WIRELESS COMMUNICATIONS AND MOBILE COMPUTING
Wirel. Commun. Mob. Comput. 2009; **9**:759–771
Published online 21 April 2008 in Wiley InterScience
(www.interscience.wiley.com). DOI: 10.1002/wcm.627

Stability-aware multi-metric clustering in mobile ad hoc networks with group mobility

Hui Cheng[1,2*,†] Jiannong Cao[1], Xingwei Wang[3], Sajal K. Das[4], Shengxiang Yang[2]

[1] *Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong*
[2] *Department of Computer Science, University of Leicester, University Road, Leicester LE1 7RH, U.K.*
[3] *College of Information Science and Engineering, Northeastern University, Shenyang 110004, China*
[4] *Department of Computer Science and Engineering, The University of Texas at Arlington, TX, U.S.A.*

Summary

Clustering can help aggregate the topology information and reduce the size of routing tables in a mobile ad hoc network (MANET). The maintenance of the cluster structure should be as stable as possible to reduce overhead and make the network topology less dynamic. Hence, stability measures the goodness of clustering. However, for a complex system like MANET, one clustering metric is far from reflecting the network dynamics. Some prior works have considered multiple metrics by combining them into one weighted sum, which suffers from intrinsic drawbacks as a scalar objective function to provide solution for multi-objective optimization. In this paper, we propose a stability-aware multi-metric clustering algorithm, which can (1) achieve stable cluster structure by exploiting group mobility and (2) optimize multiple metrics with the help of a multi-objective evolutionary algorithm (MOEA). Performance evaluation shows that our algorithm can generate a stable clustered topology and also achieve optimal solutions in small-scale networks. For large-scale networks, it outperforms the well-known weighted clustering algorithm (WCA) that uses a weighted sum of multiple metrics. Copyright © 2008 John Wiley & Sons, Ltd.

KEY WORDS: mobile ad hoc networks; clustering; stability; multi-objective optimization

1. Introduction

A mobile ad hoc network (MANET) is a self-organizing wireless local area network without infrastructure and central administration. It has the advantages of low cost, plug-and-play convenience, and flexibility. Just like the Internet, the flat network infrastructure of MANETs encounters the scalability problem when the network size increases. Scalability is more challenging in MANETs due to node mobility. Therefore, efficient network management is extremely important. Analogous to the IP subnet concept, a MANET can also be organized into a hierarchical architecture by dividing nodes into clusters. Each cluster maintains and aggregates the information of the nodes within it. Each cluster can thus be seen as a logical node at the cluster level. The network layer only needs to maintain and manage the information of these logical nodes. Clearly, the control overhead will be reduced with the aid of clustering.

*Correspondence to: Hui Cheng, Department of Computer Science, University of Leicester, University Road, Leicester LE1 7RH, UK.
†E-mail: hc118@mcs.le.ac.uk

A clustering algorithm [1] is to find a feasible interconnected set of clusters covering the entire set of nodes in MANET. At any instant, one mobile node can only belong to one cluster. A cluster may have a clusterhead or not. Since the recruiting of clusterheads brings the advantage of easy management, most of the prior research work is on clustering with clusterhead. In this paper, our algorithm also generates the clusters with clusterheads assigned.

Despite the fact that node mobility is an intrinsic characteristic in MANETs, the cluster structure should be maintained as stable as possible. Otherwise, frequent cluster change or re-clustering adversely affects the performance of radio resource allocation and scheduling protocols. By stability, we mean that the cluster structure remains unchanged for a given reasonable time period. Clearly, stability is an important requirement on a clustering algorithm. To maintain stable cluster structure, node mobility and group mobility [2] must be investigated. Group mobility has emerged from applications where a team of mobile users stay closely and move together. Mobile nodes are organized into groups to coordinate their movement. Examples include military and disaster recovery operations, vehicular communications, etc. Although existing work [3,4] addressed the relative mobility, yet the effect of group mobility on clustering has not been studied.

Furthermore, clustering must be associated with one or more metrics such as node ID, node degree, and energy (battery power), which are defined based on the application requirements. Early work in the literature has focused on single-metric clustering. For example, in the highest degree heuristic [5], the node with the maximum number of neighbors (highest degree) is chosen as the clusterhead. But for a complex system like MANET, single metric is far from reflecting the whole network dynamics. Clustering algorithms optimizing only one metric commonly lose generality and have low performance in terms of other metrics.

Multi-metric clustering aims to create a cluster structure that optimizes several metrics simultaneously. Some existing work [6–8] considered multi-metric clustering, but adopted the traditional method of linear combination (weighted sum) of multiple metrics. It is known that a single scalar objective function on ad hoc basis not only makes the solution highly sensitive to the chosen weight vector but also requires the user to have some knowledge about the priority or influence of a particular objective parameter over another [9]. For multi-metric clustering, the same problem occurs because different metrics evaluate different capabilities of mobile nodes. Moreover, the evaluation criterion is different for different metrics. Hence, it is difficult to determine the weighing factors for the metrics in the linear combination formula. If an algorithm uses the weighted sum as a single metric, in our opinion, it is a single-metric clustering approach since it results in only one final solution. This solution cannot always optimize all the metrics simultaneously.

In this paper, we propose a stability-aware multi-metric clustering algorithm for MANETs with group mobility. The motivation comes from the property of group mobility: *the distances between two neighboring nodes in the same group exhibit the relative stability*. To exploit this property, we define the concept of relatively stable neighbors, and based on it construct a relatively stable network topology. Then we run the multi-metric clustering procedure on the relatively stable topology to achieve stable clusters. Hence, the proposed clustering algorithm considers both stability and multi-metric optimization.

We define three clustering metrics as optimization objectives: total node degree differences, total power consumption, and minimum remaining battery lifetime. They respectively represent three important requirements for clustering: load balance, energy efficiency, and maximum lifetime. Our algorithm adopts a promising multi-objective evolutionary algorithm (MOEA), called Strength Pareto Evolutionary Algorithm 2 (SPEA2), that provides Pareto-optimal solutions with elaborate problem-specific design and modification [10]. We conduct simulations to evaluate the performance in terms of stability and multi-metric optimization. The results show that our proposed algorithm can generate stable cluster structures and high-quality clusterhead sets regarding all the clustering metrics.

