Heuristics for Multiway Partitioning in Hexagonal Cellular Systems

Kyungshik Lim and Yann-Hang Lee
Computer and Information Sciences Department
University of Florida
Gainesville, FL 32611

Abstract

Given a hexagonal mesh of base stations in cellular systems we consider the problem of finding a cover of disjoint clusters of base stations which generate multiple types of traffic among themselves. The problem differs from general graph partitioning problems in that it considers not only communication costs but also the underlying topology among base stations, such that base stations in a cluster are connected in their physical topology. The objective is to minimize the total communication cost for the entire system where inter-cluster communication is more expensive than intra-cluster communication for each type of traffic. The problem is transformed into the dual based on a topology matrix and a relative cost matrix. We develop several heuristics for the dual. These heuristics produce optimal partitions with respect to the initial partition, based on the techniques of moving or interchanging the boundary nodes between adjacent clusters. The heuristics are compared and shown to behave quite well through experimental tests and analysis.

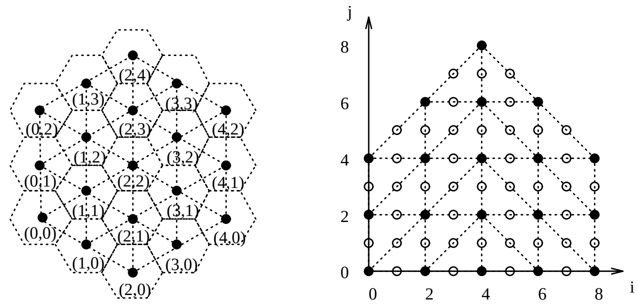
1 Introduction

In wireless communications networks, whenever there is a need to establish communication with any particular user, the network has first to find out which one of base stations can communicate with the user. In the full-information strategy, each mobile user transmits location update messages whenever it moves to a new base station and every base station maintains a complete location information for every user. This makes a *move* operation to perform handoff very expensive but makes a *find* operation to locate the current base station of a mobile user very cheap. In the no-information strategy, on the other hand, mobile users never send location update messages and whenever there is a need to locate a particular user, the global search over the entire network is performed. This makes a move operation very cheap but makes a find operation very expensive. Based on the combination of these two extreme strategies, a number of efficient location tracking strategies have been reported in the literature [1, 2, 3, 4]. A hierarchical location server structure using the graph-theoretic concept of regional matching is constructed to give the upper bound of the communication cost for a sequence of find and move operations in [1, 2]. A partial information strategy based on the reporting center concept is presented in [3] and the issues of querying locations in wireless environments are discussed in [4].

Let us assume that we have an efficient location tracking strategy which involves the two primitive operations. Given the frequencies of move and find operations among base stations, we consider the problem of assigning location servers to base stations. Location servers are connected among themselves and to base stations by fixed networks. Since base stations are used as the interface between mobile users and fixed networks, they are regarded as traffic sources and destinations from the prospective of location servers. The cost of tracking mobile users within the area administered by a location server is usually much lower than that of between the areas administered by different location servers. Thus, the location server assignment problem is concerned with optimal partitioning of base stations, so as to minimize the total communication cost of the entire system for a sequence of move and find operations.

In addition to the frequencies of move and find operations among base stations, the problem also considers the physical deployment of base stations so that base stations in a cluster are contiguously connected in their physical topology, not scattered. While the frequency is represented by a complete directed graph among base stations for a possible communication between mobile hosts through different base stations, the physical topology is represented by a linear graph in highway cellular systems and by a hexagonal mesh graph in cellular systems. Hence, the location server assignment problem is concerned with the two types of graphs, the topology graph for partitioning base stations and the frequency graph for optimizing communication cost.

For general graphs, the problem of finding a cover of disjoint sets such that the sum of all edge weights, whose two endpoints are in two different sets, is equal to or less than a given positive integer is known NP-complete for the arbitrary size of a set[5]. The k-cut problem of finding a partition of vertices into k nonempty clusters such that the total edge weight between clusters is minimum is also NP-complete for arbitrary k. The k-cut problem with specified vertices which is an extension of the 2-cut problem solvable via repeated applications of a max-flow min-cut algorithm becomes NP-hard even for $k = 3$[6]. Given an s-outerplanar embedding of an s-outerplanar graph, an optimal solution for maximum independent set can be obtained in time $O(8^s n)$ by a dynamic programming technique, where n is the number of nodes[8]. If an n-vertex planar graph G is given with an s-outerplane-separable planar embedding of G, then the optimal partition of two clusters of fixed size is determined in time $O(s^2 n^3 2^{3s})$ by a dynamic programming technique[9].



