Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 30 (2007) 1899–1912
computer communications
www.elsevier.com/locate/comcom

Analysis and algorithms design for the partition of large-scale adaptive mobile wireless networks

Bin Xiao [a,*], Jiannong Cao [a], Zili Shao [a], Qingfeng Zhuge [b], Edwin H.-M. Sha [b]

[a] Department of Computing, Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[b] Department of Computer Science, University of Texas at Dallas, Richardson, TX 75083, USA

Received 13 February 2007; accepted 23 February 2007
Available online 28 February 2007

Abstract

In a large-scale adaptive mobile wireless network, mobile units can communicate via either ad hoc or server-based communications. Ad hoc communication allows mobile units in close proximity to exchange messages directly. Server-based communication allows long distance contact between mobile units but must be supported by mobile servers. This paper investigates the partitioning problem as it applies to the assignment of mobile nodes, which contain mobile units in close proximity, to mobile servers under constraints of wireless transmission range and available bandwidth. This problem is even more difficult when the topologies of the mobile node connection graph and the mobile server network graph are dynamically changing. Given appropriate definitions for valid partitions in our framework, this paper shows the associated decision-based partition problems are NP-complete. In this paper, we propose assigning mobile nodes to mobile servers using efficient heuristic algorithms such that communication requirements among mobile nodes are successfully met by mobile servers. The simulation environment simulates a dynamically modified network topology of a wireless network consisting of roaming mobile nodes. The results show that proposed heuristic algorithms can yield effective assignments with a performance similar to that produced by exhaustive approaches.

Keywords: Partition; Adaptive mobile networks; NP-complete

1. Introduction

A large-scale adaptive mobile wireless network is composed of mobile servers and mobile units working together. Such a system [1,2] can be found for example on a battlefield, where mobile units such as tanks, airplanes and command centres are required to communicate in real-time. In some cases this would occur directly through point-to-point connections, but on most occasions communications must be supported by mobile servers. Because mobile units have a short transmission range, mobile servers can route messages between distant units. When a mobile unit is inserted, deleted or moved to another geographical area, its connection requirements will change dynamically. At the same time, the mobile server network is also likely to change, for example, by the addition or deletion of being a mobile server. The challenge for such an adaptive wireless communication system is then to design efficient algorithms for adaptively assigning mobile units to mobile servers.

A large-scale adaptive mobile wireless network is not the only type of wireless communication system to make use of both mobile servers and mobile units, although it is the type to incorporate both mobile servers and direct communications between mobile units. Cellular mobile networks [3,4] for example consist of mobile units and mobile servers (or base stations). There are no direct communications among mobile units and message exchange must be via a mobile server. Mobile ad hoc networks [5–7] differ from both cellular and adaptive systems in that they contain no mobile servers, requiring some mobile units to function as routers. In this approach, routing protocols select a

* Corresponding author. Tel.: +852 27667270.
E-mail addresses: csbxiao@comp.polyu.edu.hk (B. Xiao), csjcao@comp.polyu.edu.hk (J. Cao), cszlshao@comp.polyu.edu.hk (Z. Shao), qfzhuge@utdallas.edu (Q. Zhuge), edsha@utdallas.edu (E.H.-M. Sha).

0140-3664/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.02.016

mobile unit to act as a virtual mobile server in a group of mobile hosts [8–10]. The drawback of this approach is that it places a heavy computational burden on the selected unit that can create a potential system bottleneck [11–13]. Neither the cellular nor the mobile ad hoc network model can handle a dynamically changing large-scale wireless network well. In previous work [14–17], the authors discussed the combination of cellular wireless networks and mobile ad hoc wireless networks to produce a hybrid wireless network. In this paper, we investigate a mixed infrastructure of a wireless network that achieves scalability by incorporating direct communications and reliability by making use of mobile servers.

To represent the configuration of a large-scale adaptive mobile wireless network, we use two graphs, the *mobile node connection graph* and the *mobile server network graph*. The *mobile node connection graph* models the connection requirements of mobile nodes. Each mobile node represents a group of mobile units, within which mobile units can directly contact each other without the need for routing assistance from mobile servers. Communication between any two mobile nodes, however, must be sustained by mobile servers. The *mobile server network graph* represents the network topology of the mobile servers. If a direct connection does not exist between two mobile servers, message delivery is routed through intermediate mobile servers. The configuration of these two graphs change as the adaptive mobile network changes.

The problem of assigning mobile units to mobile servers is a problem of how to partition a mobile node graph given available resources in a mobile server network graph. In some parameter types, this partitioning problem is NP-complete [18]. This issue has been well addressed in previous work. The complexity of special cases of the graph partition problem is analyzed in [19]. [20] proposes a balanced bipartition heuristic algorithm to solve the min-cut problem in terms of network flow. An efficient partition can minimize the overall cost of a system [21], balance the workload in a wireless communication network [22], as well as maintain optimal network structure in a mobile-switch ATM network [23]. A successful partitioning assignment satisfies constraints in both mobile servers (e.g., bandwidth capacity) and mobile nodes (e.g., position location). An efficient partition aims to minimize the number of nodes that are not serviced in an adaptive wireless system given limited resources.

In this paper, we propose assigning mobile nodes to mobile servers using efficient heuristic algorithms. We first model a large-scale wireless network by using two graphs and formally define a valid partition for mapping a set of mobile nodes to a set of mobile servers. We then prove that the partitioning problem is NP-complete when we search for optimal solutions with given conditions. To support the maximum number of mobile nodes within a limited number of mobile servers, we developed three heuristic algorithms: *ANS*, *GREEDY*, *IGREEDY*. The *ANS* algorithm shows a means of distributed calculation of a partition while the *GREEDY*, *IGREEDY* algorithms need to acquire the system overall resources. We compared them with another two algorithms: *MINcut* and *MAXcv* that exhaustively search for optimal solutions. Simulations show that an initial partition generated by the *MINcut* algorithm can cover mobile nodes maximally in some cases. However, exhaustive algorithms are not suitable for a large wireless network consisting of a large number of mobile nodes and servers. Nevertheless, a dynamically changing wireless environment, such as mobile nodes moving around and mobile servers insertion/deletion etc., makes the performance of the *MINcut* algorithm close to the *GREEDY* and *IGREEDY* algorithms that have polynomial time running. The *IGREEDY* algorithm produced the best result of the three proposed heuristic algorithms, reducing the number of mobile nodes that were not serviced by any mobile servers by up to 55.2% compared with the *ANS* algorithm.

The rest of the paper is organized as follows. In Section 2 we introduce graph models to represent a large-scale adaptive mobile network. Section 3 gives definitions related to the partitioning problem. In Section 4, the partitioning problem is proved to be NP-complete under certain conditions. In Section 5 we present efficient algorithms for solving the partitioning problem. Section 6 describes the simulation model and discusses the simulation results. Section 7 offers our conclusions.

2. Adaptive mobile network models

In this section, we describe graph models for characterizing a large-scale wireless communication system, in which there may exist both mobile units and mobile servers. In some scenarios, such as on a battlefield or in an outdoor real-time exploration, the communication channel between two units must be maintained for a long time. The communication requirements of all of the mobile units can be captured using a mobile node connection graph. Mobile servers are able to provide long distance contact between two mobile units because they have longer transmission range and can behave as routers. A mobile server network graph is used to describe available resources that can sustain mobile units.

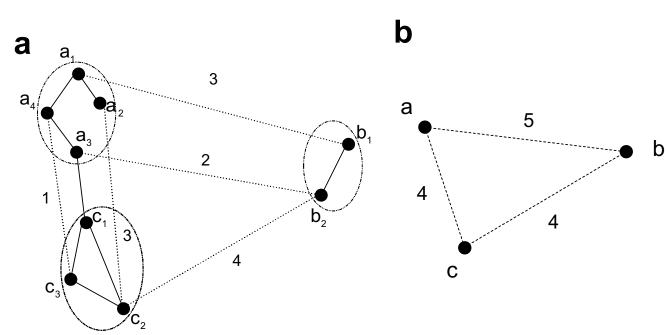


Fig. 1. (a) Wireless communication requirements among mobile units. (b) The simplified mobile node connection graph.

Fig. 1(a) provides an example of the communication requirements of a number of mobile units. Fig. 1(b) is a simplified mobile node connection graph. Each node in Fig. 1(a) represents a mobile unit. An edge connecting two mobile units shows that a communication requirement exists between them. Two closely neighboring mobile units can have direct contacts. We use a solid line to represent direct contact. Mobile units a_1 and a_2, a_3 and a_4, a_3 and c_1 etc. have direct communications and we use solid lines to connect them. A dotted line indicates that two mutually distant units require the routing service of a mobile server. A number beside a dotted line indicates the channel bandwidth requirement. As clusters contain mobile units in close proximity and these units can contact each other directly, connections within clusters are indicated by dotted lines. Fig. 1(b) simplifies Fig. 1(a) by representing each cluster as a mobile node, within which, it should be remembered, there is no need for the routing work of a mobile server. The new bandwidth requirement between any two mobile nodes in a simplified graph is the weighted sum of all original dotted connection lines between two clusters. The value of 4 for the link (a,c) in Fig. 1(b) comes from the addition of the weight (a_4,c_3) and (a_2,c_2), which is $1+3$. Based on the understanding of the mobile node connection graph in a wireless system, we have Definition 2.1.

