An Entropy-based Weighted Clustering Algorithm and Its Optimization for Ad Hoc Networks

Yu-Xuan Wang
School of Commu. & Inform. Eng. , Nanjing Univ. of P. & T. , Nanjing 210003, CHINA
Email : logpie@gmail.com
Forrest Sheng Bao
Clinical School of Medical College, Nanjing University, Nanjing 210002, CHINA
Email : forrest.bao@gmail.com

Abstract—As a newly-proposed weighing-based clustering algorithm, WCA has improved performance compared with other previous clustering algorithms. But the high mobility of nodes will lead to high frequency of reaffiliation which will increase the network overhead. To solve this problem, we propose an entropy-based WCA(EWCA) which can prompt the stability of the network. Meanwhile, in order better to facilitate the optimal operation of the MAC protocol and stabilize the network structure, this paper presents an approach using Tabu Search to choose a near optimal dominant set. Consequently, less clusterheads are required to manage the network. Simulation study indicates that the revised algorithm(EWCA-TS) has improved performance respecting to original WCA, especially on clusterhead numbers and reaffiliation frequency. Furthermore, EWCA-TS can achieve these improvements with relatively low computational cost.

Index Terms—Ad-Hoc Networks, Entropy, Clustering, Reaffiliation, Tabu Search

I. INTRODUCTION

Unlike traditional wireless networks, the Ad-Hoc network is an infrastructureless network consisting of mobile autonomous moving nodes. Every node in the network performances as a router or a package forwarder. They interconnect with each other at the same peer level to enable the network function.

Although originally designed for military purpose, now it has been used in various civilian applications in the last decade. Since Ad Hoc networks differ to traditional hierarchical networks, its tremendous importance and the lack of rigorous methodologies motivate the in-depth research in this area.

Analogous to traditional cellular networks, the partitioning in Ad Hoc networks, known as clustering, is used to solve the inefficient use of power and bandwidth for every node to communicate directly. Each cluster elects one clusterhead, the upper layer node, to manage the cluster and coordinate with other clusters. The set of clusterheads is known as dominant set.

Several heuristic have been proposed to choose clusterheads in ad hoc networks, such as Highest-Degree heuristic[1] [2], Lowest-ID heuristic[3] [4] [5], Node-Weight heuristic [6] [7], etc.

The WCA(Weighted Clustering Algorithm) was firstly proposed by M. Chatterjee, S.K Das and D. Turgut [8]. A node is selected to be the clusterhead when it has the minimum weighted sum of four indices: the number of potential members; the sum of the distances to other nodes in its radio distance; the node’s average moving speed (where less movement is desired); and time of it being a clusterhead(this takes battery life into account).

When a node has moved out of its cluster, it will firstly check whether it can be a member of other clusters. If such a cluster exists, it will detach from its current cluster and attach itself to that cluster. The process of joining a new cluster is known as reaffiliation. If the reaffiliation fails, the whole network will recall the clusterhead election routine. One disadvantage of WCA is high reaffiliation frequency when network scenario changes very fast. High frequency of reaffiliation will increase the communication overhead. Thus, reduce the reaffiliation is necessary in ad hoc networks.

In practice, to better facilitate the management of the network, a good dominant set is required. But the underlying optimal assignment problem is NP-Hard [9]. We can apply an approximation algorithm to obtain a near optimal solution.

In this paper we propose two improvements of the WCA. Firstly, we replace one indices, the average moving speed of nodes, by the entropy[10] of local networks. This approach can reduce the frequency of reaffiliation. Secondly, we use Tabu Search to optimize the election routine which forms the near optimal dominant set.

The rest of this paper is organized as follows. In section II, we present the improved WCA algorithm combined with entropy(EWCA). In section III, we propose the optimized EWCA using Tabu Search(EWCA-TS). Simulation results are presented in section IV while conclusions are offered in section V.

II. ENTROPY-BASED WCA(EWCA)

As we mentioned before, the WCA calculates the weighted sum using (1)

\[ W_v = w_1 \Delta_v + w_2 D_v + w_3 M_v + w_4 P_v \]  

where \( \Delta_v \) is the degree-difference between the number of a node’s member and the number of nodes it can handle under ideal condition, \( D_v \) is the sum of the distances of the members to the clusterhead, \( M_v \) is the average speed of the node, and \( P_v \) is the accumulative time of a node being a clusterhead. \( w_1, w_2, w_3 \) and \( w_4 \) are the corresponding weighing factors. The node
with the minimum $W_v$ is chosen to be the clusterhead. Once a node becomes the clusterhead, either that node or its members will be marked as “considered”. Then the election process interacts on all “unconsidered” nodes. (Initially, all nodes are “unconsidered”). The election algorithm will terminate once all the nodes have been considered.