(a) The (column, row) indexing (b) The two-dimensional labeling

Figure 1: The Labeling of the Physical Topology for H_3

Given the frequencies of move and find operations among n base stations in highway cellular systems, we have shown an efficient optimal partitioning algorithm of $O(mn^2)$ for an arbitrary number of clusters m by dynamic programming[12]. In this paper, we consider the location server assignment problem in hexagonal cellular systems. Section 2 introduces a labeling scheme to formalize the problem mathematically. With the labeling scheme, the problem is transformed into the dual based on a topology matrix and a relative cost matrix. The topology matrix reflects the underlying topology constraint on clustering, while the relative cost matrix reflects the communication cost of multiple types of traffic to be optimized. Section 3 presents several heuristics for the dual. These heuristics produce optimal partitions with respect to the initial partition, based on the techniques of moving or interchanging the boundary nodes between adjacent clusters. The heuristics are compared and shown to behave quite well through experimental tests and analysis in Section 4. Finally we conclude in Section 5.

2 The Problem Formalization

2.1 The Labeling Scheme

The physical deployment of n base stations is represented by a planar graph of a hexagonal mesh of n base stations, where the vertices represent base stations and the edges represent the adjacency of base stations. The vertices on the exterior face of the graph are level 1 vertices and the vertices on the exterior face of the subgraph induced by removing level 1 vertices are level 2 vertices, and so on. A planar graph is *s-outerplanar* if it has no vertices of level greater than s. Denote H_s as an s-outerplanar graph of a hexagonal mesh. It is then known that the number of vertices in an H_s is $n = 3s^2 - 3s + 1$ and the number of columns in each of three directions is $d = 2s - 1$. Before formalizing the problem, it is necessary to introduce a labeling scheme to describe the problem mathematically.

Starting from the left-most column of an H_s, each column is indexed from 0 through $d-1$ in sequence. Then the bottom vertices of every column constitute row 0, the next vertices of every column row 1, and so forth. Once the column index c and the row index r of a vertex is determined, the vertex is labeled (i,j) such that $i = 2c$ and $j = 2r$. In this two-dimensional coordinate system of an H_s, a point (i,j) for $0 \leq i,j \leq 2(d-1)$ can represent either a vertex v_{ij} if i and j are even or an edge e_{ij} otherwise. Figure 1 illustrates the labeling of the physical topology for an H_3 of $n = 19$ nodes.

If given a vertex v_{ij}, at most six edges, each leading to an adjacent vertex, are directly identified by the labeling scheme as follows:

- $e_{i(j+1)}$, $e_{(i+1)(j+1)}$, $e_{(i+1)j}$, $e_{i(j-1)}$, $e_{(i-1)(j-1)}$, and $e_{(i-1)j}$ if $i < d-1$;
- $e_{i(j+1)}$, $e_{(i+1)j}$, $e_{(i+1)(j-1)}$, $e_{i(j-1)}$, $e_{(i-1)(j-1)}$, and $e_{(i-1)j}$ if $i = d-1$;
- $e_{i(j+1)}$, $e_{(i+1)j}$, $e_{(i+1)(j-1)}$, $e_{i(j-1)}$, $e_{(i-1)j}$, and $e_{(i-1)(j+1)}$ if $i > d-1$.

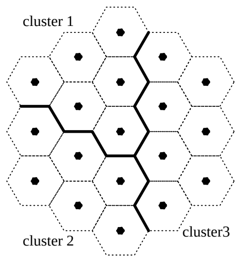
On the other hand, if given an edge e_{ij}, the two vertices connected by the edge are directly identified by the labeling scheme as follows:

- $v_{(i-1)j}$ and $v_{(i+1)j}$ if i is odd and j is even;
- $v_{i(j-1)}$ and $v_{i(j+1)}$ if i is even and j is odd;
- $v_{(i-1)(j-1)}$ and $v_{(i+1)(j+1)}$ if both i and j are odd and $i < d$;
- $v_{(i+1)(j-1)}$ and $v_{(i-1)(j+1)}$ if both i and j are odd and $i > d$.

In the example of Figure 1(b), a vertex v_{24} leads to the six edges e_{13},e_{14},e_{23},e_{25}, e_{34}, and e_{35}, which are connected to the neighboring vertices, v_{02},v_{04},v_{22},v_{26},v_{44}, and v_{46}, respectively.

2.2 The Topology Matrix

To handle the underlying topology constraint on clustering, we construct a topology matrix $\mathbf{T} = (t_{ij})$ for $0 \leq i,j \leq 2(d-1)$, where t_{ij} corresponds to either a vertex v_{ij} if i and j are even or an edge e_{ij} otherwise in the labeling scheme. If v_{ij} currently belongs to cluster P_x, the element t_{ij} is set to x. If the two vertices connected by e_{ij} belong to clusters P_x and P_y, the element t_{ij} is set to xy for $x \leq y$. Figure 2(b) shows the topology matrix $\mathbf{T}$ derived from an initial partition of 3 clusters in Figure 2(a).



