph G, which means $E = E'$, $V_N = V'$ and $\forall e, w(e) = u(e)$. Let $J = \sum w(e), \forall e \in E$, the weight of cv, ce for each base station be J. We build S of graph G_N such that each base station can cover B nodes, which are from nodes in V_N and node s is only in b_1 while t only for b_2. Let $K = M$. The transformation to graph G_N and G_B can be done in polynomial time.

   Two base stations are connected with weight ce'. We know that a cut with ce' less or equal that K is an answer for Question 3. If we have a solution for Question 3 with graph G_N, we find a partition with $ce' = K$, which is also a cut with value no more than $M = K$ for the *MINIMUM CUT INTO BOUNDED SETS* problem with graph G.

□

4. ALGORITHMS

We already know that the partition of nodes to different base stations under certain criteria is an NP-complete problem. For a small system, it is possible to get an answer by exhaustively generating all possible partitions. However, in general we have many nodes and base stations. An exhaustive approach is very much time consuming and thus inefficient. Further, even if we get the best solution for an initial partition, when the mobile nodes move the partition has to be recalculated. Otherwise, some nodes may not be able to communicate.

In this section, we present heuristic algorithms to solve the partition problem. According to Definition 2.1, each node can be in transmission range of several base stations. We propose three different algorithms below:

- **ANB:** Assign a node to the closet base station.
- **GREEDY:** Assign a node to the base station that has the most available capacity.
- **IGREEDY:** Assign a node to a base station to maintain the maximum system overall available bandwidth.

For a small network such as 10 mobile nodes with 2 base stations, one may use the exhaustive approach as follows.

- **MIN CUT:** Generate all possible partitions and select the one that has a minimum cut. If there is a tie, the partition which contains the maximum number of nodes will be selected first. This is according to Question 3 defined in Section 2.
- **MAX(MIN cv left):** For all possible partitions, select one that has maximum value of all the minimum available cv capacity for all base stations. If tie, the partition which contains the maximum number of nodes will be selected first. This is according to the Question 1 defined in Section 2.

Note that we do not propose any method to solve Question 2 defined in Section 2 by an exhaustive approach. This is because when cv value is small for a partition, the ce value would be large. Usually, cost ce' for the communication between base stations are more expensive compared with cost cv' for the communications inside a base station.

4.1. ANB Algorithm

Given a node connection graph G_N and a base station network graph G_B, for node N_i, $S(N_i) = \{b_{i_1}, b_{i_2}, \ldots, b_{i_k}\}$, which means node N_i is in the range of any base station from b_{i_1} to b_{i_k}. ANB algorithm is shown as following:

1. **Initialization:** Let node set $V_{all} = V_N$, which contains all nodes in G_N. Let node set $V_{partitioned} = \phi$.
2. **Partition:** Arbitrarily select one node from node set V_{all} and add it to set $V_{partitioned}$. Let this node be N_i. If node N_i is in base station range of b_{i_1} for a legal mapping, this node is assigned to base station b_{i_1}. Otherwise, we continue to test all other available base stations in set $S(N_i)$ until we find a legal mapping. If none of them satisfies the legal mapping, this node can not be assigned to any base station.
3. If $V_{all} \neq \phi$, go to step 2.

If $|V_N| = n$, in Step 2, the time complexity is $O(n)$. ANB algorithm finished after n loops, which can be seen from Step 3. Thus the time complexity for this algorithm is $O(n^2)$.

4.2. GREEDY Algorithm

The $GREEDY$ algorithm is similar to ANB in Step 1 and Step 3. The difference is in Step 2, which is as follows.

Partition: Arbitrarily select one node from node set V_{all} and add it to $V_{partitioned}$. Let this node be N_i. For all available base stations in $S(N_i)$, select one which has the most available bandwidth. Assign node N_i to this base station. If this is a legal mapping, go to Step 3. Otherwise, we continue to test all other available base stations in $S(N_i)$, starting from the maximum available capacity in descending order until we find a legal partition. If we cannot find a legal mapping, this node cannot be assigned to any base station.

The available bandwidth for base station b_i can be described as $\gamma * (cv(b_i) - cv'(b_i)) + (1-\gamma) * (ce_{base}(b_i) - ce'_{base}(b_i))$ and γ is a parameter which is $0 \leq \gamma \leq 1$. The time complexity for $GREEDY$ algorithm is the same as ANB, which is $O(n^2)$.