Higher reaffiliation frequency will lead to more recalculation of the cluster assignment, hence, increase the communication overhead. This phenomena invokes us looking for better criteria of clusterhead election in order to form a more “stable” network.

In the paper of Beongku An and Symeon Papavassiliou[10], they introduced an entropy-based model for evaluating the route stability in Ad Hoc networks. Entropy presents uncertainty and a measure of the disorder in a system. So we consider it a better methodology to measure the stability and mobility of the Ad Hoc network.

Denote the positions of node $m$ and $n$ at time $t$ as $\mathbf{p}(m, t)$ and $\mathbf{p}(n, t)$ respectively. The positions of nodes are calculated periodically during a time interval $\Delta t$. The relative position between node $m$ and $n$ at time $t$ is defined as:

$$\mathbf{p}(m, n, t) = \mathbf{p}(m, t) - \mathbf{p}(n, t)$$  \hspace{1cm} (2)

The variable feature of the system(network) considered here is the relative position between two nodes, $m$ and $n$. It is defined as

$$a_{m,n} = \frac{1}{N} \sum_{i=1}^{N} |\mathbf{p}(m, n, t_i)|$$  \hspace{1cm} (3)

where $t_i$ refers to the moment of the $i$-th calculation and $N$ is the number of discrete times $t_i$ within time interval $\Delta t$.

Then we can define the entropy. Denote the entropy of node $m$ as $H_m(t, \Delta t)$. We have

$$H_m(t, \Delta t) = \frac{-\sum_{k \in F_m} P_k(t, \Delta t) \log P_k(t, \Delta t)} {\log C(F_m)}$$  \hspace{1cm} (4)

where $P_k(t, \Delta t) = (a_{m,k}/\sum_{i \in F_m} a_{m,i})$.

In the relation by $F_m$ we denote the set(or any subset) of the neighboring nodes of node $m$, and by $C(F_m)$ the cardinality (degree) of set $F_m$.

In this paper, $F_m$ refers to the local network centered by node $m$, hence $H_m$ presents the stability of this local network, the set of all nodes that can reach node $m$ in one hop. It should be noted that the entropy, as defined here, is small when the change of variable values in the given region is severe and large when the change of the values is small[10].

We replace one term of (1), the average speed of nodes($M_v$), by the entropy defined in (4). Hence, the new formula to calculate $W_v$ becomes:

$$W_v = c_1 \Delta v + c_1 D_v + c_3 (-H_v) + c_4 P_v$$  \hspace{1cm} (5)

Simulation study given later will indicate that this replacement can significantly reduce the frequency of reaffiliation.

III. THE OPTIMIZED EWCA(EWCA-TS)

The idea of Tabu Search(TS) was firstly proposed by Fred Glover[11][12]. Inspired by human intelligence procedure, TS has achieved great success on combinatorial optimization problems. In order to avoid the entrapment in local minima, TS introduces a policy to forbidden certain classified moves. Attributes with good fitness value are marked in the tabu list to prevent cycling so that the solution space can be enlarged. TS undertakes an adaptive tabu list and associated tabu strategy to guarantee the diversity of search. Meanwhile, to prevent the loss of promising solutions, TS introduces Aspiration Criterion. If the fitness of a solution advances “best so far” state, we ignore its tabu status and moreover, we adopt it as the current solution directly.

The process of the simple TS is given as follows.

Step 1: Randomly generate an initial solution $s$. Empty the tabu list. Denote the best fitness value as $f_s = \text{fitness}(s)$ and the optimal solution $s^* = s$.

Step 2: Generate specified number of neighboring trial solutions $s'$ of $s$. Evaluate the quality of neighboring solutions using fitness function. Choose solutions of high fitness value as candidates.

Step 3: For candidate $s'$, check whether the aspiration criterion is satisfied.

If not, jump to step 4.

If satisfied, set current solution $s$ and the optimal solution $s^*$ as $s'$ and the best fitness value $f_s = \text{fitness}(s')$. Meanwhile, add the associated attribute of $s'$ into the tabu list and set its tenure as the tabu length. The tenures of all existing attributes will be updated either.

Step 4: Check the tabu status of associated attributes of all candidates. Choose the non-tabued solution with best fitness value as candidates.

Step 5: Repeat above steps until specified iteration time is reached.

Neighborhood structure, candidates, tabu length, tabu attributes and aspiration criterion are key factors that affect the performance of Tabu Search algorithm. Neighborhood function reflects the aspect of local neighborhood search. The purpose of tabu list is to avoid cycling search. Aspiration criterion is an award to good candidates, preventing loss of promising solutions. Due to TS has adaptive memory ability, aspiration criterion and tolerance to bad solutions, it has a strong “climbing” ability which enhances the probability of obtaining a global optimum.