Definition 2.1. A mobile node connection graph $G_N = \langle V_N, E_N, w, C \rangle$ is an edge-weighted undirected graph where V_N is a set of nodes representing mobile nodes, E_N is a set of edges representing communication links between mobile nodes, $w(e_i)$ indicates bandwidth requirement of edge e_i, and $C(v_i)$ represents a set of mobile server candidates for node $v_i \in V_N$.

Given a node $v \in V_N$, the set $C(v)$ represents those mobile servers that node v is able to directly connect to without route relay. $w(e_{uv})$ denotes the weight of an edge $e_{uv} \in E_\mathrm{N}$ that shows the bandwidth requirement for a successful communication between mobile nodes u and v. This communication link can either be supported within a mobile server, or through a routing path relayed by several mobile servers when $C(u) \cap C(v) = \emptyset$.

The scenario depicted in Fig. 2(a) shows a mobile server connection. Fig. 2(b) represents this scenario as a graph. We chose to use mobile servers because in a large wireless communication system the mobile server infrastructure can improve system performance tremendously, allowing, for example, long distance contact and broadband communication. In Fig. 2(a), each server reserves a bandwidth capacity for the communication of mobile nodes within its coverage, 10 Mbits/s for both server s_1 and server s_2. Another bandwidth exists between two mobile servers for their communication if they are inside the transmission range of each other. The available bandwidth is shown by a connecting edge and its weight. Thus, edge e_{12} connects mobile servers s_1 and s_2 and the assigned bandwidth is 17 Mbits/s. Definition 2.2 defines the mobile server network graph.

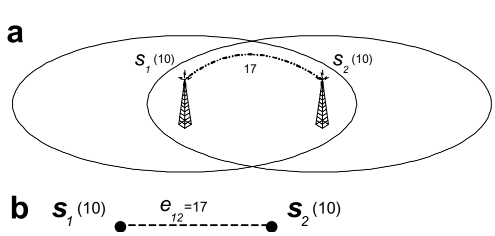


Fig. 2. (a) Available resources in mobile servers. (b) The characterized mobile server connection graph.

Definition 2.2. A mobile server network graph $G_S = \langle V_S, E_S, ce, cv \rangle$ is a node-weighted edge-weighted undirected graph where V_S is a set of nodes representing mobile servers, E_S is a set of edges representing communication links between mobile servers, the edge weight $ce(e_s)$ stands for the available bandwidth capacity between those two mobile servers connected by edge e_s, and the node weight $cv(v_s)$ stands for the bandwidth capacity of a mobile server v_s, which will provide service for a set of mapped mobile nodes that communicate with each other through mobile server v_s.

Given a graph G_S, let $MinCut(s_i,s_j)$ denote the minimum cut of all paths from s_i to s_j. We define $ce_\mathrm{server}(s_i,s_j)$ to be the available maximum bandwidth between mobile server s_i and s_j, which implies $ce_\mathrm{server}(s_i,s_j) = \sum_{e \in MinCut(s_i,s_j)} w(e)$.

3. Definitions of a valid partition

It is difficult to design a valid partition for mapping a set of mobile nodes to a set of mobile servers because a mobile server has a limited transmission range and limited bandwidth capacity. The design problem becomes even harder when both mobile servers and mobile nodes can freely move, as in an adaptive wireless network. The geographical topology of a mobile server network can change, and so can the topology of a mobile node connection graph. Topology changes may be the result of dynamic insertions or deletions of both mobile nodes and servers as well as being the result of their movement. For instance, in a defined mobile node connection graph model G_N, the mobility of a mobile node v leads to the dynamic change of $C(v)$. This may happen when v leaves an area covered by a particular group of servers.

System performance can be improved if a greater number of mobile nodes can obtain services in a dynamic wireless communication environment. This requires the valid partition of mobile nodes to mobile servers. This partition problem can be described as follows: Establish a mapping from a set of nodes in G_N to a set of nodes in G_S, such that the connection as well as bandwidth requirements for any two connected mobile nodes in G_N are satisfied, given the bandwidth capacity and location constraints on mobile servers in G_S. In some cases, only part of a mobile node can be mapped to a mobile server. In our formal description, a valid partition must conform to three constraints: the position constraint for a mobile node to its candidate mobile servers, the bandwidth constraint that a mobile server be able to provide services to communication requirements inside it, and the bandwidth constraint on a mobile server functioning as a relay router.

Definition 3.1. Given a valid partition that is a mapping from the node set V_N in a mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$ to the node set V_S in a mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, then a valid partition satisfies following constraints:

- If v_i ($v_i \in V_N$) is mapped to s_j ($s_j \in V_S$), s_j must be in $C(v_i)$.
- If any two mobile nodes connected by an edge e_i ($e_i \in E_N$) are mapped into the same mobile server s_j, then $cv(s_j) \geqslant \sum w(e_i)$.
- Let $W_{i,j}$ represent the total communication bandwidth requirement between mobile server s_i and s_j. Then, $ce_{\text{server}}(s_i, s_j) \geqslant W_{i,j}$.

If a node v is *mapped to* (or *covered by*) a server s, the server s is able to provide routing services to meet all communication requirements of v and this node v does not need to be mapped to another server. If an edge e is *mapped to* (or *covered by*) a server s after a partition, the server s should build a wireless channel that reserves a quantity of available bandwidth equivalent to the edge weight $w(e)$. We define some edge sets relevant to the computing of a valid partition in Definition 3.2 where the edge set $E_{\text{inside}}(s_i)$ consists of all edges in the node connection graph G_N that are totally covered by mobile server s_i, and the edge set $E_{\text{across}}(s_i,s_j)$ consists of all edges that are covered by both mobile server s_i and mobile server s_j.

Definition 3.2. Given a mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$, a mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, and a mapping

- let $E_{\text{inside}}(s_i)$ be a set of edges $e_{uv} \in E_N$ for $s_i \in V_S$, such that both mobile nodes u, v are mapped to mobile server s_i.
- let $E_{\text{across}}(s_i, s_j)$ be a set of edges $e_{xy} \in E_N$ for $s_i, s_j \in V_S$, where the communication channel between mobile node x and y is built through mobile servers s_i and s_j.

To define the amount of server bandwidth consumed after a mapping, we have Definition 3.3.

Definition 3.3. Given a mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$, a mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, and a mapping

- let $cv'(s_i)$ be the consumed bandwidth of server s_i and $cv'(s_i) = \sum w(e_j)$, $\forall e_j \in E_{\text{inside}}(s_i)$.
- let $ce'_{\text{server}}(s_i, s_j)$ be the consumed bandwidth between mobile server s_i and s_j in the graph G_S and $ce'_{\text{server}}(s_i, s_j) = \sum w(e_k)$, $\forall e_k \in E_{\text{across}}(s_i, s_j)$, $i \neq j$.
- let $ce'_{\text{server}}(s_i)$ be the consumed bandwidth of mobile server s_i and $ce'_{\text{server}}(s_i) = \sum_{\forall j \neq i} ce'_{\text{server}}(s_i, s_j)$.

In this paper, we investigate undirected graphs G_N and G_S, which implies that the minimum cut in a mapping has the property as $MinCut(s_i,s_j) = MinCut(s_j,s_i)$, and the amount of consume bandwidth has the property as $ce_{\text{server}}(s_i,s_j) = ce_{\text{server}}(s_j,s_i)$ and $ce'_{\text{server}}(s_i,s_j) = ce'_{\text{server}}(s_j,s_i)$. Following above definitions, Theorem 3.1 is given below.