Recently, MOEAs have been extensively used in research on networking, for example, mobile multicast [9], RSVP performance evaluation [11], and so on. To our best knowledge, the proposed clustering algorithm is the first to optimize multiple metrics based on MOEA. It can produce a set of good solutions instead of a single solution to meet the requirements of multi-metric clustering.

The paper is organized as follows. Section 2 provides an overview of clustering in MANETs and discusses related work. Section 3 discusses the stability issue and defines three node metrics for clustering. Section 4 introduces the multi-objective evolutionary optimization technique. The design and implementation of the proposed algorithm is described in

DOI: 10.1002/wcm

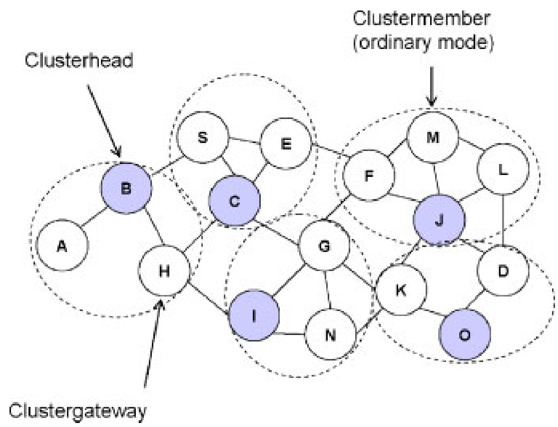


Fig. 1. Example of clustering in MANET.

Section 5. Performance evaluations are conducted in Section 6. Finally, a summary of results is made in Section 7.

2. Related Work

A typical cluster structure in a MANET is shown in Figure 1. Within one cluster, mobile nodes may play different roles, such as clusterhead, clustergateway, or clustermember. A clusterhead normally serves as a local coordinator for its cluster, performing intra-cluster transmission control, data forwarding, and so on. A clustergateway is a non-clusterhead node with inter-cluster links, so it can access neighboring clusters and forward data between clusters. A clustermember is an ordinary node, which is a non-clusterhead node without any inter-cluster links.

2.1. Clusterhead Selection

The primary step in clustering is the selection of clusterheads. The clusterhead can be the leader node, for example, the node with the maximum power. The selection is based on different criterion derived from specific communication requirements. For one-hop clustering, the cluster structure is determined once the clusterheads are determined. In the following, we formalize the clusterhead selection problem.

A MANET is represented as an undirected graph $G=(V, E)$, where V represents the set of mobile nodes and E represents the set of links between nodes. E always changes with the creation and deletion of links. Let $N(v)$ be the neighborhood of node v, defined as

$$N(v) = \bigcup_{v' \in V,\, v' \neq v} \{v' | \mathrm{dist}(v, v') < r\} \qquad (1)$$

where r is the transmission range of node v.

The generalized procedure for selecting the clusterhead is as follows:

Step 1. From G, select one mobile node v as a clusterhead according to a certain rule.
Step 2. Delete node v and all its neighbors (i.e., all nodes in $N(v)$) from G.
Step 3. Repeat Steps 1–2 for the remaining nodes in G until G is empty.

The above three steps generate a set of clusterheads. In Step 1, the rule determines which node is selected as the clusterhead. Different clustering algorithm defines different rules, such as the lowest node-ID, the highest node-degree, the least node-weight, etc.

2.2. Multi-Metric Clustering

A well-known weighted clustering algorithm (WCA), which optimizes a linearly combined weight consisting of four metrics, was presented in Reference [6]. It takes nodes with less mobility as a better choice for clusterheads. But this may not always be useful. Consider the case that all the nodes are moving rapidly except one slow-speed node, which lags behind. How can it play the role of the clusterhead? So instead of absolute mobility, relative mobility is a more reasonable metric. In addition, WCA specifies the values of weighing factors rather arbitrarily since it is hard to determine them precisely.

In References [7,8], two intelligent optimization techniques, genetic algorithm (GA) and simulated annealing (SA), are used to optimize WCA such that the number of clusterheads is minimized while load in the network is as evenly balanced as possible among all the clusters. Both of these approaches optimize the WCA further, but they still use a weighted linear combination of the associated metrics. In other words, they still address the multi-metric clustering by single-objective optimization.

3. Preliminaries

3.1. Stability-Aware Clustering

A MANET can be dynamically organized into clusters to maintain a relatively stable and effective topology. If clusters exist, the distances between the clusterhead

DOI: 10.1002/wcm

and cluster members should stabilize over a certain period of time. As shown in Reference [2], two neighboring mobile nodes in the same mobility group show relative stability of distances. Assume that all nodes have identical and fixed transmission range r. If the distance between two mobile nodes is within r, they can communicate with each other directly. However, this does not necessarily mean that they belong to the same group. Imagine that two mobile nodes briefly fall in the transmission range geographically and separate again, due to different moving directions. Based on these observations, the term adjacently grouped pair (AGP) of nodes is defined as follows.

Definition 1 ([2]). *Nodes A and B form an AGP, denoted by* $A \overset{0}{\sim} B$, *if their distance* $\|AB\|$ *obeys normal distribution with a mean* $\mu < r$, *and a standard deviation* $\sigma < \sigma_{max}$.

The definition shows that if two adjacent nodes are in the same group over a period of time, the distance between them stabilizes around a mean value μ with small variations, where $\mu < r$. Based on the AGP definition, we propose the concept of relatively stable neighbors.

Definition 2. *Node A and B are relatively stable neighbors if they form an AGP.*

The relatively stable neighbors of a node can be determined by measuring the distances between it and its neighbors for a fixed number of rounds l, where l is a pre-determined size of the sampling buffer. The relatively stable topology is constructed for clustering over relatively stable neighbors. The construction method is described in Subsection 5.3.

3.2. Node Metrics

In the proposed clustering algorithm, we only consider relatively stable neighbors. If one node is selected as the clusterhead, only its relatively stable neighbors can join this cluster. If node i is node j's relatively stable neighbor but has already joined another cluster, node i cannot join the cluster served by node j again. Hence, when calculating the clustering metrics for node j, node i should be excluded from node j's available relatively stable neighbors. Each node can decide how well suited it is for being a clusterhead by the following three metrics.