	0	1	2	3	4	5	6	7	8
0	2		2		1				
1									
2	2		2		1		1		
3									
4	2		2		1		1		1
5									
6	3		3		3		3		
7									
8	3		3		3				

(a) A partition of 3 clusters

	0	1	2	3	4	5	6	7	8
0	2	22	2	12	1				
1	22	22	22	12	11	11			
2	2	22	2	12	1	11	1		
3	22	22	22	12	11	11	11	11	
4	2	22	2	12	1	11	1	11	1
5	23	23	23	13	13	13	13	13	
6	3	33	3	33	3	33	3		
7	33	33	33	33	33	33			
8	3	33	3	33	3				

(b) The topology matrix T

Figure 2: The Topology Matrix **T** for a Partition of H_3

From the topology matrix **T**, the indices of the boundary edges between a pair of adjacent clusters P_x and P_y can be represented by a list of edge elements $L_{xy} = \{e_{ij} | t_{ij} = xy, x < y\}$. Using the list L_{xy}, the boundary vertices between them can be directly identified by the labeling scheme itself. Denote $V_{\overline{x}y}$ and $V_{x\overline{y}}$ as the indices of the boundary vertices for P_x and P_y, respectively. In the example of Figure 2, $L_{12} = \{e_{03}, e_{13}, e_{23}, e_{33}, e_{43}\}$, $V_{\overline{1}2} = \{v_{04}, v_{24}, v_{44}\}$, and $V_{1\overline{2}} = \{v_{02}, v_{22}, v_{42}\}$.

2.3 The Relative Cost Matrix

The communication among base stations is considered as a full mesh of point-to-point logical network to represent a possible communication between mobile hosts through different base stations. The communication network is described by a complete directed graph $G = (V, E)$, where $|V| = n$. The vertices of the graph represent base stations and the edges represent directional communication links between base stations. Each edge is assigned a move frequency by a function $f_m : V \times V \rightarrow R^+$ and a find frequency by $f_f : V \times V \rightarrow R^+$. Denote w_m^1 and w_m^2 as the weight of a move operation within a cluster and between clusters, respectively, and w_f^1 and w_f^2 as that of a find operation within a cluster and between clusters, respectively. Then we define a relative cost function $c : V \times V \rightarrow R^+$ for all $(v_{ij}, v_{kl}) \in V \times V$ as

$$c(v_{ij}, v_{kl}) = f_m(v_{ij}, v_{kl})(w_m^2 - w_m^1) + f_f(v_{ij}, v_{kl})(w_f^2 - w_f^1), \tag{1}$$

where $(w_m^2 - w_m^1)$ is the relative weight of a move operation and $(w_f^2 - w_f^1)$ is the relative weight of a find operation. The cost $c(v_{ij}, v_{kl})$ represents the total relative cost of $f_m(v_{ij}, v_{kl})$ move and $f_f(v_{ij}, v_{kl})$ find operations from v_{ij} to v_{kl} if they are in different clusters. Denote $c(v_{ij}, v_{kl})$ as $c_{ij;kl}$. The relative cost matrix $\mathbf{C} = (c_{ij;kl})$ for $0 \leq i, j, k, l \leq 2(d-1)$ is derived by the equation (1) from the move and find frequencies among n vertices.

2.4 The Dual

Let $\Pi = \{P_1, \ldots, P_m\}$ be a partition of m clusters of contiguous vertices such that $P_x \cap P_y = \emptyset$ and $\cup_x P_x = V$ for $x, y = 1, \ldots, m$ and $x \neq y$. The intra-cluster communication cost for all $(v_{ij}, v_{kl}) \in V \times V$ is

$$Cost_{intra} = \sum_{(v_{ij}, v_{kl}) \in V \times V} (f_m(v_{ij}, v_{kl}) w_m^1 + f_f(v_{ij}, v_{kl}) w_m^2)$$

and the relative inter-cluster communication cost of Π is

$$Rcost_{inter}(\Pi) \quad = \quad \sum_{x=1}^{m-1} \sum_{v_{ij} \in P_x, v_{kl} \in P_y, x<y} (c_{ij;kl} + c_{kl;ij}).$$

The total communication cost of Π is then

$$Cost_{total}(\Pi) = Cost_{intra} + Rcost_{inter}(\Pi).$$

Because $Cost_{intra}$ is constant independent of how to partition, it holds that Π minimizes $Cost_{total}$ iff Π minimizes $Rcost_{inter}$. Let us define the relative intra-cluster communication cost of Π as

$$Rcost_{intra}(\Pi) = \sum_{x=1}^{m} \sum_{v_{ij}, v_{kl} \in P_x} (c_{ij;kl} + c_{kl;ij}).$$

Because the sum of $Rcost_{intra}$ and $Rcost_{inter}$ is constant, our task of finding an optimal partition Π which minimizes $Rcost_{inter}$ is equivalent to that of finding an optimal partition Π which maximizes $Rcost_{intra}$.

Hence, the dual problem is: given the topology matrix $\mathbf{T} = (t_{ij})$ and the relative cost matrix $\mathbf{C} = (c_{ij;kl})$ for a system of n hexagonal cells, find a cover of m disjoint clusters of contiguous base stations $\Pi = \{P_1, \ldots, P_m\}$, so as to maximize $Rcost_{intra}(\Pi)$.