4.3. IGREEDY Algorithm

This algorithm is improved from $GREEDY$ algorithm. The difference is that we don't select nodes in arbitrary sequence. Let $\#(N_i)$ be $|S(N_i)|$ for node N_i. We first consider nodes whose $\#(N_i)$ is small and assign partition to partition them. Then the algorithm can be described as follows.

1. **Initialization:** Let node set $V_{all} = V_N$, which contains all nodes in it. Let node set $V_{partitioned} = \phi$ and $Level = 0$, and $n = |V_B|$.
2. Select all nodes N_i from node set V_{all} with $\#(N_i) = Level$. Add all these nodes to set $Node(N_i)$.
3. From set $Node(N_i)$, select one node N_i such that if we assign it to be in one available base station, the new system cost $(\sum_{i=1}^{n} cv'(b_i) + \sum_{i=1}^{n} ce'_{base}(b_i))$ is smallest. Move this node N_i from node set V_{all} to node set $V_{partitioned}$. Remove this node from node set $Node(N_i)$.
4. If $Node(N_i) \neq \phi$, go to step 3. Otherwise, let $Level = Level + 1$.
5. If $V_{all} \neq \phi$, go to step 2.

In Step 3 if node N_i cannot be assigned to any base station because of capacity constraints, the new system cost can be infinite. If there are ties, we arbitrarily choose one. The system cost here is used only for deciding which node to be considered next. In Step 3, the time complexity to get the new smallest system cost is $O(n^2)$ because for every node in the same level, we can compute the smallest system cost in $O(n)$ and we need to search the smallest cost for all nodes in this level, which takes another $O(n)$ time. Thus the improved greedy algorithm has the time complexity $O(n^3)$.

Given two base stations, as a and b, each one has the same radius which is 50 meters and they are in each other's transmission range. Ten mobile nodes are created randomly in a rectangle area by 135 (meters) * 80 (meters). The nodes closed to each other will have a high rate of busy communication. Thus, we assign an edge between them with a large weight. Each node has the number of edges between 1 and 4. The graph is shown in Figure 2.

From Table 1, we can see that different algorithms have different partition results under the graph in Figure 2. This is not the case however. With some graphs, some algorithms will have the same results. The resulting partitions show that

Algorithms	Base Station a		Base Station b		$ce'(a,b)$	Nodes not covered
	covered nodes	cv'	covered nodes	cv'		
ANB	0,1,2,4,5	14	6,8,9	0	15	3,7
MIN CUT	0,2,7,9	11	1,4,5,6,8	15	11	3
MAX(MIN cv left)	0,2,5,7	10	1,4,6,8,9	10	17	3
GREEDY	2,5,7,9	12	0,1,4,6,8	9	16	3
IGREEDY	2,4,7	5	0,1,6,8,9	4	13	3,5

Table 1: Partition comparison for different algorithms.

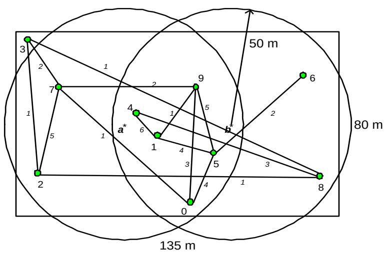

Figure 2: An example for arbitrary nodes in two base stations.

MIN CUT, MAX(MIN cv left) and *GREEDY* have a better result which has only one node not covered by any base station. Note that when mobile nodes are moving, some nodes may not be covered by any base station because they roam away from its current base station. Thus the good initial partition will result in the minimized number of nodes which are not assigned to a base station on overall. In Table 1, it is shown that *MIN CUT* has least ce' cost which is 11 while *MAX(MIN cv left)* with maximum of minimum $cv - cv'$ which is 6.

5. SIMULATION MODEL

We evaluate the performance of different algorithms through graphs created randomly where mobile nodes are moving as time passes. The performance of algorithms is measured by the number of nodes which are not covered by any base stations during a fixed time interval. In the simulations, we compare different algorithms.

As we proved in Section 2 that *MIN CUT* and *MAX(MIN cv left)* are both NP-complete, the simulation for these two algorithms for a large system takes a lot of time. We need to search all possible partitions and select the best partition where the complexity is exponential. From Figure 2 to generate all possible partitions of nodes $0,1,4,5,9$ would require that we compare 3^5 partitions (and in some cases we require 3^{10} comparisons if all ten nodes are both in base a and base b). We prepare two systems, called small system and large system. In the small system, all five algorithms are compared, while in the large system we only simulate *ANB, GREEDY* and *IGREEDY* algorithms.