3.2.1. Degree difference

In our algorithm, the degree of a node is only the number of its relatively stable neighbors. Suppose D_v is the number of relatively stable neighbors of node v, and δ is the number of neighbors that a clusterhead can ideally handle. Then the degree-difference Δ_v is used as one metric to evaluate the load of node v.

$$\Delta_v = |D_v - \delta| \tag{2}$$

The less Δ_v, the more suitable for node v to be a clusterhead.

3.2.2. Power consumption

It is known that the power required for supporting a link is inversely proportional to some exponent power of the distance in wireless communications. But because the distance between two neighboring nodes in a MANET is usually rather small (approximately hundreds of meters) as compared to the distance between mobile devices and base stations (the order of 2–3 miles), the power for supporting a wireless link can be regarded as being proportional to the distance in MANET [6]. So we use Dist_v, the sum of the distances between node v and its each available relatively stable neighbor, to evaluate the power consumed for communication between cluster members and node v. Thus,

$$\mathrm{Dist}_v = \sum_{v' \in N(v)} \{\mathrm{dist}(v, v')\} \tag{3}$$

where $N(v)$ is the set of available relatively stable neighbors of node v, and $\mathrm{dist}(v, v')$ is the measured average distance between node v and v'.

3.2.3. Remaining battery lifetime

Each mobile node v can easily estimate its remaining battery energy E_v. Since the power consumed by node v to communicate with its relatively stable neighbors is Dist_v, its remaining battery lifetime, Rbl_v, can be represented as

$$\mathrm{Rbl}_v = \frac{E_v}{\mathrm{Dist}_v} \tag{4}$$

It is expected that the nodes with longer remaining battery lifetime are selected as clusterheads.

DOI: 10.1002/wcm

4. Multi-Objective Evolutionary Optimization

Conventional search techniques, such as hill climbing [12], are often incapable of optimizing non-linear multimodal functions. In such cases, a random search method might be required. Evolutionary algorithm (EA, also called GA) is a well-known guided random search and optimization technique. It is based on the basic principles of evolution: survival of the fittest and inheritance. Generally, EA is applied to find an approximate optimal solution with respect to a fitness function for NP-hard problems.

Many real-life optimization problems have multiple objectives. In such optimization problems, the objectives often conflict across a high-dimensional problem space. Solving these problems is generally very difficult and may require extensive computational resources. The presence of multiple objectives in a problem, in principle, gives rise to a set of compromised solutions (largely known as Pareto-optimal solutions), instead of a single optimal solution. The definition of Pareto-optimal is as follows [13].

Definition 3. *A point x^* is Pareto-optimal if for every x either $\cap_i(f_i(x) = f_i(x^*))$ or there is at least one i such that $f_i(x) > f_i(x^*)\ \forall i \in I$ (set of integers), where $f_i(x)$ is the fitness function. In other words, x^* is Pareto-optimal if there exists no feasible vector x which would decrease some criterion without causing a simultaneous increase in at least one other criterion.*

Solution A is said to dominate solution B if A is better than B in at least one objective value and is no worse in all other objective values. A Pareto-optimal solution is called a non-dominated solution. Table I gives a simple example to explain it. There are three solutions A, B, C. Each solution has three objective values. Suppose that the less the objective value, the better it is. In our example, A dominates B. For both A and C, since no other solutions dominate them, they are non-dominated solutions, that is, Pareto-optimal solutions. The goal of multi-objective optimization is to find as many Pareto-optimal solutions as possible.

The particular MOEA used in this work is SPEA2. As shown in Reference [10], SPEA2 provides good performance in terms of convergence and diversity, and compares well to other representative MOEAs on various well-known test problems.

Table I. A simple example for Pareto-optimal solution.

Solution	Object value 1	Object value 2	Object value 3
A	1.5	3	2
B	1.6	4	3
C	0.5	4	4

5. Algorithm Details

5.1. Problem Encoding

Chromosome is the basic element in an EA. A certain number of chromosomes form a population. The encoding of a chromosome is important. First, each chromosome should represent a feasible solution, which is randomly distributed in the solution space. Second, a good encoding method benefits the realization of genetic operations.

Each solution produced by our algorithm stands for a set of clusterheads, which are selected from all the nodes in the network. Hence, a random permutation of node IDs will result in a random set of clusterheads. In this algorithm, we use random permutation of node IDs to represent a chromosome. It is important to guarantee that there is no duplicate node ID in each chromosome. Each node ID in the chromosome is called a gene. For example, in a MANET consisting of eight nodes with IDs ranging from 1 to 8, a random permutation (4 3 8 7 1 6 2 5) represents a chromosome.

We need to derive a set of clusterheads from each chromosome. Let us explain this method with an example. Assume the chromosome is (4 3 8 7 1 6 2 5). Figure 2 shows the relatively stable topology

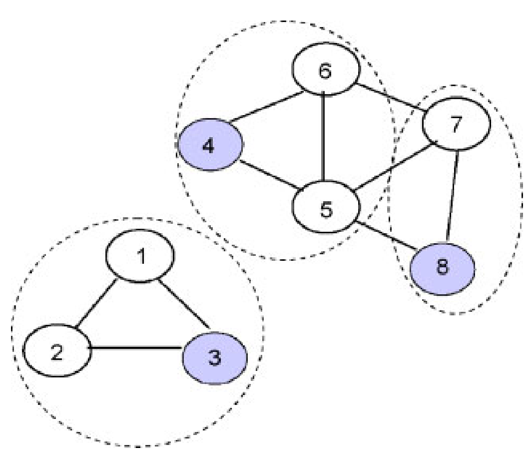


Fig. 2. A relatively stable topology.

DOI: 10.1002/wcm

Table II. Procedure for deriving a set of clusterheads from a chromosome.

Step	Candidate genes for clusterheads	Set of clusterheads
1	(4 3 8 7 1 6 2 5)	{ }
2	(- 3 8 7 1 - 2 -)	{4}
3	(- - 8 7 - - - -)	{4,3}
4	(- - - - - - - -)	{4,3,8}

constructed from the network. First, we add the first gene 4 into the clusterhead set. Then all the relatively stable neighbors of node 4 are no longer allowed to be clusterheads. From Figure 2, we know the relatively stable neighbors of node 4 are nodes 5 and 6. We continue to check the next gene and add node 3 into the clusterhead set. The relatively stable neighbors of node 3 are nodes 1 and 2. So nodes 1, 2, 5, and 6 are not considered as clusterheads any more. Then we add node 8 into the clusterhead set. The available relatively stable neighbors of node 8 are node 7. Hence, node 7 is also forbidden to be the clusterhead. Until now, all the nodes have been checked and a clusterhead set $\{4, 3, 8\}$ is generated. Table II illustrates the procedure of clusterhead selection and Figure 2 illustrates the clustering results.