Theorem 3.1. *Given a mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$ and a mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, a partition is valid:*

- *If v_i ($v_i \in V_N$) is mapped to s_i ($s_i \in V_S$), s_i must be in $C(v_i)$.*
- *$\forall s_i \in V_S$, $cv(s_i) \geqslant cv'(s_i)$ and $\forall i \neq j$, $ce_{\text{server}}(s_i, s_j) \geqslant ce'_{\text{server}}(s_i, s_j)$.*

Proof 1. By Definitions 3.1–3.3 □

Given a graph G_N, the partitioning problem is depicted as exploring a successful mapping of nodes in V_N to nodes in G_S, such that the sum of the weights of the edges covered inside a mobile server s_i is less than $cv(s_i)$ and the total bandwidth requirement between s_i and s_j is smaller than $ce_{\text{server}}(s_i,s_j)$. Let us consider a simple scenario in which two mobile servers are available and any node can be mapped into any mobile server. Fig. 2(b) shows a mobile server network graph G_S featuring two mobile servers $V_S = \{s_1, s_2\}$ and one connecting edge $E_S = \{e_{12}\}$. The system resources are $ce(e_{12}) = 17$, $cv(s_1) = cv(s_2) = 10$, which implies $ce_{\text{server}}(s_1, s_2) = 17$. Fig. 3(a) shows a mobile node connection graph G_N featuring seven nodes and $V_N = \{a, b, c, d, e, f, g\}$. The data on each edge stands for a communication bandwidth requirement between two mobile nodes. Fig. 3(b) shows a valid partition. According to Definition 3.2, we know $E_{\text{inside}}(s_1) = \{e_{ab}, e_{ad}, e_{ac}, e_{cd}\}$, $E_{\text{inside}}(s_2) = \{e_{ef}, e_{eg}, e_{fg}\}$, $E_{\text{across}}(s_1,s_2) = \{e_{be}, e_{bf}, e_{df}, e_{cg}\}$. Furthermore, $cv'(s_1) = w(e_{ab}) + w(e_{ad}) + w(e_{ac}) + w(e_{cd}) = 10$, $cv'(s_2) = w(e_{ef}) + w(e_{eg}) + w(e_{fg}) = 10$, $ce'_{\text{server}}(s_1, s_2) = w(e_{be}) + w(e_{bf}) + w(e_{df}) + w(e_{cg}) = 17$. In this example, the partition meets $cv(s_1) \geqslant cv'(s_1)$, $cv(s_2) \geqslant cv'(s_2)$ and $ce_{\text{server}}(s_1, s_2) \geqslant ce'_{\text{server}}(s_1, s_2)$. The valid partition complies with Theorem 3.1. It happens that there is no more bandwidth left after the mapping under the given conditions.

4. Partitioning problems – NP-complete

The partitioning problem can vary according to different objectives. In this section, we pose three questions. We prove that it is NP-hard to obtain a valid partition for these subinstances.

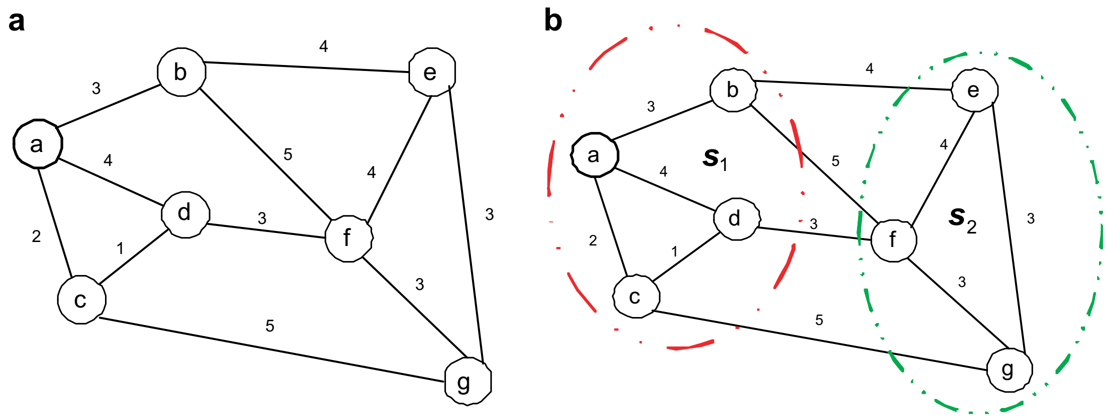


Fig. 3. (a) Graph G_N with seven mobile nodes. (b) A valid mapping from G_N to G_S in Fig. 2(b).

INSTANCE. Mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$, mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, $|V_S| = n$, nonnegative integer K.

- *Question 1.* Can we partition V_N into n disjoint sets $V_{S_1}, \ldots, V_{S_n}$ such that $\forall i$, mobile nodes in the set V_{S_i} are covered by mobile server s_i, $cv(s_i) - cv'(s_i)$ is at least K and $\forall i \neq j$, $ce_{\text{server}}(s_i, s_j) \geqslant ce'_{\text{server}}(s_i, s_j)$?
- *Question 2.* Can we partition V_N into n disjoint sets $V_{S_1}, \ldots, V_{S_n}$ such that $\forall i$, mobile nodes in the set V_{S_i} are covered by mobile server s_i, $\frac{1}{n}\sum_{i=1}^{n} cv'(s_i)$ is at most K, $cv(s_i) \geqslant cv'(s_i)$ and $\forall i \neq j$, $ce_{\text{server}}(s_i, s_j) \geqslant ce'_{\text{server}}(s_i, s_j)$?
- *Question 3.* Can we partition V_N into n disjoint sets $V_{S_1}, \ldots, V_{S_n}$ such that $\forall i$, mobile nodes in the set V_{S_i} are covered by mobile server s_i, $\sum_{i=1}^{n}\sum_{j>i}^{n} ce'_{\text{server}}(s_i, s_j)$ is at most K, $cv(s_i) \geqslant cv'(s_i)$ and $\forall i \neq j$, $ce_{\text{server}}(s_i, s_j) \geqslant ce'_{\text{server}}(s_i, s_j)$?

Given the instance, Question 1 tries to maintain a left communication bandwidth of K inside any mobile server. Question 2 seeks the lowest total bandwidth consumption of all mobile servers. Question 3 aims to minimize the total bandwidth used for the routing through all mobile servers. We will show that to find a solution for any of these questions is *NP-hard*.

Theorem 4.1. *The problem described in Question* 1 *is NP-complete.*

Proof 2.

1. The problem is NP. Given a partition we can easily verify it in polynomial time.
2. We show that the well-known *Bin Packing* [18] problem can be transformed into an instance of *Question* 1. The *Bin Packing* [18] problem can be defined as follows:*INSTANCE*: A finite set U of m items; for each item u_i $(1 \leqslant i \leqslant m)$ in U a positive integer size $s(u_i)$; positive integers B (called the *bin capacity*) and $n \leqslant m$.*QUESTION*: Can U be partitioned into n disjoint sets $U_1, \ldots, U_n$ such that for each U_i $(1 \leqslant i \leqslant n)$ the total sum of the sizes of the items in U_i does not exceed B?
3. Given an instance of the *Bin Packing* problem, we build an instance of *Question* 1 as follows: Let the mobile server number be n and $K = 0$. Let $cv(s_i) = B$ $(1 \leqslant i \leqslant n)$ and the weight of ce for each mobile server be 0, which means that there are no connections among mobile servers. The construction of graph G_S is complete.Given each $u_i \in U$, we draw an edge with weight $s(u_i)$. The two nodes connected by the edge are named v_{2i} and v_{2i+1}, i.e., $V_N = \{v_{2i}, v_{2i+1} | \forall u_i \in U,\ i = 1, \ldots, m\}$ and $|V_N| = 2 \cdot |U| = 2m$. We draw edges, of weight 0, that connect node v_{2i} and every other nodes except node v_{2i+1}. Thus graph G_N is a complete graph and the weight of each edge is either in the set U (if the edge connects nodes v_{2i} and v_{2i+1}) or 0. Suppose that each node in V_N can be covered by any mobile server. The construction of graph G_N is complete. The building of the instance of *Question* 1 is done in polynomial time. Because a valid mapping has to meet the bandwidth requirements and because the weight of ce for each mobile server is 0, an edge having its weight in the set U cannot be the edge of two nodes covered by two different mobile servers. Thus, a partition of node set V_N in graph G_N into n node sets from V_{S_1} to V_{S_n} guarantees that all edges having weights in the set U are covered inside mobile servers. Let P be the partition that contains these n node sets and $P = \{V_{S_1}, \ldots, V_{S_n}\}$. Given each u_i $(1 \leqslant i \leqslant m)$, its corresponding edge (v_{2i}, v_{2i+1}) in G_N should be exactly in one node set in P. From V_{S_i} we derive a node set T_i such that T_i contains a node u_j if and only if $(v_{2i}, v_{2i+1}) \in V_{S_i}$. Let $Q = \{T_1, T_2, \ldots, T_n\}$ and Q is a partition of U into n disjoint sets. Because we define $K = 0$ and $cv(s_i) = B$, the partition P is satisfiable to *Question* 1 if and only if Q is satisfiable to the *Bin Packing* problem. □

Fig. 3(b) provides a solution to Question 1 given the partitioning requirements $K = 0$ and $n = 2$. When $ce(e_{12}) = 17$, $cv(s_1) = cv(s_2) = 10$, we reach the conclusion that $cv(s_i) - cv'(s_i)$ is at least 0 and $ce_{\text{server}}(s_1, s_2) = ce'_{\text{server}}(s_1, s_2) = 17$ $(ce_{\text{server}}(s_1, s_2) \geqslant ce'_{\text{server}}\ (s, s_2))$ for each mobile server.

Theorem 4.2. *The problem described in Question* 2 *is NP-complete.*

Proof 3.