In practice, in order better to balance the network load, prompt the stability of clusters and prolong the lifetime of the network, we need to choose the best dominant set in which each clusterhead handles the maximum possible number of mobile nodes. Thus, it will better facilitate the optimal operation of the MAC protocol and reduce the number of clusterheads. Meanwhile, the overhead on network communication will be reduced.

As we discussed before, the selection of the optimal dominant set is an NP-hard combinatorial optimization problem. Some meta-heuristic algorithms, such as SA[13], PC[14] and GA[15] are applied onto it.

In this paper, we propose a TS approach to optimize the EWCA for further improvement on performance, especially
in terms of the reaffiliation and the number of clusterheads.

The pseudo-code of EWCA-TS is given as follows. The goal of TS is to minimize the objective function \( F(s) = \sum_{v \in s} W_v \).

**Algorithm 1 EWCA-TS**

1: Randomly generate an initial solution \( s \), \( s^* \leftarrow s \). \( F^* \leftarrow F(s) \).
2: repeat
3:   \( \text{Generate } NS \) neighboring solutions of \( s \) and sort them by the fitness value.
4:   Select \( CS \) best candidate solutions from sorted neighboring solutions.
5:   for each solution \( s_i \) in \( CS \) sorted candidates do
6:     if \( F(s_i) < F^* \) then
7:       \( s^* \leftarrow s_i \), \( F^* \leftarrow F(s_i) \).
     Tabu the associated attribute of \( s_i \).
     Set its tenure to \( TL \).
9:   else if \( s_i \) has not been tabued then
10:     \( s \leftarrow s_i \).
     Tabu the associated attribute to \( s_i \).
     Set its tenure to \( TL \).
11: end if
12: end for
13: Update \( Tabu \_List \).
14: until \( MAX \_IT \) iteration time is reached
15: Dominant Set (Clusterhead) \( \leftarrow s^* \).

Where \( s \) is the current solution (dominant set), \( s^* \) is the best known solution, \( F^* \) is the best fitness value found so far. \( TL \) is short for tabu length which limits the maximum tenure of tabued attributes in the tabu list. When the tenure of a tabued attributes is 0, this attribute will be erased. \( CS \) stands for the size of the candidate set. \( NS \) means the size of neighborhood solution set from which the algorithm selects \( CS \) candidates with highest fitness value. \( MAX \_IT \) is the maximum iteration time.

A good neighborhood structure can effectively improve the quality of solutions and accelerate the convergence. We utilize the following neighborhood solution generating procedure.

**Step 1:** Randomly discard the clusterhead property of two clusters, \( c_1 \) and \( c_2 \). Meanwhile, members of \( c_1 \) and \( c_2 \) lost their affiliations. Prohibit \( c_1 \) and \( c_2 \) to be clusterhead in newly-generated dominant set.

**Step 2:** For all nodes, if one node is neither a clusterhead nor a cluster member, and its degree is less than predefined maximum degree, this node becomes a clusterhead.

**Step 3:** Set all other unassigned nodes to be clusterheads.

New dominant set generated by this procedure contains both the part of original dominant set and random components. This will make TS better explore the space of potential solutions. Note: tabu attribute in the tabu list is not the solution (dominant set) itself but the two discarded clusterheads \( c_1 \) and \( c_2 \).

IV. SIMULATION STUDY

This section contains two parts. The first part evaluates the effect of replacing average speed by entropy in terms of reducing reaffiliation frequency. The later part evaluates the optimized EWCA using Tabu Search.

**A. Simulation study of EWCA**

In order to demonstrate the influence of entropy, we set the weighing factor in original WCA as \( w_1 = w_2 = w_4 = 0, w_3 = 1 \) and the weighing factor in EWCA as \( c_1 = c_2 = c_4 = 0, c_3 = 1 \). Thus, only contrast the effectiveness of the entropy and the average speed in terms of reducing reaffiliation.

We simulate a system of 30 nodes on a 100 × 100 grid. The relationship between reaffiliation frequency and transmission range is illustrated in fig. 1. The nodes could move in all possible directions with displacement varying from 0 to a maximum value (\( max \_disp \)). Considering the reaffiliation phenomena is obvious in networks with high-speed moving nodes, we choose the maximum displacement as 30. Fig. 1 indicates that reaffiliation per unit time of EWCA descend obviously in transmission range 20 to 40, compared with original WCA. EWCA and WCA has similar reaffiliation per unit time in other intervals. We explain the reason as follows. When transmission range is small, every clusterhead only manages few nodes. So the reaffiliation frequency is not high. While the transmission range becomes large, one cluster can cover a large area. Thus, it is not easy for a node to move out of the transmission range of its clusterhead.