Since we consider a MANET with group mobility, without loss of generality, we assume all the nodes form groups and the average group size is $\bar{g}$. Hence, the number of groups is $n/\bar{g}$. Due to the fact that a clusterhead and all its relatively stable neighbors belong to the same group, there are no inter-group clusters. Within a group, each group member needs to send the message about its relatively stable neighbors information to the group leader. Normally, the group leader stays at the group center. We assume the group diameter is d. Thus, the average number of hops that a message travels is $(1 + d/2)/2$. Then the total number of clustering messages in a group is given by $(\bar{g} - 1) \times [(1 + d/2)/2]$. Since there is no need to send inter-group clustering messages, the total number of clustering messages in the network is $(n/\bar{g}) \times (\bar{g} - 1) \times [(1 + d/2)/2]$. As a result, the time complexity of the technique used to derive a clusterhead set is $O(n \times d)$.

5.2. Optimization Objectives

In Subsection 3.2, we define three metrics to evaluate the suitability of a node as the clusterhead. Since each node can calculate these metrics based on its local information, we assume that every node is aware of the current values of its metrics. In our problem, we should evaluate each clusterhead set instead of each single clusterhead. Hence, we need to give an overall evaluation on the clusterhead set in terms of each metric. Since both degree difference and power consumption are additive metrics, it is natural to use the sum of the metric value of each clusterhead as the overall optimization objective (i.e., clustering metric). The sum of degree difference of each clusterhead reflects the overall deviation of the node degrees from the ideal case. The sum of power consumed by each clusterhead reflects the total power consumed by all the clusterheads. However, the metric for the remaining battery lifetime is a concave function. Hence, the reasonable evaluation object is the minimum remaining battery lifetime among all the clusterheads because it determines the maximum lifetime of the whole clusterhead set.

Assume $s\_CH = \{c_1, c_2, \ldots, c_m\}$ is a set of clusterheads. We define the following three optimization objectives for s_CH:

(1) The total degree differences of all the clusterheads is given as:

$$\Delta_{s\_CH} = \sum_{c_i \in s\_CH} \Delta_{c_i} \tag{5}$$

(2) The power that all the clusterheads consume is given as

$$D_{s\_CH} = \sum_{c_i \in s\_CH} D_{c_i} \tag{6}$$

(3) The minimum remaining battery lifetime is given by

$$\mathrm{Rbl}_{s\_CH} = \mathrm{Min}\{\mathrm{Rbl}_{c_i} | c_i \in s\_CH\} \tag{7}$$

For both Δ_{s_CH} and D_{s_CH}, the less the value, the better the clusterhead set. This is due to the fact that we expect each clusterhead to serve just δ cluster members and consume as little power as possible for intra-cluster communication. However, for Rbl_{s_CH}, we expect its value as large as possible. So our objective is to minimize both Δ_{s_CH} and D_{s_CH}, and maximize Rbl_{s_CH}.

5.3. Stability-Aware Multi-Objective Clustering Algorithm

We first construct a relatively stable network topology for a MANET by the following method:

DOI: 10.1002/wcm

Step 1. For each node v, find out all its relatively stable neighbors $N(v)$.
Step 2. For each node $w \in N(v)$, if there is no link between v and w, add a bidirectional link to connect them.
Step 3. Repeat Steps 1–2 until all the mobile nodes have been processed.

The relatively stable topology can be regarded as a 'quasi-static' network topology over a certain period of time. The following multi-metric clustering procedure just runs on this relatively stable topology. Just like the distributed clustering algorithm [14], we assume that the network topology does not change during the execution of the clustering algorithm.

In the following, we present the formal description of the proposed clustering algorithm as shown in Figure 3. In the beginning, the algorithm constructs the relatively stable topology, which is used to discover the relatively stable neighbors by each node. Line 3 creates the initial population P_0 and the empty Pareto set Q_0. Thus, P_0 consists of a certain number of chromosomes, which are represented by random permutation of node IDs, whereas Q_0 is the final output of this algorithm and initialized to be empty. Both P and Q have constant size in the algorithm. In Line 4, T denotes the maximum number of evolutionary generations and t denotes the current generation number that the population has evolved to. The algorithm stops when T is reached.

1. **begin**
2. construct relatively stable topology
3. create the initial population P_0 and the empty Pareto set Q_0
4. let T be the maximum generation number and set counter $t = 0$
   while $t < T$ do
5. **for** each chromosome $i \in P_t \cup Q_t$ do
6. derive the clusterhead set s_CH from i
7. **for** each node $v \in s\_CH$
8. node v calculates its own metric values: Δ_v, $Dist_v$, and Rbl_v
9. **end** of for each node $v \in s\_CH$ loop
10. calculate the three optimization objectives for s_CH: Δ_{s_CH}, D_{s_CH}, and Rbl_{s_CH}
11. **end** of for each chromosome $i \in P_t \cup Q_t$ loop
12. copy all non-dominated chromosomes in both P_t and Q_t to Q_{t+1}
13. perform SPEA2 environmental selection on Q_{t+1}
14. **if** ($t >= T$) then break
15. perform SPEA2 mating selection on Q_{t+1} to generate the mating pool
16. apply crossover and mutation operators to the mating pool and then set the resulting population to be P_{t+1}
17. $t = t + 1$
18. **end** of while loop
19. return Q_t
20. **end** of the algorithm

Fig. 3. Formal description of the stability-aware multi-metric clustering algorithm.