1. The problem is NP. Given a partition we can easily verify it in polynomial time.
2. We show that a known NP-complete problem-*MAX CUT* problem in [18] can be transformed into an instance of *Question* 2. The *MAX CUT* problem is defined 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. Given an instance of the *MAX CUT* problem, we build an instance of *Question* 2 as follows: Let the mobile server number be $n=2$. Suppose that two mobile servers are s_1 and s_2 ($V_S=\{s_1,s_2\}$) with a connecting edge e_{12} ($E_S=\{e_{12}\}$). From graph *G*, let $J=\sum_{\forall e\in E'}u(e)$ and be the weight of *cv*,*ce* for each mobile server, i.e., $cv(s_1)=cv(s_2)=ce(e_{12})=J$. Let graph G_N be isomorphic to graph *G*, that is $E=E'$, $V_N=V'$, and $\forall e\in E'$, its corresponding edge in *E* has the same weight $w(e)=u(e)$. We assume that each node in V_N can be covered by both mobile server s_1 and mobile server s_2. Such assumption defines *C* for graph G_N. Graphs G_N and G_S are completely constructed. Let $K=\frac{1}{2}(J-M)$. The building of the instance of *Question* 2 is done in polynomial time.
   If we solve the problem for *Question* 2, we can find a solution for the *MAX CUT* problem. Since the mobile server number is two, we have $ce_{\text{server}}(s_1,s_2)=ce(e_{12})=J$. A cut splits graph G_N into two parts. Each part is covered by one mobile server. This means that the sum of the weights of the edges with end nodes in different parts is $ce'_{\text{server}}(s_1)=ce'_{\text{server}}(s_2)$, which is the value of a cut. Since $J=\sum w(e)$, $\forall e\in E$ in graph G_N, *J* is equal to $cv'(s_1)+cv'(s_2)+ce'_{\text{server}}(s_1)$ and is a fixed value. If a partition satisfies $ce'_{\text{server}}(s_1)\geqslant M$, then $\frac{1}{2}(cv'(s_1)+\ cv'(s_2))\leqslant\frac{1}{2}(J-M)=K$. Thus given that each node can be covered by both s_1 and s_2, if we have a solution (a cut) for *Question* 2, then $\frac{1}{2}\sum_{i=1}^{2}cv\prime(s_i)\leqslant K$. This implies that the cut of the partition is at least *M*, which is the solution for the *MAX CUT* NP-complete problem. □

Fig. 4(a) shows the maximum cut partition of the graph in Fig. 3(a) assuming that every mobile node locates within the transmission range of each mobile server. Given the maximum cut value $M=23$, we have an answer for Question 2 with $K=\frac{1}{2}\{cv'(s_1)+cv'(s_2)\}=\frac{1}{2}\{10+4\}=7$.

Theorem 4.3. *The problem described in Question* 3 *is NP-complete.*

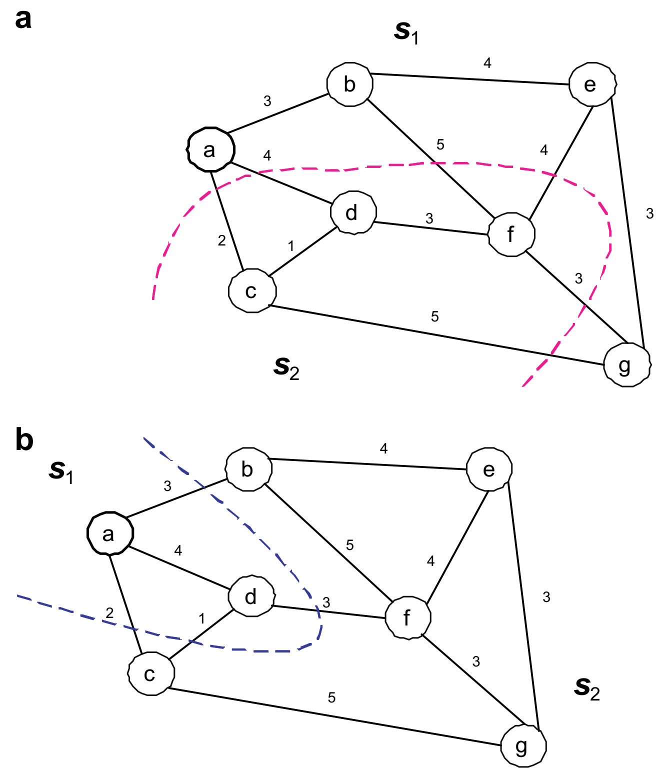


Fig. 4. (a) The maximum cut for a graph partition. (b) The minimum cut when mobile node *d* is assigned to mobile server s_1 and mobile node *f* is assigned to mobile server s_2.

Proof 4.

1. The problem is NP. Given a partition we can easily verify it in polynomial time.
2. We show that a known NP-complete problem (*MINIMUM CUT INTO BOUNDED SETS* problem in [18]) can be transformed into an instance of *Question* 3. The *MINIMUM CUT INTO BOUNDED SETS* problem is defined as follows:*INSTANCE.* Graph $G=(V',E')$, weight $u(e)\in Z^+$ for each $e\in E'$, specified vertices $s,t\in V$, positive integer $B\leqslant|V|$, positive integer *M*.*QUESTION.* Is there a partition of V' into disjoint sets V_1 and V_2 such that $s\in V_1,t\in V_2$, $|V_1—\leqslant B$, $—V_2|\leqslant B$, 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. Given an instance of the *MINIMUM CUT INTO BOUNDED SETS* problem, we build an instance of *Question* 3 as follows: Let the mobile server number be $n=2$. Suppose that two mobile servers are s_1 and s_2 ($V_S=\{s_1,s_2\}$) with a connecting edge e_{12} ($E_S=\{e_{12}\}$). From graph *G*, let $J=\sum_{\forall e\in E'}u(e)$ and be the weight of *cv*,*ce* for each mobile server, i.e., $cv(s_1)=cv(s_2)=ce(e_{12})=J$. Let graph G_N be isomorphic to graph *G*, that is $E=E'$, $V_N=V'$, and $\forall e\in E'$, its corresponding edge in *E* has the same weight $w(e)=u(e)$. We build *C* of graph G_N such that each mobile server can only cover *B* number of nodes. These *B* nodes are randomly

selected except that node s is only covered by mobile server s_1 while node t is only covered by mobile server s_2. Graph G_N and G_S are completely constructed. Let $K = M$. The building of the instance of *Question* 3 is done in polynomial time. If we solve the problem for *Question* 3, we can find a solution for the *MINIMUM CUT INTO BOUNDED SETS* problem. Since there are only two mobile servers, $ce'_{\text{server}}(s_1, s_2)$ denotes the capacity of a cut. The capacity less than or equal to K is an answer for Question 3. If we have a solution to Question 3, we find a partition that makes $ce'_{\text{server}}(s_1, s_2) = K$. This is also a partition with a capacity no more than $M = K$ for the *MINIMUM CUT INTO BOUNDED SETS* problem and the covered nodes inside each part is less than or equal to B. □

We show a solution for the minimum cut problem in Fig. 4(b) given the graph G_N in Fig. 3(a). In this example, d is assigned to mobile server s_1 and f to mobile server s_2. The minimum cut has the capacity of 9 and provides a valid partition to Question 3 with $K = 9$.

5. Algorithms

In this section, we present heuristic and exhaustive algorithms for solving the partitioning problem of mapping a set of nodes (mobile nodes) to another set of nodes (mobile servers). This problem under given graph constraints is an NP-complete problem. Thus, it is possible to generate an optimal solution by exhaustively exploring all possible partitions, such as the presented *MINcut* and *MAXcv* algorithms, in a small system that has fewer mobile nodes and servers. However, in the real world, an exhaustive approach would not be a practical approach to the partitioning problem, which may contain a large number of communication units. In a large system, we show three heuristic algorithms, i.e., *ANS*, *GREEDY*, and *IGREEDY*, to obtain valid mappings. All three algorithms seek to cover more mobile nodes within the constraints of limited resources. A better-designed algorithm would compute a valid partition with a lower overhead yet at the same time would provide coverage for a large number of mobile nodes even in a dynamic environment. To help in understanding, we provide an example showing how all proposed algorithms can be applied to the partitioning problem in a wireless network environment. Three heuristic algorithms, *ANS*, *GREEDY*, and *IGREEDY* can be applied in a large-scale adaptive wireless network.

- *ANS*. Assign a mobile Node to an available Server.
- *GREEDY*. Assign a mobile node to the server that has the most available bandwidth.
- *IGREEDY*. Assign a mobile node to a server to maximally maintain the system's overall available bandwidth.

Given a small wireless system, exhaustive approaches can be applied as follows.

- *MINcut*. Compare all possible partitions and select the one that has the maximum number of mobile nodes covered. If there is a tie, the partition that generates the minimum cut value will be chosen first.
- *MAXcv*. Compare all possible partitions and select the one that has the maximum number of mobile nodes covered. If there is a tie, the partition that maximizes the minimum value of $cv(s_i) - cv'(s_i)$ among all mobile servers will be chosen first.