3 Heuristics

We consider heuristics for the problem: starting with an arbitrary partition Π of m clusters, we try to increase the initial relative intra-communication cost $Rcost_{intra}(\Pi)$ by repeated applications of a two-way optimization procedure to pairs of adjacent clusters. In the two-way optimization procedure, we try to increase the sum of the initial relative intra-communication cost of each cluster by moving or interchanging boundary nodes until no further improvement is possible. The resulting pair of adjacent clusters are then pairwise optimal with respect to the initial partition. Because the two-way optimality for a pair of adjacent clusters may affect that for another pair of adjacent clusters, more than one pass through all pairs of adjacent clusters may be required. This process can be repeated with another initial partition Π of m clusters, and so on, so as to obtain as many locally maximum partitions as we desire. Then, one of the resulting partitions has a fairly high probability of being a globally maximum partition.

3.1 Interchanging or Moving Boundary Nodes

Consider a pair of adjacent clusters P_x and P_y in the two-way optimization procedure. The interchange of a pair of boundary nodes, $v_{ij} \in V_{\overline{x}y}$ and $v_{kl} \in V_{x\overline{y}}$, is said to be *feasible* if the resulting clusters preserve the underlying topology constraint. In other words, all nodes in P_x (P_y) adjacent to v_{ij} (v_{kl}) must be connected in their physical topology before the interchange and v_{ij} (v_{kl}) must be connected to at least one of the nodes in P_y (P_x) after the interchange.

To determine feasible pairs of interchanging nodes, we use the topology matrix **T** where an element t_{ij} corresponds to a node v_{ij} or an edge e_{ij}. Given a node $v_{ij} \in V_{\overline{x}y}$, at most six edges adjacent to the node can be directly identified by the labeling scheme itself. Denote $Adj(v_{ij})$ as an ordered list of those edges which are sequentially arranged in a circular fashion. Then it is said that the node v_{ij} can be moved into P_y for the interchange when the following two conditions are satisfied:

1. There is only one continuous subsequence of edges which are equal to xx in $Adj(v_{ij})$.

2. There is at least one edge in $Adj(v_{ij})$, which is equal to xy and does not lead to the node $v_{kl} \in V_{x\overline{y}}$.

Assume that initially all nodes of each cluster are connected in their physical topology. Condition 1 implies that the remaining nodes of P_x after removing v_{ij} still preserve the connectivity in their physical topology. Condition 2 implies that the removed node v_{ij} is also connected to its adjacent cluster P_y whose nodes are already connected in their physical topology. In the same way, the node v_{kl} must also satisfy the above two conditions so that it can be moved into P_x for the interchange. Therefore, the interchange of a pair of boundary nodes $v_{ij} \in V_{\overline{x}y}$ and $v_{kl} \in V_{x\overline{y}}$ is feasible iff both v_{ij} and v_{kl} satisfy Condition 1 before the interchange and Condition 2 after the interchange.

In the example of Figure 2(b), we consider interchanging $v_{22} \in P_2$ and $v_{44} \in P_1$. For node v_{22}, since the only one continuous subsequence $\{e_{12}, e_{11}, e_{21}, e_{32}\}$ in $Adj(v_{22}) = \{e_{23}, e_{12}, e_{11}, e_{21}, e_{32}, e_{33}\}$

	0	1	2	3	4	5	6	7	8
0	2	22	2	12	1				
1	22	12	12	12	11	11			
2	2	12	1	11	1	11	1		
3	22	22	12	12	12	11	11	11	
4	2	22	2	22	2	12	1	11	1
5	23	23	23	23	23	13	13	13	
6	3	33	3	33	3	33	3		
7	33	33	33	33	33	33			
8	3	33	3	33	3				

(a) after the interchange of (2,2) and (4,4.)

	0	1	2	3	4	5	6	7	8
0	2	22	2	12	1				
1	22	22	22	12	11	11			
2	2	22	2	12	1	11	1		
3	22	12	12	22	12	11	11	11	
4	2	12	1	12	2	12	1	11	1
5	23	13	13	23	23	13	13	13	
6	3	33	3	33	3	33	3		
7	33	33	33	33	33	33			
8	3	33	3	33	3				

(b) after the interchange of (4,2) and (4,4)

Figure 3: Interchanging The Boundary Node Elements

is equal to 22, Condition 1 holds. In addition, since $e_{23} = 12$ and it does not lead to v_{44}, Condition 2 also holds. Thus, v_{22} can be moved to P_1 for the interchange. At the same time, for node v_{44}, since the only one continuous subsequence $\{e_{45}, e_{34}\}$ in $Adj(v_{44}) = \{e_{34}, e_{33}, e_{43}, e_{53}, e_{54}, e_{45}\}$ is equal to 11, Condition 1 holds. In addition, since $e_{43} = 12$ and it does not lead to v_{22}, Condition 2 also holds. Thus, v_{44} can be moved to P_2 for the interchange. Hence, v_{22} and v_{44} are a feasible pair of interchanging nodes between P_1 and P_2. Figure 3(a) shows the resulting topology matrix **T** after the interchange. On the other hand, the nodes $v_{42} \in P_2$ and $v_{44} \in P_1$ cannot be interchanged, as depicted in Figure 3(b). The node v_{42} is isolated after the interchange due to the fact that no edge except e_{43} adjacent to v_{42} is equal to 12 in Figure 2(b).