From each chromosome $i \in P_t \cup Q_t$, we first derive the corresponding clusterhead set, s_CH. Then each clusterhead in this set calculates its three node metrics Δ_v, $Dist_v$, and Rbl_v according to Equations (2)–(4), respectively. After all these values are obtained, the algorithm calculates the three optimization objectives (i.e., clustering metrics) Δ_{s_CH}, D_{s_CH}, and Rbl_{s_CH} following Equations (5)–(7). Thus, for each chromosome $i \in P_t \cup Q_t$, the values of its three optimization objectives are determined. Based on these values, all the non-dominated (i.e., Pareto-optimal) chromosomes in $P_t \cup Q_t$ are determined.

In Line 12, all the non-dominated chromosomes in $P_t \cup Q_t$ are copied to Q_{t+1}, the Pareto set at the $(t+1)$th evolutionary generation. It is possible that the number of non-dominated chromosomes in $P_t \cup Q_t$ is not equal to the specified size of Q_{t+1}. To solve this problem, the SPEA2 algorithm adopts the so-called environmental selection method. If the number of non-dominated chromosomes exceeds the size of the Pareto set, an archive truncation procedure is invoked, which iteratively removes chromosomes from Q_{t+1} until its size satisfies the requirement. The chromosome, which has the minimum distance to another chromosome, is removed at each iteration. If the non-dominated chromosomes cannot fulfil Q_{t+1}, the best dominated individuals in $P_t \cup Q_t$ will be added into Q_{t+1}. The algorithm then checks if the maximum generation number is reached. If so, it stops. Otherwise, the algorithm enters the SPEA2 mating selection phase, where chromosomes from Q_{t+1} are selected by means of binary tournaments to generate the mating pool.

Once the mating pool is formed, the algorithm applies crossover and mutation operators to the chromosomes in it. Crossover and mutation are two important genetic operators. Crossover helps generate two offspring chromosomes from two parent chromosomes. All the genes in each offspring chromosome are inherited from different parts of the two parent chromosomes. In this algorithm, we employ the well-known X-Order1 method. Mutation generates an offspring chromosome from only one parent chromosome by changing some genes' values. We employ the simple and effective gene swapping method for

DOI: 10.1002/wcm

mutation. Finally, the Pareto set Q_t is output as the set of solutions, each of which corresponds to a clusterhead set.

Since the initial population consists of chromosomes, which are randomly generated, there may be some duplicate chromosomes in it. In addition, the crossover and mutation operators applied to the mating pool may also produce some duplicate chromosomes in the resulting population. Therefore, there may have some duplicate chromosomes in the final Pareto set.

5.4. Cluster Reconfiguration

In this paper, we consider a MANET with group mobility, in which most of the nodes form groups. The relatively stable topology is discovered based on the property that we exploit from group mobility, that is, the relative stability of distances between two neighboring nodes belonging to the same group. Since a MANET is a dynamic system, we assume dynamic group membership. Hence, it is allowed that a new node joins a group or a group node leaves its group.

The dynamic group membership leads to the cluster reorganization. When a new node v joins a group, it first finds out all its relatively stable neighbors, $N(v)$, by the method mentioned in Subsection 3.1. A clusterhead periodically broadcasts a HELLO packet indicating its role and its cluster size. If there is no existing clusterhead in $N(v)$, node v claims itself as a clusterhead. Otherwise, among all the clusterheads in $N(v)$, node v joins the one with the smallest cluster size.

A group node intending to leave its group may be an ordinary group member or a clusterhead. When an ordinary group member leaves, its clusterhead detects its departure and deletes it from the list of relatively stable neighbors. The clusterhead then reduces its cluster size by one. When a group leader leaves, each of its clustermembers searches its own list of relatively stable neighbors. Similar to a new node, each node v claims itself to be a clusterhead or joins the clusterhead with the smallest cluster size in $N(v)$.

However, since mobile nodes form groups purposely, the group membership change rarely occurs. Hence, the effect of dynamic group membership to cluster stability is trivial.

6. Performance Evaluation

We conduct simulation study to demonstrate the effectiveness of the proposed stability-aware multimetric clustering algorithm. The standard SPEA2 source codes [15] are adopted for function modules of environmental selection and mating selection; thus the correctness of multi-objective evolutionary process is guaranteed.

Table III. The parameter values.

Population size	10
Pareto set size	20
Crossover rate	0.8
Mutation rate	0.1
Maximum evolutionary generation number	50

The main parameters used in the algorithm are population size, Pareto set size, crossover rate, mutation rate, and maximum evolutionary generation number. In our simulation experiments, the parameters are set as shown in Table III. Both the initial population size and the Pareto set size are set to be two times the normal population size. We run extensive simulations by adjusting different combinations of parameter values to achieve the best one. The simulation area is a square of $1 \times 1\,\text{km}^2$. The network size varies from 20 to 200 in different simulation scenarios. Each mobile node has the same radio transmission range of radius $r = 100\,\text{m}$.

6.1. Stability Evaluation

Cluster stability is an important performance metric in our algorithm. We evaluate it in a MANET of 200 nodes. Most of the mobile nodes form groups with various sizes. We allow 1 per cent of the nodes to be single nodes and move freely. The group nodes follow the Reference Velocity Group Mobility Model [16], in which the nodes in the same group share the common group velocity. We assume that the group velocity conforms to the random waypoint model [17] and the maximum speed varies from 5 to 25 m/s in different simulation scenarios. The single nodes also move with a random speed, which is less than the maximum speed in each simulation.

To evaluate the stability performance of the clustering algorithm, we define a new metric—the number of remaining stable clusters. It counts the number of clusters whose structures have not been changed after a time period T since the clustered topology is created. T is determined as follows:

$$T = k \times \frac{r}{0.5 \times \text{MaxSpeed}} \tag{8}$$

DOI: 10.1002/wcm

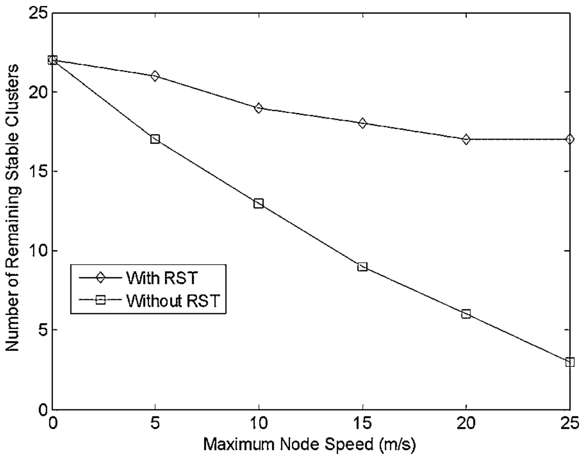


Fig. 4. Comparison of the number of remaining stable clusters for clustering with and without considering the relatively stable topology (RST).

where r is the radio transmission range, MaxSpeed is the specified maximum node speed during each simulation, and k is a constant.