The *MINcut* approach provides a solution to Question 3 while the *MAXcv* approach provides a solution to Question 1 (Questions 1 and 3 are posed in Section 3). The *MINcut* approach seeks to accommodate the largest number of mobile nodes that acquire full support in a wireless system. It also emphasizes the minimum cut value of a partition. The *MAXcv* approach, however, search for a solution to maximize the value of $cv(s_i) - cv'(s_i)$ of each mobile server. Note that we do not propose any exhaustive approaches to solve Question 2 because when the overall consumed bandwidth of cv' is small in a partition, the consumed bandwidth ce' would be large and we assume that ce bandwidth resources are more stringent.

5.1. ANS algorithm

Given a mobile node connection graph $G_N = \langle V_N, E_N, w, C\rangle$ and a mobile server network graph $G_S = \langle V_S, E_S, ce, cv\rangle$, let node v_i be in the transmission range of k mobile servers from s_{i_1} to s_{i_k}, which means $C(v_i) = \{s_{i_1}, s_{i_2}, \ldots, s_{i_k}\}$. The *ANS* algorithm is described as follows and every node itself can acquire a network access as in the *Partition* step:

1. *Initialization.* Generate a node set V_{all}, which contains all nodes in G_N ($V_{\text{all}} = V_N$).
2. *Partition.* Arbitrarily remove a node v_i from node set V_{all}. From $C(v_i)$ we randomly select a server s_{i_1}. Node v_i is assigned to mobile server s_{i_1} when the communication requirements of node v_i are all met. Otherwise, we continue to test other available mobile servers in the set $C(v_i)$ until node v_i is covered. If none of them satisfies requirements of v_i, node v_i can not be assigned to any mobile servers.
3. If $V_{\text{all}} \neq \emptyset$, go to step 2.

Assuming that $n = |V_N|$ and $s = |V_S|$, let the maximum edge degree be e_d in graph G_N. Then the computational complexity for *ANS* algorithm is $O(n \cdot s \cdot e_d)$. The asymptotic complexity of assigning node v_i in step 2 is $O(s \cdot e_d)$ because there are at most s mobile servers tested for node v_i. In each test, all connecting edges of v_i are investigated and there are at most e_d edges. The *ANS* algorithm ends after n iterations since n nodes must be removed from V_{all}. Its computational complexity is $O(n \cdot s \cdot e_d)$.

5.2. GREEDY algorithm

The difference between algorithm *GREEDY* and *ANS* is in Step 2. In the *GREEDY* algorithm, Step 2 is redefined as follows.

Partition. Arbitrarily remove a node v_i from the node set V_{all}. From all candidate mobile servers in $C(v_i)$, we assign node v_i to the mobile server that has the most free bandwidth. If the assignment satisfies all bandwidth requirements from node v_i, go to Step 3. Otherwise, we continue to test all other non-tested mobile servers in the set $C(v_i)$, in descending order starting from the one that has the maximum free bandwidth until node v_i is covered. If none of them satisfies requirements of v_i, node v_i cannot be assigned to any mobile server and thus is not covered.

The free bandwidth of a mobile server s_i is calculated by $\gamma * (cv^{(}s_i) - cv'(s_i)) + (1-\gamma) * (\sum_{\forall j \neq i} ce_{\text{server}}(s_i, s_j) - ce'_{\text{server}}(s_i))$, where $0 \leqslant \gamma \leqslant 1$ and can be adjusted during the mobile node assignment process. The computational complexity of the *GREEDY* algorithm is the same as that of the *ANS*, $O(n \cdot s \cdot e_d)$.

5.3. IGREEDY algorithm

We improve the *GREEDY* algorithm to the *IGREEDY* algorithm. The improvement is achieved through the mobile node assignment sequence. Given a node v_i, $|C(v_i)|$ shows how many mobile server candidates are available. Instead using an arbitrary mobile node assignment sequence as in the *ANS* and *GREEDY* algorithms, we first assign a node whose $|C(v_i)|$ is small. Let $Node(i)$ $(0 \leqslant i \leqslant |V_S|)$ be a node set which is initially reset to be empty. The *IGREEDY* algorithm can be described as follows.

1. *Initialization.* Generate a node set V_{all}, which contains all nodes in G_N $(V_{all} = V_N)$. Let $Level = 0$, and $s = |V_S|$.
2. Remove all mobile nodes $v_i \in V_N$ whose $|C(v_i)| = Level$ from V_{all}. Add them to the node set $Node(Level)$.
3. From node set $Node(Level)$, remove node v_i such that if we validly assign it to a mobile server, the new system cost $\sum_{i=1}^{s}(cv\prime(s_i) + ce'_{\text{server}}(s_i))$ increases minimally.
4. If $Node(Level) \neq \emptyset$, go to Step 3. Otherwise, $Level = Level + 1$.
5. If $V_{all} \neq \emptyset$, go to Step 2.

Note that in Step 3 if node v_i cannot be assigned to any mobile servers, this node is directly deleted from $Node(Level)$. Among all nodes in the node set $Node(Level)$, the node that incurs the smallest increment to the system cost $\sum_{i=1}^{s}(cv\prime(s_i) + ce'_{\text{server}}(s_i))$ will be assigned to a mobile server first. If there are two or more mobile servers that can cover v_i and incur the same smallest increment to the system cost, we arbitrarily choose one. The next node to be assigned follows the same principle of minimally increasing the system cost.

The *IGREEDY* algorithm has the computational complexity of $O(n^2 \cdot s \cdot e_d)$. Let $e = |E_N|$. The computational complexity of calculating the system cost $(\sum_{i=1}^{s}(cv'(s_i) + ce'_{\text{server}}(s_i)))$ is no more than the complexity of scanning all edges that have been covered by mobile servers during the mapping process. Thus, the calculation time of system cost is $O(e)$. In Step 3, finding the node that creates the new smallest system cost requires time $O(|Node(Level)| \cdot s \cdot e)$. This is because each node in the node set $Node(Level)$ requires the computation time $O(s \cdot e)$ to find its smallest system cost since it can be assigned to one of s candidate mobile servers and each assignment requires a system cost calculation $O(e)$. Searching for the smallest cost in the node set $Node(Level)$ requires another time $O(|Node(Level)|)$. Therefore, the computational complexity in Step 3 is $O(|Node(Level)| \cdot s \cdot e_d)$ to remove a node from the node set $Node(Level)$. In total, —$Node(Level)$— number of nodes must be removed, which implies that the computation time from Step 3 to Step 4 is $O(|Node(Level)|^2 \cdot s \cdot e_d)$. In the *IGREEDY* algorithm, n nodes are removed from the node set V_{all} and the total computational complexity becomes $O((\sum_{Level=0}^{s} | Node(Level)|^2) \cdot s \cdot e_d) \leqslant O(n^2 \cdot s \cdot e_d)$ $(\because \sum_{Level=0}^{s} | Node(Level) |= n)$.

6. Performance evaluation

In this section, we show the performance evaluation of the five proposed algorithms. The algorithms were measured in two systems, the small system and the large system. It is possible to test all five algorithms in a small system that has fewer mobile nodes and mobile servers, including *MINcut* or *MAXcv* algorithms using exhaustive approaches. To derive a valid partition in a large-scale system, it is practical to apply the heuristic algorithms *ANS*, *GREEDY* and *IGREEDY*. An effective partitioning algorithm should have more mobile nodes serviced in a dynamic wireless environment. Through simulations, we analyze the effectiveness of every algorithm in terms of how many mobile nodes are covered within an interval.

6.1. Simulation systems

To test the proposed algorithms we used two types of simulation environments, the *small system* and the *large system*. In a simulation environment with fewer mobile nodes (e.g., a small system), we evaluated all five algorithms. The heuristic algorithms *ANS*, *GREEDY* and *IGREEDY* have polynomial complexities whereas *MINcut* and *MAXcv* exhaustively search for an optimal solution. Thus, in a larger system with more mobile nodes and servers it becomes impossible to quickly deliver a partition using *MINcut* and *MAXcv*. Since the partitioning problem is NP-complete (Section 4), these two methods would require a huge amount of time to generate an optimal solution by exhaustive searching. The two simulation environments are described in detail in the following.

- *Small system.* Inside a small system, two mobile servers had fixed centers and could communicate with each other within a transmission radius of 50 meters. There were 10 mobile nodes, which were randomly scattered in a rectangle area of 135 (meters) ∗ 80 (meters).
- *Large system.* A large system was composed of $30 \sim 100$ mobile nodes and $5 \sim 11$ mobile servers. All nodes were randomly scattered in a rectangle area by 150 (meters) ∗ 150 (meters), while the centers of all mobile servers were uniformly scattered in a small rectangle area by 110 (meters)∗ 110 (meters). The transmission radius of a mobile server was fixed at 50 m. Edge connections among mobile nodes were arbitrarily chosen and edge weights were randomly given in the range [1,6] (meters).