It should be noted that the move of a boundary node $v_{ij} \in V_{\overline{x}y}$ into P_y is rather simple than the interchange because we need to check that only all nodes in P_x adjacent to v_{ij} are connected in their physical topology before the move.

3.2 Computing Gains

3.2.1 The Interchange

Assume that a pair of boundary nodes, $v_{ij} \in V_{\overline{x}y}$ and $v_{kl} \in V_{x\overline{y}}$, is feasible for the interchange. Define the internal cost of node v_{ij} with respect to P_x for the interchange to be

$$I_e(v_{ij}) = \sum_{v_{i'j'} \in P_x, v_{ij} \neq v_{i'j'}} (c_{ij;i'j'} + c_{i'j';ij}),$$

and the external cost of node v_{ij} with respect to P_y for the interchange to be

$$E_e(v_{ij}) = \sum_{v_{k'l'} \in P_y, v_{kl} \neq v_{k'l'}} (c_{ij;k'l'} + c_{k'l';ij}).$$

Similarly, we define $I_e(v_{kl})$ and $E_e(v_{kl})$ for the boundary node v_{kl}. If v_{ij} and v_{kl} are interchanged, then the gain g of the increase in cost is given by

$$g = (E_e(v_{ij}) + E_e(v_{kl})) - (I_e(v_{ij}) + I_e(v_{kl})).$$

3.2.2 The Move

Assume that a node $v_{ij} \in V_{\overline{x}y}$ is feasible for the move. Define the internal cost of node v_{ij} with respect to P_x for the move to be

$$I_m(v_{ij}) = \sum_{v_{i'j'} \in P_x, v_{ij} \neq v_{i'j'}} (c_{ij;i'j'} + c_{i'j';ij}),$$

and the external cost of node v_{ij} with respect to P_y for the move to be

$$E_m(v_{ij}) = \sum_{v_{k'l'} \in P_y} (c_{ij;k'l'} + c_{k'l';ij}).$$

If $v_{ij} \in P_x$ is moved into P_y, then the gain g of the increase in cost is given by

$$g = E_m(v_{kl}) - I_m(v_{ij}).$$

3.3 Heuristic.1

Given a pair of adjacent clusters P_x and P_y, Heuristic.1 interchanges only one feasible pair of boundary nodes with a positive maximum gain g between the two clusters. This process is repeated with an updated boundary until no feasible pair produces a positive gain. Let k be the number of feasible pairs interchanged. Then the the total gain with respect to the sum of the initial costs for the two clusters is $G_{xy} = \sum_{i=1}^{k} g_i$. Note that a pair of nodes interchanged in the previous step is not interchanged again at the next step because the positive maximum gain in the previous step becomes the minimum gain with the same negative value in the next step.

Heuristic.1(Π)
for every pair of adjacent clusters P_x and P_y in Π
    begin
        Determine the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
        forever
            begin
                Determine a set of feasible pairs
                if there is no feasible pair
                    break

```
            Compute gains for all feasible pair
            if there is no feasible pair with a positive gain
                break
            Choose a feasible pair $(v_{ij}, v_{kl})$ with the maximum gain
            Interchange $v_{ij}$ and $v_{kl}$
            Update the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
        end
    end
```

3.4 Heuristic.2

Once a set of feasible pairs of boundary nodes is determined from the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$, Heuristic.2 interchanges all feasible pairs with positive gains before updating the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$. Initially, all feasible pairs are unmarked. A feasible pair with the maximum positive gain is first interchanged and then all remaining unmarked feasible pairs which involve the boundary nodes of the interchanged pair are marked so that in the next step they cannot be considered again. After marking, the gains for all unmarked feasible pairs are computed again and then an unmarked feasible pair with the largest positive gain is next interchanged.

```
Heuristic.2(Π)
for every pair of adjacent clusters $P_x$ and $P_y$ in Π
    begin
        Determine the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
        forever
            begin
                Determine and unmark a set of feasible pairs
                if there is no feasible pair
                    break
                Compute gains for all feasible pairs
                while there are unmarked feasible pairs with positive gains
                    begin
                        Choose an unmarked feasible pair $(v_{ij}, v_{kl})$ with the largest gain
                        Interchange $v_{ij}$ and $v_{kl}$
                        Mark all unmarked feasible pairs involving $v_{ij}$ or $v_{kl}$
                        Compute gains for all unmarked feasible pairs
                    end
                Update the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
            end
    end
```