In each simulation scenario with different MaxSpeed, the network is clustered using the proposed algorithm with and without considering relatively stable topology. After time T, we count the number of remaining stable clusters in these two cases. Each simulation is run three times with the averages recorded.

The comparison results are shown in Figure 4. When the network is a static ad hoc network, that is, the MaxSpeed is 0, the stability performance is the same for both cases due to no mobility. However, when the nodes move, with the relatively stable topology constructed, the stability performance achieved by the clustering algorithm is better than the case without the construction of relatively stable topology. When the network becomes more and more dynamic, that is, the node speed increases, the advantage of exploiting group mobility is more distinguished.

6.2. Multi-Metric Optimization Evaluation

It has been proved that finding an optimal set of clusterheads with one or more clustering metrics is NP-hard [6]. For small-scale network topology, since the optimal solution can be found by exhaustive search, the solutions achieved by our algorithm are compared with the optimal solutions. However, for large-scale network topology, the exhaustive search for the optimal solution becomes infeasible due to exponential time complexity. Hence, we compare the proposed algorithm with WCA [6] and its two improvements by GA [7] and SA [8].

6.2.1. Evaluation on small-scale network topology

A MANET consisting of 20 nodes is used as the small-scale network in our experimental study. We run the proposed multi-metric clustering algorithm on its relatively stable topology. Then count the number of Pareto-optimal solutions regarding each clustering metric obtained at various evolutionary generation numbers. Figure 5 shows the results.

Since the Pareto set size, that is, the number of solutions, is 20, we finally get 20 clusterhead sets, some of which may be duplicate. Figure 5 shows that after only two generations of evolution, the algorithm can achieve the optimal clusterhead sets regarding each clustering metric on the small-scale network. For the total degree differences, the ratio of the number of optimal solutions to the Pareto set size is above 50 per cent when the number of evolutionary generations exceeds 5. For power consumption, the number of optimal solutions fluctuates when the generation number is less than 8. But when it exceeds 8, the ratio of the number of optimal solutions to the Pareto set size stabilizes around 35 per cent. For the remaining battery lifetime, the ratio also stabilizes around 35 per cent when the number of generations exceeds 4.

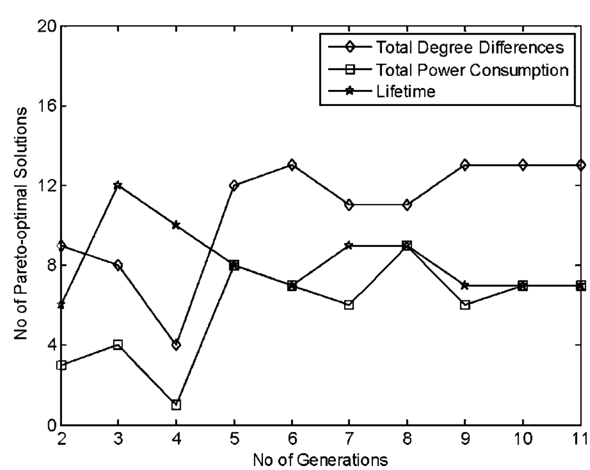


Fig. 5. The number of Pareto-optimal clusterhead sets regarding different clustering metrics.

DOI: 10.1002/wcm

6.2.2 Comparison with WCA and its two improvements

WCA uses the combined weight of four clustering metrics as the single optimization objective. The WCA algorithm has been further optimized by GA [7] and SA [8]. Here, these two improvement algorithms are named as WCA_GA and WCA_SA, respectively. We implement WCA, WCA_GA, and WCA_SA, and compare our algorithm with them to evaluate its performance in terms of the solution quality in a MANET of 200 nodes.

In the following, we simply describe the basic ideas of WCA, WCA_GA, and WCA_SA. First, WCA marks the node with the best weight as a clusterhead and all its neighbors as the cluster members. Then WCA deletes both the clusterhead and all its neighbors from the network topology. The weights of the remaining nodes are recalculated based on the remaining network topology. The above process is repeated until no node is left in the network topology.

In WCA_GA, there are also a population of chromosomes, each one of which represents a random clusterhead set. After evolving a certain number of generations, the algorithm stops since it meets one of the following termination requirements: the maximum generation number is reached or the population converges. The Elitist model is employed in WCA_GA to record the current best solution among the population. In WCA_SA, there is only one initial solution instead of a population. At each iteration, the algorithm randomly searches a solution neighboring to the present one in the solution space. If the neighbor solution is better than the current one, WCA_SA will replace the present solution by the neighbor solution. Otherwise, the algorithm will accept the neighbor solution with a probability.

Since WCA, WCA_GA, and WCA_SA are actually heuristic clustering algorithms considering single metric, we use them to find a clusterhead set for each optimization objective. Then regarding the same clustering metric, the best solution in the final Pareto set of our algorithm is compared with the clusterhead sets obtained by WCA, WCA_GA, and WCA_SA, respectively. Figures 6–8 shows the comparison results for various evolutionary generation numbers. In both WCA and WCA_SA, there is no concept of evolution. Hence, the performance of these two algorithms is not related to the number of evolutionary generations. For comparison purpose, we still show the same value of one metric with respect to different generation number.

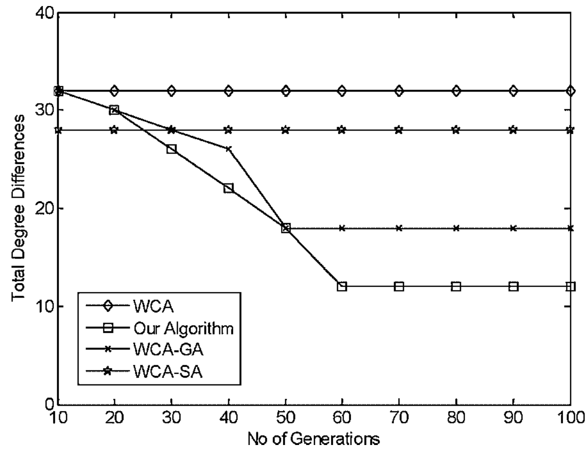


Fig. 6. Comparison of the total degree differences.