- **Small System:** A small system use the configuration as in Figure 2. Two base stations have fixed center and they both can communicate with the other. Each one has a radius of 50 meters. The number of mobile nodes is 10, which are randomly scattered in a rectangle area by 135 (meters) * 80 (meters).
- **Large System:** 30 mobile nodes and 5 base stations are simulated for a large system. Initially, all nodes are randomly scattered in a rectangle area by 120 (meters) * 100 (meters), while all base stations' center are arbitrarily scattered in a small rectangle area by 80 (meters) * 70 (meters), which is shown in Figure 3. The radius of all base stations is the same which is 60 meters.

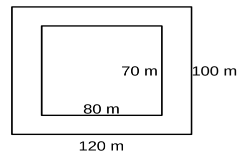

Figure 3: The large rectangle contains mobile nodes and the small rectangle contains scattered base station centers.

Each base station has the same bandwidth capacity cv. This is determined by the total edge weights and the number of base stations in the system. The available bandwidth ce for links between base stations is selected in such a way that all available $ce_{base}(b_i)$, for the base station b_i is a little bigger than its cv.

The weight of an edge in the *node connection graph* means that the two corresponding mobile nodes require the same bandwidth for their communication. One node is covered by a base station if and only if all its outgoing edges consume the same from the base station, except for some edges to other nodes that are not covered by any base station. For one edge whose nodes are in the same base station, its weight will consume the cv of the base station. For an edge $u \to v$ where u, v are in two different partitionsb_1, b_2, $w(u,v)$ will be subtracted from $ce(b_1, b_2)$. When nodes are moving, their edges' weights are kept the same. For example, the partition of graph in Figure 2 by algorithm ANB makes base station a to cover nodes $0, 1, 2, 4, 5$, its used channel bandwidth is $cv' = 14$ and $ce' = 15$. The cost of cv' comes from $w(0,5) + w(1,4) + w(1,5) = 4 + 6 + 4 = 14$, while ce' derives from $w(0,9) + w(1,9) + w(2,8) + w(4,8) + w(5,6) + w(5,9) = 3 + 1 + 1 + 3 + 2 + 5 = 15$.

When we have more than two base stations, two base stations may not communicate with each other directly. We should find a path through several base stations, which forms a communication path for two connected nodes. Thus the weight of the edge will consume the ce along the path. During the simulation for algorithms $GREEDY$ and $IGREEDY$, we try to find the shortest path between two connected nodes, which has a small number of base stations along the path.

After finding an initial partition for a given graph, nodes can move randomly. For the initiated graph, nodes which are closed to each other have a high probability for their busy communication, which means the edges connecting between them have a large weight. For connection for two nodes with a large distance, the edge weight will be small. If a node moves away from its current base station, it may need to find a new base station. The system is simulated many iterations. An iteration begins with an arbitrarily created graph and is simulated for 10,000 time slots. After several time slots, any node who is not covered by a base station will request for available base station. If there are several base stations available at the same time, above algorithms will arbitrarily select one.

6. SIMULATION RESULTS

The performance of an algorithm is measured by the number of nodes that are not covered by any base station for a fixed period. In every iteration, a new *node connection graph* is simulated for the duration of 10,000. In the large system, we have a new *base station network graph* for each iteration. For a small system, we compare the results of all five algorithms assuming two fixed base stations. The results are shown in Table 2.

Algorithms	ANB	MIN CUT	MAX	GREEDY	IGREEDY
1 iteration	16,061	6,061	6,061	6,061	6,061
5 iterations	12,757	6,601	9,616	10,185	8,807
10 iterations	14,246	9,548	11,055	11,321	11,020

Table 2: Average # mobile nodes not covered for a small system with five different algorithms in one iteration.

From Table 2, when each base station has limited bandwidth capacity, the ANB algorithm gives the worst result while $\mathsf{MIN\ CUT}$ gives the best result. However the data show that the other four algorithms do not have too much difference when the iteration numbers become large. This is because the algorithms give an initial partition. Even though the initial partition shows that most nodes are covered, it cannot guarantee that when nodes are moving the same partition will still remain the best result. When only with one iteration, the results have much random property for special case. However with more iterations, we get a fluent reasonable results. Thus for a big system, even in a real system GREEDY and $\mathsf{IGREEDY}$ algorithms can be efficient partition methods. For a large system with 30 nodes and 5 base stations, we compare three algorithms, ANB, GREEDY and $\mathsf{IGREEDY}$, as shown in Table 3.