![Fig. 1. Reaffiliation per unit time vs. Transmission range, max_disp=30, number of nodes=30](image)

Since EWCA has better performance in transmission range 20 to 40, we take a deeper investigation to this scenario. The relationship between reaffiliation frequency and maximum displacement is illustrated in fig. 2 where the number of nodes is 30 and transmission range is 30. We observe that WCA and EWCA have similar reaffiliation frequency when \( max \_disp \) is small. The larger the \( max \_disp \) is (the faster nodes move), the more obvious the reaffiliation frequency difference between
TABLE I
PARAMETERS USED IN EWCA-TS

<table>
<thead>
<tr>
<th>Metric</th>
<th>N</th>
<th>WCA</th>
<th>WCA-TS</th>
<th>EWCA</th>
<th>EWCA-TS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>20</td>
<td>118.79</td>
<td>97.11</td>
<td>57.12</td>
<td>46.33</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>166.37</td>
<td>134.80</td>
<td>66.18</td>
<td>52.79</td>
</tr>
<tr>
<td>number of</td>
<td>20</td>
<td>8.21</td>
<td>6.45</td>
<td>7.91</td>
<td>6.41</td>
</tr>
<tr>
<td>clusters</td>
<td>30</td>
<td>9.35</td>
<td>7.38</td>
<td>9.08</td>
<td>7.04</td>
</tr>
<tr>
<td>reaffiliation</td>
<td>20</td>
<td>2.05</td>
<td>1.44</td>
<td>1.87</td>
<td>1.28</td>
</tr>
<tr>
<td>frequency</td>
<td>30</td>
<td>3.21</td>
<td>2.15</td>
<td>2.94</td>
<td>1.88</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>4.32</td>
<td>2.81</td>
<td>3.85</td>
<td>2.47</td>
</tr>
</tbody>
</table>

WCA and EWCA becomes. EWCA can reduce about 30% reaffiliation frequency of WCA.

B. Simulation study of EWCA-TS

In this subsection, we present the performance of EWCA-TS. The comparison on number of clusters among EWCA-TS, EWCA and WCA are illustrated in fig. 3.

![Fig. 3. Average number of clusters, number of nodes=30, tx_range=30](image)

We compare three metrics: costs, average number of clusters and reaffiliation frequency. The results are listed in Table II. Each entry in the table is an average of 20 runs. N means the number of nodes, varying from 20 to 40. WCA-TS means optimized WCA using TS. The reason of taking WCA-TS into account is to evaluate the effect of tabu search with original WCA. The cost is defined as \[ \sum_{v \in S} W_v \]. Weighing factors used in EWCA and EWCA-TS are 0.6, 0.05, 0.3, 0.05 respectively. Weighing factors used in WCA and WCA-TS are 0.7, 0.05, 0.2, 0.05 respectively. Parameters in TS are listed in Table I.

The performance of four approaches(WCA, WCA-TS, EWCA and EWCA-TS) can be viewed in Table II. EWCA has similar number of clusters with WCA, but less reaffiliation frequency. Tabu search can optimize the procedure of clusterhead election. Both the number of clusters and the reaffiliation frequency are reduced by using the TS. EWCA-TS and WCA-TS has much less reaffiliation frequency and number of clusters, compared with their non-TS-optimized versions. In all four approaches, EWCA-TS achieved the best performance.

Although TS is a stochastic optimization algorithm based on Monte-Carlo method, we note that EWCA-TS and WCA-TS achieved the results listed in Table II only after calling the neighborhood solution generating function 50×50=2500 times. Table III presents the comparison on reducing the costs(average value of the objective function) between WCA-SA [13] and WCA-TS after calling the neighborhood solution generating function 2500 times. In SA, we set the initial temperature \( T_0 = 50 \), the constant used to reduce the temperature \( \alpha = 0.9 \) and the number of sample solutions checked before reducing the temperature \( L = 50 \). We can clearly observe that, under the same circumstance, TS can achieve promising results with less computational costs compared with SA. So, we consider that it’s more suitable for utilizing TS to optimize the dominant set election routine in real applications since it can get a good solution quickly.

V. Conclusion

This paper proposed an entropy-based WCA(EWCA) and its optimization(EWCA-TS). EWCA mainly focuses on reducing the reaffiliation caused by high-speed moving nodes. For enhancing the performance in diverse metrics, such as longer battery life, lower frequency of network assignment, we use tabu search to optimize EWCA and achieved significant performance promotion with relatively low computational costs, especially with regard to average number of clusters and reaffiliation frequency. Consequently, each clusterhead can maximize the number of its members and the network can stabilize its structure much longer. The simulation study shows these goals can be achieved.

REFERENCES