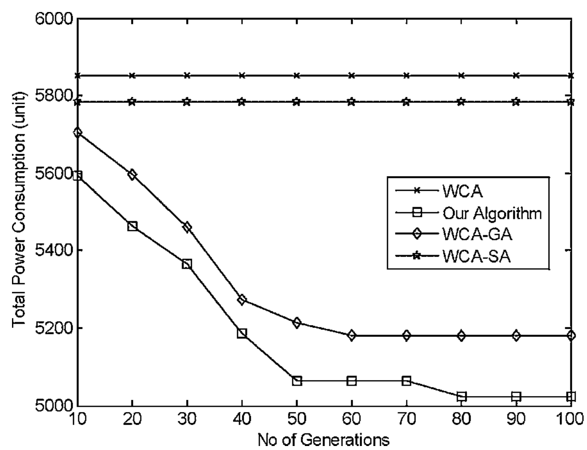


Fig. 7. Comparison of the total power consumption.

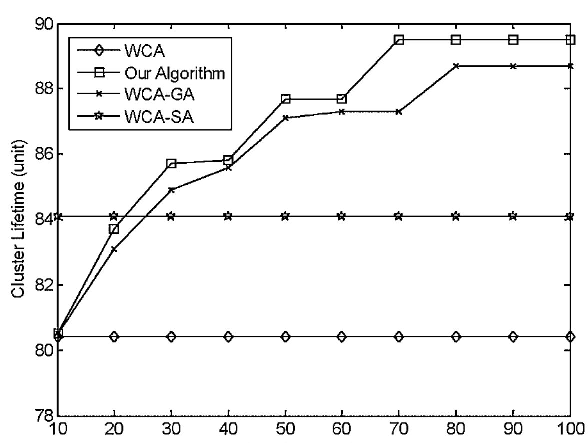


Fig. 8. Comparison of the cluster lifetime.

DOI: 10.1002/wcm

From Figures 6–8, we observe that the proposed clustering algorithm outperforms WCA, WCA_GA, and WCA_SA in terms of all the clustering metrics. When the evolutionary generation number exceeds 50, our algorithm stabilizes around some Pareto-optimal solutions. More importantly, our algorithm finds these good solutions in one Pareto set. However, WCA, WCA_GA, and WCA_SA cannot.

7. Conclusions

The selection of the optimal clusterhead set is proved to be a NP-hard problem. Hence, even for single clustering metric, we cannot find the best solution using an algorithm with polynomial time complexity. For multi-metric clustering, the problem becomes more difficult to solve. Only heuristics can be developed for multi-metric clustering in MANETs.

In this paper, we first exploit the relatively stable topology resulted by group mobility to improve the stability of the cluster structure. We then define three clustering metrics. Based on the relatively stable topology and the three clustering metrics, a stability-aware multi-metric clustering algorithm for MANETs is proposed. The algorithm can achieve a population of solutions, which are the Pareto-optimal clusterhead sets with respect to the three clustering metrics. Moreover, it can generate the Pareto-optimal solution that does not provide best possible value for any individual metric, yet it offers Pareto-optimal solution when considering all three metrics together. Such a solution is often useful for applications with a fair compromise between multiple optimization objectives.

Performance evaluations are conducted on both stability and multi-metric optimization. Simulation results show that our algorithm has good stability performance and achieves better clusterhead sets than a well-known clustering algorithm WCA and its two improvements WCA_GA and WCA_SA.

Acknowledgments

The authors thank the anonymous reviewers for their valuable comments that helped us improve the quality of the paper. This work was supported in part by the UGC of Hong Kong under the CERG grant PolyU 5170/03E, by the Engineering and Physical Sciences Research Council (EPSRC) of U.K. under Grant EP/E060722/1, by the National High-Tech Research and Development Plan of China under Grant No. 2006AA01Z214; the National Natural Science Foundation of China under Grant No. 60673159 and 70671020.

References

1. Yu Y, Chong HJ. A survey of clustering schemes for mobile ad hoc networks. *IEEE Communications Surveys and Tutorials* 2005; **7**(1): 32–48.
2. Li BC. On increasing service accessibility and efficiency in wireless ad-hoc networks with group mobility. *Wireless Personal Communications* 2002; **21**(1): 105–123.
3. An B, Papavassiliou S. A mobility-based clustering approach to support mobility management and multicast routing in mobile ad-hoc wireless networks. *International Journal of Network Management* 2001; **11**(6): 387–395.
4. Er II, Seah WKG. Mobility-based ad-hop clustering algorithm for mobile ad hoc networks. In *Proceedings of Wireless Communications and Networking Conference*, 2004.
5. Gerla M, Tsai JTC. Multicluster, mobile, multimedia radio network. *Wireless Networks* 1995; **1**(3): 255–265.
6. Chatterjee M, Das SK, Turgut D. WCA: a weighted clustering algorithm for mobile ad hoc networks. *Cluster Computing* 2002; **5**(2): 193–204.
7. Turgut D, Das SK, Elmasri R, Turgut B. Optimizing clustering algorithm in mobile ad hoc networks using genetic algorithmic approach. In *Proceedings of GlobeCom*, 2002.
8. Turgut D, Turgut B, Elmasri R, Le TV. Optimizing clustering algorithm in mobile ad hoc networks using simulated annealing. In *Proceedings of Wireless Communications and Networking Conference*, 2003.
9. Roy A, Das SK. QM^{2}RP: a QoS-based mobile multicast routing protocol using multi-objective genetic algorithm. *Wireless Networks* 2004; **10**(3): 271–286.
10. Zitzler E, Laumanns M, Thiele L. SPEA2: improving the strength Pareto evolutionary algorithm. In *Evolutionary Method for Design, Optimisation and Control*, Giannakoglou K, Tsahalis D, Periaux J, Papailiou K, Fogarty T (eds.). International Center for Numerical Methods in Engineering (CIMNE): Barcelona, Spain, 2002.
11. Komolafe O, Sventek J. RSVP performance evaluation using multi-objective evolutionary optimisation. In *Proceedings of INFOCOM*, 2005.
12. Russell S, Norvig P. *Artificial Intelligence: A Modern Approach*. Prentice Hall: NJ, 2003.
13. Srinivas N, Deb K. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation* 1995; **2**(3): 221–248.
14. Basagni S. Distributed clustering for ad hoc networks. In *Proceedings of ISPAN*, 1999; 310–315.
15. Laumanns M. http://www.tik.ee.ethz.ch/pisa/selectors/spea2/spea2_c_source.html
16. Wang KH, Li BC. Group mobility and partition prediction on wireless ad-hoc networks. In *Proceedings of IEEE International Conference Communications*, 2002.
17. Bettstetter C, Resta G, Santi P. The node distribution of the random waypoint mobility model for wireless ad hoc networks. *IEEE Transactions on Mobile Computing* 2003; **2**(3): 257–269.