Algorithms	ANB	GREEDY	IGREEDY
1 iteration	28,692	12,639	18,692
5 iterations	35,032	24,696	14,092
10 iterations	37,553	26,150	21,681

Table 3: Average # mobile nodes not covered for a large system with three different algorithms.

The simulation results in both tables show that $\mathsf{IGREEDY}$ algorithm always comes up with good graph partition compared with ANB and GREEDY algorithms. For a small system, the partition given by $\mathsf{IGREEDY}$ algorithm shows about 1.1 nodes not covered by any base station in one time slot for a small system for 10 iterations. While in a large system about 2.1 nodes are not covered by any base station per time slot. The available base stations and their available bandwidth directly affect results. For the large system in Table 3 if we increase cv and ce by half, new results are shown in Table 4. All results show that the number of nodes not covered by a base station decreased for every algorithm.

Algorithms	ANB	GREEDY	IGREEDY
1 iteration	18,692	12,639	18,692
5 iterations	23,032	19,241	13,284
10 iterations	28,553	23,423	21,277

Table 4: Average # mobile nodes not covered for a large system when bandwidth available is increased for three different algorithms.

7. CONCLUSION

In a mobile network, base station is fixed and cannot move. A mobile node may not be covered by a base station when it moves. The challenge of the problem is that new mobile nodes and new base stations maybe introduced dynamically. With this dynamic nature, we propose algorithms to find a legal initial partition, which assign mobile nodes to base stations to minimize the nodes not covered by any base station. We show that, to find the best partition for the problem will be NP-complete. Our simulation results show that the proposed algorithms, i.e. GREEDY and $\mathsf{IGREEDY}$, give the performance close to optimal ones by $\mathsf{MIN\ CUT}$ and $\mathsf{MAX(MIN}\ cv\ \mathtt{left})$. The results also show that on average $\mathsf{IGREEDY}$ algorithm has a better performance than the GREEDY algorithm. The time complexities of both algorithms are $O(n^3)$ and $O(n^2)$ respectively.

8. REFERENCES

[1] D. B. Johnson, “Routing in ad hoc networks of mobile hosts,” in *Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications*, Dec 1994.

[2] E. Royer and C.-K. Toh, “A review of current routing protocols for ad hoc mobile wireless networks,” *IEEE Personal Communications*, vol. 6, pp. 46–55, April 1999.

[3] Y.-C. Tseng, S.-L. Wu, S.-L. Wu, and C.-M. Chao, “Location awareness in ad hoc wireless mobile networks,” *IEEE Computer*, vol. 34, pp. 46–52, June 2001.

[4] Z. Haas and M. Pearlman, “The performance of a new routing protocol for the reconfigurable wireless networks,” in *Communications, 1998. ICC 98. Conference Record. 1998 IEEE International Conference on*, vol. 1, 1998.

[5] M. Pearlman and Z. Haas, “Determining the optimal configuration for the zone routing protocol,” *Selected Areas in Communications, IEEE Journal on*, vol. 17, pp. 1395–1414, Aug. 1999.

[6] C. R. Lin and M. Gerla, “Adaptive clustering for mobile wireless networks,” *IEEE Journal of Selected Areas in Communications*, vol. 15, no. 7, pp. 1265–1275, 1997.

[7] D. Kim, S. Ha, and Y. Choi, “Cluster-based dynamic source routing protocol in wireless ad-hoc packet radio network with variable-sized clusters and variable transmission ranges,” in *Vehicular Technology Conference*, vol. 3, pp. 2441–2445, 1999 IEEE 49th, 1999.

[8] S. Basagni, “Distributed clustering for ad hoc networks,” in *Proceedings of the 1999 International Symposium on Parallel Architectures, Algorithms, and Networks (I-SPAN’99),IEEE Computer Society*, pp. 310–315, June 1999.

[9] M. R.Garey and D. S. Johnson, *Computers and Intractability–A Guide to the Theory of NP-Completeness*. San Francisco, CA: Freeman, 1979.