3.5 Heuristic.3

Heuristic.3 is based on the observation that infeasible pairs of boundary nodes might become feasible pairs after interchanging feasible pairs. For example, consider a pair of adjacent clusters P_1 and P_2 in the topology matrix of H_3, where $P_1 = \{v_{00}, v_{02}, v_{04}\}$ and $p_2 = \{v_{20}, v_{22}, v_{24}, v_{26}\}$. Then $V_{\overline{1}2} = P_1$ and $V_{1\overline{2}} = P_2$. A pair of boundary nodes (v_{02}, v_{22}) is infeasible for the interchange because v_{02} and v_{22} violate Condition 1. However, if a feasible pair (v_{04}, v_{20}) is interchanged, the infeasible pair (v_{02}, v_{22}) becomes a feasible pair for the next interchange.

The algorithm of Heuristic.3 is obtained by slightly modifying that of Heuristic.2. Before performing line 15 in Heuristic.2, infeasible pairs of boundary nodes which become feasible pairs due to the interchange of nodes v_{ij} and v_{kl} in line 13 are added to the set of unmarked feasible pairs.

3.6 Heuristic.4

While the previous three heuristics interchange boundary nodes between a pair of adjacent clusters, Heuristic.4 moves a boundary node in a cluster into the other cluster between a pair of adjacent clusters. Since the move of a boundary node between a pair of adjacent clusters changes their cluster sizes, the constraint on the cluster size is given by the minimum and maximum cluster sizes.

Given a pair of adjacent clusters, Heuristic.4 determines a feasible boundary node with a maximum positive gain for each cluster and compares them to determine a boundary node to be moved. Once the boundary node to be moved is determined, it is removed from its cluster and added to the other cluster as long as its move does not violate the constraint on the cluster size. This process is repeated until there is no feasible boundary node for the move or one of the clusters has the minimum or maximum cluster size.

Heuristic.4(Π)
for every pair of adjacent clusters P_x and P_y in Π
    begin
        Determine the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
        while the constraint on the cluster size is satisfied
            begin
                Determine a set of feasible nodes
                if there is no feasible node
                    break
                Compute gains for all feasible nodes
                if there is no feasible node with a positive gain
                    break
                Choose a feasible node v_{ij} with the maximum gain

```
            Move $v_{ij}$ into the other cluster
            Update the boundary node lists $V_{\overline{x}y}$ and $V_{x\overline{y}}$
        end
    end
```

3.7 Heuristic.5, Heuristic.6, and Heuristic.7

Heuristic.5, Heuristic.6, and Heuristic.7 are based on a repeated application of two-phase optimization. In the first phase, the heuristics use Heuristic.1, Heuristic.2, and Heuristic.3 for interchanging the boundary nodes between adjacent clusters, respectively. Their second phase uses Heuristic.4 for moving the boundary nodes between adjacent clusters to improve the gains obtained by their first phase if possible. The improvement by their second phase implies the change of the boundary nodes and the possibility that their first phase further improves their gains. Thus, the two-phase optimization is repeatedly applied until no improvement is possible by their second phase. The basic idea of the heuristics is that changing the cluster size by their second phase might overcome the limited improvement due to preserving the cluster sizes of the initial partition by their first phase.

3.8 Heuristic.8, Heuristic.9, and Heuristic.10

Heuristic.8, Heuristic.9, and Heuristic.10 are also based on a repeated application of two-phase optimization. However, in the first phase, the heuristics use Heuristic.4 for moving the boundary nodes between adjacent clusters. Their second phase uses respectively Heuristic.1, Heuristic.2, and Heuristic.3 for interchanging the boundary nodes between adjacent clusters to improve the gains obtained by their first phase if possible. The heuristics are derived from the fact that the limited gain of increase in cost due to the constraint on the cluster size in their first phase might be further improved by interchanging the boundary nodes without violating the constraint on the cluster size.

4 Experimental Testing and Analysis

We obtain the initial partition for experimental testing of the heuristics by two different methods: *random* and *centering*. The random partition is based on the topology matrix because it is only concerned with the geographical arrangement of base stations and the cluster size constraint. On the other hand, the centering partition is based on both the topology and relative cost matrices because it further considers the traffic pattern among base stations.

The centering partition of m clusters is achieved by a two-phase algorithm. In the first phase,

Heuristics	Random n=19	Random n=37	Random n=61	Random n=91	Centering n=19	Centering n=37	Centering n=61	Centering n=91	Total
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	17	6	2	1	17	8	1	2	54
5	84	48	47	37	88	76	56	50	486
6	84	51	43	37	86	80	70	45	496
7	85	51	47	33	85	71	70	45	487
8	70	59	39	41	76	62	57	49	453
9	70	62	41	40	76	63	60	52	464
10	70	61	43	38	75	62	61	52	462

Table 1: The Number of Times in 100 Trials a Heuristic Produced a Solution with Maximal Total Cost with Respect to Other Heuristics When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

the m center nodes on which traffics are concentrated are selected by using the relative cost matrix. Each center node forms an initial intermediate cluster. For every intermediate cluster, the second phase identifies the adjacent nodes of the cluster which are not involved in other clusters by using the topology matrix and chooses one of them by using the relative cost matrix, which produces a maximum increase in cost when it is involved in the cluster. Then the maximum node is added to the intermediate cluster. This process is repeated until all nodes are contained one of the m clusters.