Both systems shared some common parameters. Each mobile server had the same bandwidth capacity *cv*. The configured *cv* and *ce* values could vary in simulations to observe their impact. The center positions of mobile servers remained fixed during experiments while mobile nodes were able to move freely. If a node moved away from its current mobile server, it needed to find a new one to continue its service. During simulations, we randomly generated new wireless communication systems in an iteration and each tested iteration lasted for 10,000 time slots. Each mobile node constantly checked its state and could request service provided from an available close mobile server during one iteration.

The performance of an algorithm in generating a partition is measured by how many nodes are not covered during a simulation interval. A good partition resulting from an algorithm should have a larger number of mobile nodes that have their communication requirements supported. In the following, we applied all 5 algorithms, i.e., *ANS*, *GREEDY*, *IGREEDY*, *MINcut* and *MAXcv*, in a small wireless communication system but only the first 3 in a large system. Given a newly generated network topology in one iteration, every algorithm could yield a distinct partition result. Note that when mobile nodes are moving, some nodes may become isolated without service support because they roam away from their current service areas. The initial partition of resources will definitely impact the future coverage of such roaming nodes. A good initial partition in a wireless system where mobile nodes can move freely should have fewer nodes that are not covered by any mobile server.

6.2. Small system

In this section, we first show an example for the partition in a small system and demonstrate that how previously proposed five algorithms are applied. Then we compare the performance of different partition algorithms in a small system in detail.

Fig. 5 illustrates a small system in which ten mobile nodes are arbitrarily distributed in a rectangle area by 135 (meters) * 80 (meters). Routing services are provided by two mobile servers a and b. The edge weight of two mobile nodes shows the bandwidth requirement between them. The total edge weight in the mobile node connection graph is 41. Two mobile servers have the same transmission radius of 50 meters and they are in each other's transmission range. Let two mobile servers have the same bandwidth capacity with $cv = 16$ and $ce(e_{ab}) = 17$. In a valid mapping, if a mobile server covers an edge e, its cv' value is increased by the edge weight $w(e)$. If two nodes x, y with a connecting edge e_{xy} are mapped into different mobile servers a and b, the value of $ce'(a)$ and $ce'(b)$ are increased by $w(e_{xy})$. Otherwise, at least one node will not be covered by any mobile server.

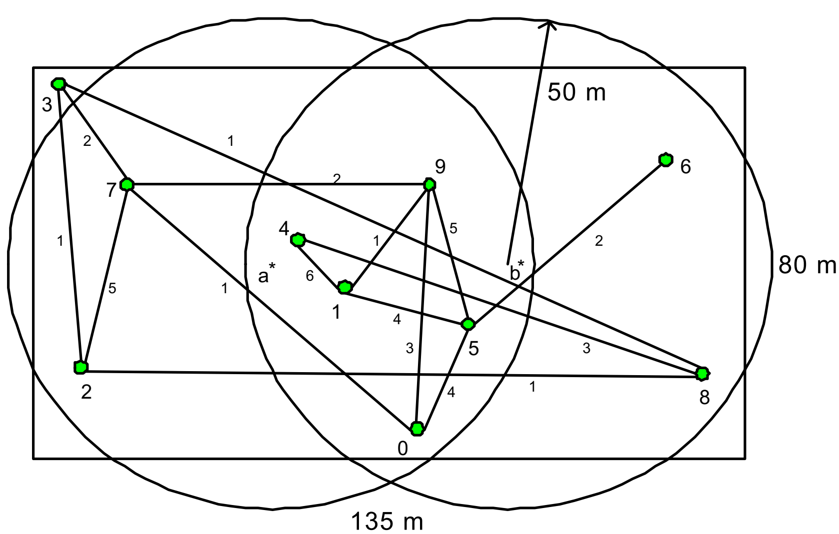


Fig. 5. Ten mobile nodes arbitrarily distributed within a rectangle.

Table 1 shows the partition results using five different algorithms. Each algorithm generates a distinct solution. As can be seen in the rightmost column, mobile node 3 is out of the transmission range of any mobile server and cannot be covered by any method. Column ce' shows the capacity consumption of mobile servers for all five algorithms. In terms of the used cv' and ce', *MINcut* has the smallest ce' cost (11) while *MAXcv* has the largest unused bandwidth ($cv - cv' = 6$) left for both mobile servers. This is consistent with their approaches. Partitioning results show that the *MINcut*, *MAXcv* and *GREEDY* algorithms have an initial better mapping of mobile nodes to servers with only one node not covered.

Table 2 shows the simulation results for the five algorithms where the data illustrates the average number of mobile nodes not covered per iteration. We get these results by running simulations in one, five, and ten iterations. The result from a simulation with more iterations can accurately show the impact from a partitioning algorithm. Under constraints of transmission range and available bandwidth of each mobile server, the *ANS* algorithm yielded the worst result while *MINcut* had the best to provide coverage of more mobile nodes. In one iteration, algorithms *MINcut*, *MAXcv*, *GREEDY* and *IGREEDY* had the same result (6,061) because they produced the same initial partition with given graphs. When we conducted simulations in more iterations, they failed to cover a similar number of nodes. This implies that in a large-scale wireless communication system *GREEDY* and *IGREEDY* algorithms can be efficient partitioning techniques to replace exhaustively searching methods, such as *MINcut* and *MAXcv*. The simulation result of the *IGREEDY* algorithm showed that only 1.1 nodes had not been covered in one time slot during 10 iterations. We attribute this to the bandwidth and transmission range constraints of mobile servers.

Table 1
Partition results for five algorithms

Algorithms	Mobile server *a*		Mobile server *b*		ce'	Nodes not covered
	Covered nodes	cv'	Covered nodes	cv'		
ANS	0,1,2,4,5	14	6,8,9	0	15	3,7
MINcut	0,2,7,9	11	1,4,5,6,8	15	11	3
MAXcv	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 2
Average number of mobile nodes not covered in a small system per iteration

Algorithms	*ANS*	*MINcut*	*MAXcv*	*GREEDY*	*IGREEDY*
1 iteration	16,061	6061	6061	6061	6061
5 iterations	12,757	6601	9616	10,185	8807
10 iterations	14,246	9548	11,055	11,321	11,020

6.3. Large system

In a large system, we compare the performance of the *ANS*, *GREEDY* and *IGREEDY* algorithms and evaluate the impact of different parameters in the system configuration. The parameters consist of the number of mobile nodes $|V_N|$, the number of servers $|V_S|$, the node weight ratio α_{cv} that is proportional to $\frac{\sum cv(v_s)}{\sum w(e_i)}$, and the edge weight ratio α_{ce} that is proportional to $\frac{\sum ce(e_s)}{\sum w(e_i)}$. Fig. 6 shows the topology of a generated large system that contains a randomly created mobile node connection graph with 50 nodes and a mobile server network graph with 10 servers (represented by dark filled rectangles). In a large simulated system, it is possible that some mobile servers cannot directly communicate with each other. To successfully build a channel between two remote mobile servers, a path could be constructed in which other mobile servers functioned as intermediate relay routers. All mobile servers along a constructed path would have an increased ce' value by the weight of a supported link. During the simulation for the *GREEDY* and *IGREEDY* methods, we find the shortest path between two mobile servers in the mobile server network graph whenever a communication link requires a relay path. We define the shortest path to be the one that contains the smallest number of mobile servers.

We first show the simulation results of three algorithms, i.e., *ANS*, *GREEDY* and *IGREEDY*, in a large system with 30 nodes and 5 mobile servers. Fig. 7 shows the detailed results of these three algorithms that illustrates the accumulated number of nodes not covered in the ith iteration. As seen from Fig. 7, the curve of the *IGREEDY* algorithm

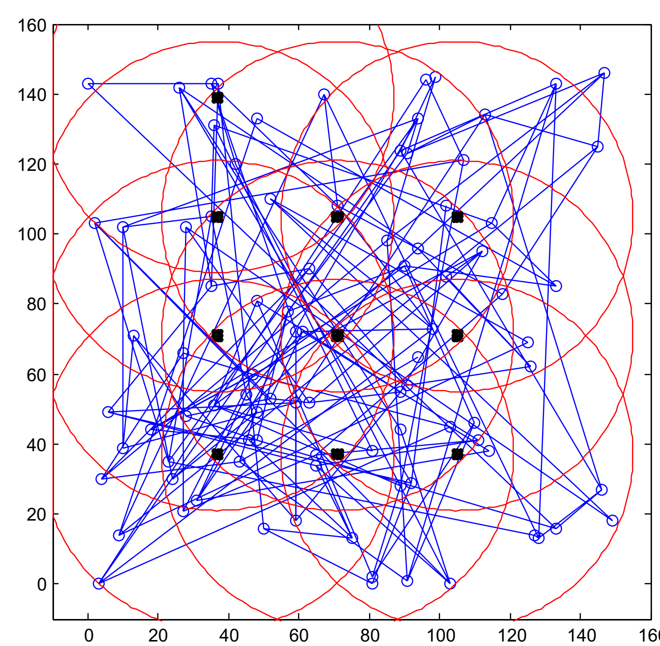


Fig. 6. A randomly generated large simulation system.