DOI: 10.1002/wcm

Authors' Biographies

Hui Cheng received his B.Sc. and M.Sc. degrees in computer science from Northeastern University, China in 2001 and 2004, respectively and his Ph.D. in computer science from The Hong Kong Polytechnic University, Hong Kong in 2007. He is currently a Postdoc Research Associate in the Department of Computer Science at University of Leicester, Leicester, U.K.

His research interests include evolutionary computation for dynamic optimization problems, mobile ad hoc networks, multicast routing, QoS mobile group communication. He has published over 20 technical papers in the above areas.

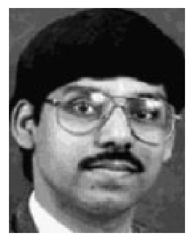
Jiannong Cao received his B.Sc. degree in computer science from Nanjing University, Nanjing, China in 1982, and his M.Sc. and Ph.D. degrees in computer science from Washington State University, Pullman, WA, U.S.A., in 1986 and 1990, respectively. He is currently a Professor in the Department of Computing at The Hong Kong Polytechnic University, Hong Kong. He is also the Director of the Internet and Mobile Computing Lab in the department. 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 computing, fault tolerance, and distributed software architecture and tools. He has published over 180 technical papers in the above areas.

Professor Cao is a senior member of China Computer Federation, a member of the IEEE Computer Society, the IEEE Communication Society, IEEE, and ACM. He is also a member of the IEEE Technical Committee on Distributed Processing, IEEE Technical Committee on Parallel Processing, IEEE Technical Committee on Fault Tolerant Computing. He has served as a member of editorial boards of several international journals, a reviewer for international journals/conference proceedings, and also as an organizing/program committee member for many international conferences.

Xingwei Wang received his B.Sc., M.Sc., and Ph.D. degrees in computer science from Northeastern University, China, in 1989, 1992, and 1998, respectively. He is currently a Professor in the College of Information Science and Engineering, Northeastern University, China.

His research interests include routing algorithms and protocols, QoS (Quality of Service) control schemes, fault-tolerance and survivability models, mobility management mechanisms and resource assignment methods in NGI (Next Generation Internet), IP over DWDM optical Internet and mobile Internet. He has published over 100 technical papers in the above areas.



Sajal K. Das is a Distinguished Scholar Professor of Computer Science and Engineering and the Founding Director of the Center for Research in Wireless Mobility and Networking (CReWMaN) at the University of Texas at Arlington (UTA). He is also a Visiting Professor at the Indian Institute of Technology (IIT), Kanpur and IIT Guwahati; Honorary Professor of Fudan University in Shanghai, China; and Visiting Scientist at the Institute of Infocomm Research (I2R), Singapore.

His current research interests include design and modeling of smart environments, sensor networks, security, mobile and pervasive computing, resource and mobility management in wireless networks, wireless multimedia, mobile Internet, mobile grid computing, biological networking, applied graph theory and game theory. He has published over 400 papers in international conferences and journals, and over 30 invited book chapters. He holds five US patents in wireless mobile networks, and coauthored the book 'Smart Environments: Technology, Protocols, and Applications' (John Wiley, 2005).

Professor Das received Best Paper Awards in IEEE PerCom'06, ACM MobiCom'99, ICOIN'02, ACM MSwiM'00 and ACM/IEEE PADS'97. He is also a recipient of UTA Academy of Distinguished Scholars Award (2006), University Award for Distinguished record of Research (2005), College of Engineering Research Excellence Award (2003), and Outstanding Faculty Research Award in Computer Science (2001

DOI: 10.1002/wcm

and 2003). He is frequently invited as keynote speaker at international conferences and symposia. He serves as the Founding Editor-in-Chief of *Pervasive and Mobile Computing (PMC)* journal (Elsevier), and Associate Editor of *IEEE Transactions on Mobile Computing*, *ACM/Springer Wireless Networks*, *IEEE Transactions on Parallel and Distributed Systems*, and *Journal of Peer-to-Peer Networking*. He is the founder of IEEE WoWMoM and co-founder of IEEE PerCom conferences. He has served as General or Technical Program Chair as well as TPC member of numerous IEEE and ACM conferences. He is a member of IEEE TCCC and TCPP Executive Committees.

Shengxiang Yang received his B.Sc. and M.Sc. degrees in automatic control and his Ph.D. in systems engineering from Northeastern University, China in 1993, 1996, and 1999, respectively. He was a Postdoctoral Research Associate in the Department of Computer Science, King's College London from October 1999 to October 2000. He is currently a Lecturer in the Department of Computer Science at University of Leicester, United Kingdom.

His research interests include evolutionary and genetic algorithms, artificial neural networks for combinatorial optimization problems, scheduling problems, dynamic optimization problems, and network flow problems and algorithms. He has published over 60 papers in books, journals, and conferences. He has co-guest-edited a special issue for the *Journal of Genetic Programming and Evolvable Machines* and has coedited a book *Evolutionary Computation in Dynamic and Uncertain Environments*, published in March 2007.

Dr Yang serves as the Area Editor, Associate Editor, and member of editorial board of three international journals. He is a member of the Working Group on Evolutionary Computation in Dynamic and Uncertain Environments, Evolutionary Computation Technical Committee, IEEE Computational Intelligence Society (CIS). He is a member of IEEE and ACM SIGEVO.

Wirel. Commun. Mob. Comput. 2009; **9**:759–771
DOI: 10.1002/wcm