In experimental testing, the 100 instances of the relative cost matrix were randomly generated for each of several values of n, where $n = 19, 37, 61,$ and 91 for H_3, H_4, H_5, and H_6, respectively. For each value of n, Heuristic.1 through Heuristic.10 were extensively tested on the 100 cost matrix instances for each of the four cases which are the combinations of the two methods for obtaining the initial partition, random or centering, and the number of clusters $m = 3$ or 4. The constraint on the cluster size is given by $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$.

Table 1 and Table 2 present the number of times a heuristic produces a solution that is maximal with respect to the solutions produced by the other heuristics when $m = 3$ and 4, respectively. The tables show that Heuristic.5 through Heuristic.10 which are the combinations of the techniques of interchanging or moving boundary nodes greatly outperform the other heuristics using only one of the techniques. This is due to the fact that although the cost between a pair of adjacent clusters cannot be improved by interchanging their boundary nodes, moving their boundary nodes might further improve the cost, which in turn changes the boundary between the adjacent clusters for

Heuristics	Random				Centering				Total
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91	
1	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0
4	8	2	2	2	7	2	2	1	26
5	43	46	41	33	73	71	58	52	417
6	43	45	43	42	73	72	61	52	431
7	64	53	56	45	84	71	63	60	496
8	68	59	34	27	80	74	66	41	449
9	68	59	36	27	80	75	71	45	461
10	68	61	37	29	80	76	66	43	460

Table 2: The Number of Times in 100 Trials a Heuristic Produced a Solution with Maximal Total Cost with Respect to Other Heuristics When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

possible interchanging at the next step, and vice versa. The tables also show that as the number of nodes n increases, the probability that a heuristic produces a maximal solution becomes lower.

It is interesting that Heuristic.1 through Heuristic.3 do not produce maximal solutions at all even though Heuristic.4 produces a small number of solutions being maximal. This is because interchanging boundary nodes does not change the number of nodes for each cluster in the initial partition and so less flexible than moving boundary nodes. Thus, unlike general graph partitioning problems, the algorithm which is only based on one of the techniques of interchanging or moving boundary nodes is no longer useful for our problem which additionally considers the underlying topology.

Table 3 and Table 4 present the percent of maximum difference in cost from the solution with the maximal cost, while Table 5 and Table 6 present the percent of average difference in cost from the solution with the maximal cost. The tables confirm the superiority of Heuristic.5 through Heuristic.10 with respect to the other hueristics. In general, the maximum differences for the centering partition of Heuristic.8 through Heuristic.10 are minimal with respect to the random partition of Heuristic.8 through Heuristic.10 and both the random and centering partitions of Heuristic.5 through Heuristic.7 with some exceptional test cases. The average differences for the centering partition of Heuristic.8 through Heuristic.10 are also minimal in general.

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	44.2	36.4	34.6	31.4	42.8	37.1	35.0	34.7
2	44.2	36.4	34.6	31.2	42.8	37.1	35.0	34.7
3	44.2	36.4	33.1	31.0	42.8	37.1	35.0	34.7
4	9.3	6.3	5.6	4.3	11.2	17.7	5.7	23.4
5	8.7	28.1	25.9	13.5	27.3	3.2	23.5	23.1
6	8.7	28.1	25.9	18.9	27.3	3.2	23.5	23.1
7	8.7	26.3	25.9	18.9	27.3	4.5	2.4	1.6
8	7.2	23.3	25.6	23.9	3.9	3.7	2.7	2.0
9	7.2	23.3	25.6	23.9	3.9	3.7	2.7	2.0
10	7.2	23.3	3.1	23.9	3.9	3.7	2.7	2.0

Table 3: The Maximum Difference in Total Cost From Maximal Solution When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	40.2	33.1	29.5	26.3	40.6	34.6	31.5	29.9
2	40.2	33.1	29.7	26.1	40.6	34.6	31.5	29.9
3	40.2	33.1	29.7	26.0	40.6	34.6	31.5	29.9
4	16.0	8.2	5.5	5.0	23.3	17.9	9.4	5.4
5	11.2	5.2	10.3	4.5	14.0	17.0	12.8	3.7
6	11.2	5.2	10.3	3.5	14.0	17.0	12.8	3.7
7	9.8	5.2	2.7	4.1	6.8	4.5	3.3	2.1
8	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9
9	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9
10	13.1	5.4	3.9	4.3	6.8	4.7	3.6	3.9