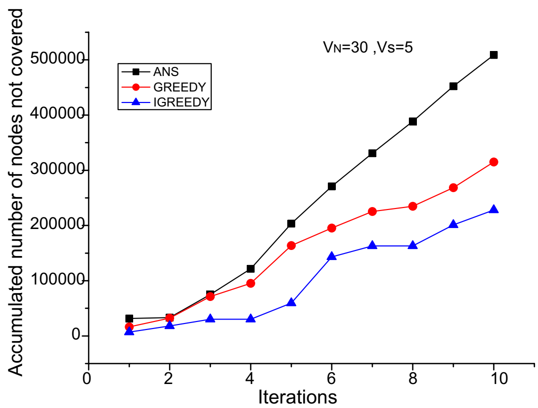


Fig. 7. The accumulated number of nodes not covered till the *i*th iteration.

to represent the number of nodes not served by any mobile server is always below the ones of the other two. In this new environment of a large system, the *IGREEDY* algorithm produced better graph partitions than *ANS* and *GREEDY* algorithms did.

Table 3 shows the accumulated number of nodes not covered in different iterations. We can conclude that the *IGREEDY* algorithm reduced up to 55.2% ($\frac{50{,}899-22{,}794}{50{,}899}$) of nodes not covered by any mobile server compared with the *ANS* algorithm and only had 2.2 nodes not covered per time slot among 30 mobile nodes. The *IGREEDY* algorithm failed to cover nodes in a similar percentage that is 11% (1.1/10) in the simulated small system and 7.3% (2.2/30) in the simulated large system.

We repeated the test of these three algorithms by increasing *cv* and *ce* values by 50% for each mobile server. Table 4 shows the new results. The increased bandwidth of mobile servers will consequently reduce the number of nodes without provided service in most cases. Although in one iteration under arbitrarily generated network graphs the new results in Table 4 might not be better than the results in Table 3, more system resources can reduce the number of nodes not covered when we conducted simulations in more iterations, which can be demonstrated as less number of nodes in Table 4 in 10 iterations than in Table 3. Fig. 8 shows the accumulated number of nodes without service in this newly tested environment with more system bandwidth capacity in detail.

Table 3
Average number of mobile nodes not covered in a large system per iteration

Algorithms	*ANS*	*GREEDY*	*IGREEDY*
1 iteration	31,731	16,548	6826
5 iterations	54,185	39,048	28,635
10 iterations	50,899	31,521	22,794

Table 4
Average number of mobile nodes not covered per iteration given more system resources

Algorithms	*ANS*	*GREEDY*	*IGREEDY*
1 iteration	51,635	34,112	11,635
5 iterations	35,548	25,036	17,202
10 iterations	28,809	21,619	13,919

Then, we show the impact of system configurations on the performance of algorithms *ANS*, *GREEDY* and *IGREEDY*. Fig. 9 shows the impact from the variance of mobile nodes. The curves display the accumulated number of nodes not covered with the number of mobile nodes increased from 30 to 100, in which we set the number of servers to be $|V_S| = 10$ in 10,000 iterations. We can see that the accumulated number of nodes not covered increases along with the increment of mobile nodes. This is because the bandwidth capacity *cv* and *ce* of each server is fixed, which leads to the decrease of average available resources for each node. Therefore, an iteration with a large number of nodes could make a few mobile nodes not able to acquire the service provided by the server. Both *GREEDY* and *IGREEDY* have a better performance than *ANS* does, which is constant with Fig. 7.

Fig. 10 shows the impact from the variance of mobile servers. The curves display the accumulated number of nodes not covered given the number of mobile servers

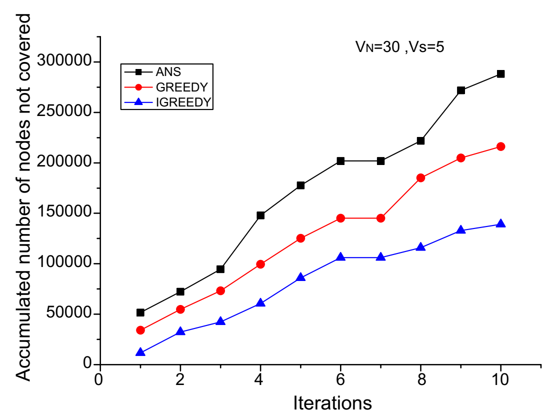


Fig. 8. The accumulated number of nodes not covered given more system resources.

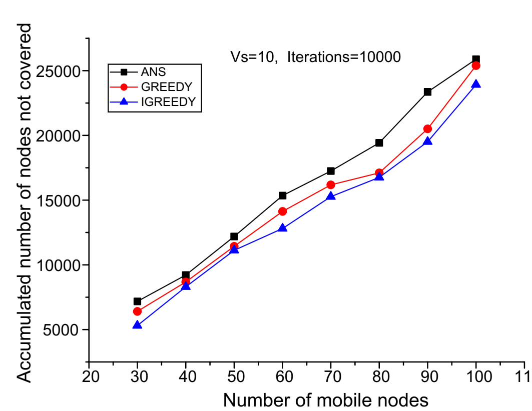


Fig. 9. The accumulated number of nodes not covered with the increment of mobile nodes.

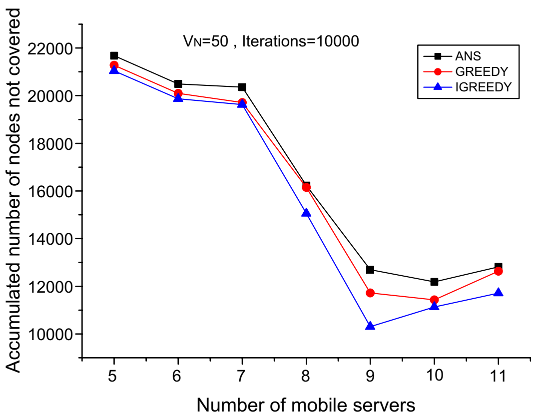


Fig. 10. The accumulated number of nodes not covered given an increased number of mobile servers.

varied from 5 to 11 in 10,000 iterations with $|V_N| = 50$. The accumulated number of nodes not covered decreases dramatically with the addition of mobile servers at the beginning, and arrives at a stable minimum when $|V_S| = 10$. More servers can provide more resources, which is helpful to cover more nodes. However, this trend becomes flat when the number of servers is great than 10, which is the result of uncovered nodes due to their locations out of the server coverage area.

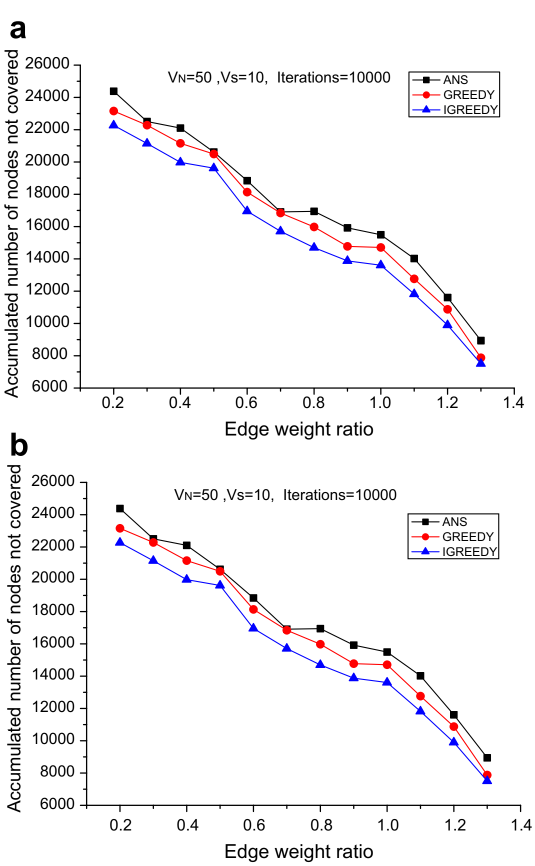


Fig. 11. The accumulated number of nodes not covered: (a) edge weight ratio α_{ce}; (b) node weight ratio α_{cv}.

Simulation results also show that the available bandwidth capacities ce, cv of mobile servers have a direct impact on partition results of *ANS*, *GREEDY* and *IGREEDY* algorithms, where *cv* stands for the available intra-server communication bandwidth while *ce* stands for the available inter-server communication bandwidth. The settings of *ce* and *cv* are denoted by α_{ce} and α_{cv}, respectively. Fig. 11 shows the accumulated number of nodes not covered when $|V_S| = 10$ and $|V_N| = 50$. Fig. 11(a) displays the simulation result when the value of α_{ce} is increasing from 0.2 to 1.3 when α_{cv} is set to be 1. Fig. 11(b) displays the impact from α_{cv} when α_{ce} is set to be 1.3. The curves of all three algorithms move downward when either α_{ce} or α_{cv} becomes larger, which complies with theoretical analysis that the more system resources available, the less number of uncovered mobile nodes the system will yield from partitions.