Table 4: The Maximum Difference in Total Cost From Maximal Solution When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	38.2	33.6	31.0	29.2	35.1	32.3	31.7	31.9
2	38.2	33.6	31.0	29.2	35.1	32.2	31.7	31.9
3	38.2	33.6	31.0	29.1	35.1	32.2	31.7	31.9
4	3.0	2.3	2.3	1.7	3.1	2.4	1.6	1.5
5	0.4	1.6	1.0	0.8	0.7	0.2	0.7	0.5
6	0.4	1.3	1.4	0.8	0.7	0.2	0.6	0.5
7	0.4	1.0	1.3	1.1	0.6	0.4	0.2	0.3
8	0.9	0.7	0.9	0.9	0.4	0.4	0.3	0.3
9	0.9	0.6	0.9	0.9	0.4	0.4	0.3	0.3
10	0.9	0.6	0.6	0.7	0.4	0.4	0.3	0.3

Table 5: The Average Difference in Total Cost From Maximal Solution When $m = 3$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

Heuristics	Random				Centering			
	n=19	n=37	n=61	n=91	n=19	n=37	n=61	n=91
1	31.3	29.2	27.5	24.2	31.8	29.3	26.1	25.4
2	31.3	29.2	27.5	24.2	31.8	29.3	26.1	25.5
3	31.3	29.2	27.4	24.2	31.8	29.3	26.1	25.4
4	5.9	3.3	2.8	2.1	5.2	3.2	2.2	2.0
5	2.3	0.7	0.7	0.7	1.1	0.7	0.5	0.4
6	2.3	0.6	0.7	0.5	1.1	0.7	0.5	0.4
7	0.9	0.5	0.4	0.4	0.5	0.5	0.3	0.3
8	0.9	0.5	0.7	0.6	0.6	0.5	0.2	0.5
9	0.9	0.5	0.8	0.6	0.6	0.4	0.2	0.5
10	0.9	0.5	0.7	0.6	0.6	0.4	0.2	0.5

Table 6: The Average Difference in Total Cost From Maximal Solution When $m = 4$ and $1/2 \times \lceil n/m \rceil \leq |P_i| \leq 2 \times \lceil n/m \rceil$

5 Conclusion

In cellular systems, a hexagonal mesh of n base stations may generate multiple types of traffic among themselves. We have considered the problem of finding a cover of disjoint clusters of base stations, so as to minimize the total communication cost for the entire system. The problem differs from general graph partitioning problems in that it additionally considers the underlying physical topology among base stations. We have developed several heuristics based on the combinations of the techniques of moving or interchanging the boundary nodes between adjacent clusters. These heuristics produce optimal partitions with respect to the initial partition obtained randomly or by centering. The heuristics are compared and shown to behave quite well through experimental testing and analysis.

References

[1] Baruch Awerbuch and David Peleg, "Concurrent Online Tracking of Mobile Users," *ACM SIGCOM'91*, pp. 221-233, 1991.

[2] Baruch Awerbuch and David Peleg, "Sparse Partitions," *1990 IEEE 31th Annual Symp. on Foundations of Computer Science*, pp. 503-513, 1990.

[3] Amotz Bar-Noy and Ilan Kessler, "Tracking Mobile Users in Wireless Communications Networks," *INFOCOM'93*, pp. 1232-1239, 1993.

[4] T. Imielinski and B. R. Badrinath, "Querying Locations in Wireless Environments," *The Third Workshop on Third Generation Wireless Information Networks*, April 1992.

[5] Michael R. Garey and David S. Johnson, *Computers and Interactability - A Guide to The Theory of NP-Completeness*, W.H. Freeman and Company, 1979.

[6] Oliver Goldschmidt and Dorit S. Hochbaum, "Polynomial Algorithm For The k-cut Problem," *29th Annual Symp. on Foundation of Computer Science*, pp. 444-451, 1988.

[7] E. Dahlhaus, D. S. Johnson, and C. H. Papadimitriou, "The Complexity of Multiway Cuts," *Proc. of The 24th Annual ACM Symp. on The Theory of Computing*, pp. 241-251, 1992.

[8] Brenda S. Baker, "Approximation Algorithms for NP-Complete Problems on Planar Graphs," *Proc. Symposium on Foundation of Computer Science*, pp. 265-273, 1983.

[9] Thang Nguyen Bui and Andrew Peck, "Partitioning Planar Graphs," *SIAM J. Comput.*, Vol. 21, No. 2, pp. 203-215, April 1992.

[10] Edward Minieka, *Optimization Algorithms for Networks and Graphs*, Marcel Dekker, Inc., 1978.

[11] L. Caccetta, "Graph Theory in Network Design and Analysis," *Recent Studies in Graph Theory*, ed. V. R. Kulli, Vishwa International Publications, 1989.

[12] Kyungshik Lim and Yann-Hang Lee, "Optimal Partitioning of Heterogeneous Traffic Sources in Highway Cellular Systems," *Personal, Indoor and Mobile Radio Communications PIMRC'94*, The Hague, The Netherlands, Sep. 1994.