7. Conclusion

In this paper, we addressed the partitioning problem in a large-scale wireless communication system, which aims at providing communication services to a large number of mobile units. Effective partitioning is particularly important when resources are limited, for example when there is only a limited number of mobile servers and when the transmission range of the mobile servers is short. To accurately describe a dynamically changing wireless system, we used two graphs, a mobile node connection graph and a mobile server network graph. Central to the partitioning problem is the issues of the valid assignment of mobile nodes to mobile servers, a problem for which searching for an optimal solution is proved to be NP-hard. The assignment problem becomes more difficult when mobile nodes and servers are able to move freely and to be inserted/deleted dynamically. We proposed five algorithms *MINcut*, *MAXcv*, *ANS*, *GREEDY* and *IGREEDY*. *MINcut* and *MAXcv* exhaustively search for the best solution under given conditions while *ANS*, *GREEDY* and *IGREEDY* are heuristic algorithms. Simulation results show that *GREEDY* and *IGREEDY* algorithms can be applied efficiently to realistic large-scale wireless communication systems and can also provide a good partition.

Acknowledgment

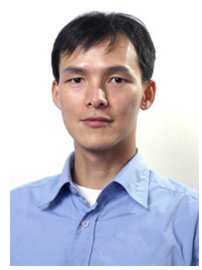
The authors thank the anonymous referees for their constructive comments. This work is supported by HK CERG B-Q827 (PolyU 5196/04E).

References

[1] M. Corson, S. Batsell, and J. Macker, Architectural considerations for mobile mesh networking, in: Proceedings of IEEE MILCOM '96, (McLean, Virginia), October 1996.

[2] V.D. Park, M.S. Corson, A highly adaptive distributed routing algorithm for mobile wireless networks, in: Proceedings of Sixteenth

Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '97), April 1997, pp. 1405–1413.
[3] W. Li, X. Chao, Modeling and performance evaluation of a cellular mobile network, IEEE/ACM Transactions on Networking 12 (2004) 131–145.
[4] N. Kruijt, D. Sparreboom, F. Schoute, R. Prasad, Location management strategies for cellular mobile networks, Journal of Electronics and Communication Engineering 10 (1998) 64–72.
[5] E. Royer, C.-K. Toh, A review of current routing protocols for ad hoc mobile wireless networks, IEEE Personal Communications 6 (1999) 46–55.
[6] Y.-C. Tseng, S.-L. Wu, W. Liao, C.-M. Chao, Location awareness in ad hoc wireless mobile networks, IEEE Computer 34 (2001) 46–52.
[7] S.R. Das, R. Castaneda, J. Yan, Simulation-based performance evaluation of routing protocols for mobile ad hoc networks, ACM Mobile Networks Applications 5 (2000) 179–189.
[8] Y.-L. Chang, C.-C. Hsu, Routing in wireless/mobile ad-hoc networks via dynamic group construction, ACM Mobile Networks and Applications 5 (2000) 27–37.
[9] H. Huang, A.W. Richa, M. Segal, Approximation algorithms for the mobile piercing set problem with applications to clustering in ad-hoc networks, ACM Mobile Networks and Applications 9 (2004) 151–161.
[10] H. Hassanein, A. Safwat, Virtual base stations for wireless mobile ad hoc communications: an infrastructure for the infrastructure-less, International Journal of Communication Systems 14 (2001) 763–782.
[11] C.R. Lin, M. Gerla, Adaptive clustering for mobile wireless networks, IEEE Journal of Selected Areas in Communications 15 (7) (1997) 1265–1275.
[12] J. Sucec, I. Marsic, Clustering Overhead for Hierarchical Routing in Mobile Ad hoc Networks, in: IEEE INFOCOM 2002, June 2002, pp. 1698–1706.
[13] C.-S. Hsu, J.-P. Sheu, Design and performance analysis of leader election and initialization protocols on ad hoc networks, Journal of Wireless Communications and Mobile Computing 3 (2003) 487–502.
[14] O. Dousse, P. Thiran, M. Hasler, Connectivity in ad-hoc and hybrid networks, in: Proceedings of IEEE INFOCOM, June 2002, pp. 1079–1088.
[15] H.-Y. Hsieh, R. Sivakumar, On using the ad-hoc network model in cellular packet data networks, in: Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing, June 2002, pp. 36–47.
[16] H.-Y. Hsieh, R. Sivakumar, Performance comparison of cellular and multi-hop wireless networks: a quantitative study, in: Proceedings of the 2001 ACM SIGMETRICS international conference on Measurement and modeling of computer systems, June 2001, pp. 113–122.
[17] A. Tanaka, K. Nakano, M. Sengoku, S. Shinoda, Analysis of communication traffic characteristics of a cellular system with ad hoc networking, Electronics and Communications in Japan (Part I: Communications) 86 (2003) 1–10.
[18] M.R. Garey, D.S. Johnson, Computers and Intractability – A Guide to the Theory of NP-Completeness, Freeman, San Francisco, CA, 1979.
[19] T. Feder, P. Hell, S. Klein, R. Motwani, Complexity of graph partition problems, in: Proceedings of the thirty-first annual ACM symposium on Theory of computing, May 1999, pp. 464–472.
[20] H. Yang, D.F. Wong, Efficient network flow based min-cut balanced partitioning, in: Proceedings of the 1994 IEEE/ACM international conference on Computer-aided design, November 1994, pp. 50–55.
[21] D.H. Lorenz, A. Orda, Optimal partition of QoS requirements on unicast paths and multicast trees, IEEE/ACM Transactions on Networking 10 (2002) 102–114.
[22] J. Sharony, An architecture for mobile radio networks with dynamically changing topology using virtual subnets, ACM Mobile Networks and Applications 1 (1996) 75–86.
[23] S.-T. Cheng, C. Chen, C. Li, C.-M. Chen, Network topology management in a mobile-switch ATM network: dynamic partition algorithms, International Journal of Network Management 12 (2002) 99–115.



Bin Xiao received the B.Sc and M.Sc degrees in Electronics Engineering from Fudan University, China. in 1997 and 2000 respectively, and Ph.D. degree from University of Texas at Dallas, USA, in 2003 from Computer Science. Now he is an Assistant Professor in the Department of Computing of Hong Kong Polytechnic University, Hong Kong. His research interests include communication and security in computer networks, peer-to-peer networks, wireless mobile ad hoc and sensor networks.

Jiannong Cao received the BSc degree in computer science from Nanjing University, Nanjing, China in 1982, and the MSc and the Ph.D degrees in computer science from Washington State University, Pullman, WA, USA, in 1986 and 1990 respectively He is currently a professor in the Department of Computing at Hong Kong Polytechnic University, Hung Hom, Hong Kong. He is also the director of the Internet and Mobile Computing Lab in the department. Before joining Hong Kong Polytechnic University, he was on the faculty of computer science at James Cook University and University of Adelaide in Australia, and City University of Hong Kong. His research interests include parallel and distributed computing, networking, mobile and wireless computing, fault tolerance, and distributed software architecture. He has published over 180 technical papers in the above areas. His recent research has focused on mobile and pervasive computing systems, developing test-bed, protocols, middleware and applications.

Zili Shao received the BE Degree in Electronic Mechanics from the University of Electronic Science and Technology of China, China, 1995. He received the MS and PhD Degrees from the Department of Computer Science at the University of Texas at Dallas, in 2003 and 2005, respectively. He has been an Assistant Professor in the Department of Computing at the Hong Kong Polytechnic University since 2005. His research interests include embedded systems, high-level synthesis, compiler optimization, hardware/software co-design and computer security.

Qingfeng Zhuge received her Ph.D. from the Department of Computer Science at the University of Texas at Dallas. She obtained her BS and MS degrees in Electronics Engineering from Fudan University, Shanghai, China. Her research interests include embedded systems, real-time systems, parallel architectures, optimization algorithms, high-level synthesis, compilers, and scheduling.

Edwin Hsing-Mean Sha received the M.A. and Ph.D. degree from the Department of Computer Science, Princeton University, Princeton, NJ, in 1991 and 1992, respectively. From August 1992 to August 2000, he was with the Department of Computer Science and Engineering at University of Notre Dame, Notre Dame, IN. He served as Associate Chairman from 1995 to 2000. Since 2000, he has been a tenured full professor in the Department of Computer Science at the University of Texas at Dallas. He has published more than 210 research papers in refereed conferences and journals. He has served as an editor for many journals, and as program committee members and Chairs in numerous international conferences. He received Oak Ridge Association Junior Faculty Enhancement Award, Teaching Award, Microsoft Trustworthy Computing Curriculum Award, and NSF CAREER Award.