A New Cloudlet Placement Method Based on Affinity Propagation for Cyber-Physical-Social Systems in Wireless Metropolitan Area Networks

KAI PENG1, (Member, IEEE), XINGDA QIAN1, BOHAI ZHAO1, KEJIA ZHANG2, AND YANG LIU3, (Member, IEEE)

1College of Engineering, Huaqiao University, Quanzhou 362021, China
2School of Mathematical Science, Heilongjiang University, Harbin 150080, China
3School of Computer Science and Software Engineering, University of Science and Technology Liaoning, Anshan 114051, China

Corresponding author: Yang Liu (liuyang_inas@163.com)

This work was supported in part by the National Science Foundation of China under Grant 61902133, in part by the Natural Science Foundation of Fujian Province under Grant 2018J05106, in part by the Education and Scientific Research Projects of Young and Middle-Aged Teachers in Fujian Province under Grant JZ160084, in part by the Scientific Research Foundation of Huaqiao University under Grant 14BS316, and in part by the Quanzhou Science and Technology Project under Grant 2015Z115.

ABSTRACT Cyber-Physical-Social Systems (CPSS) integrates the cyber, physical and social spaces together. There are a large number of mobile users in CPSS that need low latency services. Fortunately, mobile edge computing (MEC) is a novel technology which can provide such services. The edge server plays a key role in MEC, but how to manage the edge server is an important challenge. For one thing, the number of cloudlets and the resource are limited. For another, the number of mobile devices (MDs) is very large and randomly distributed. And thus, how to determine the suitable number of cloudlets while serving the maximum number of MDs is significant. To this end, a new cloudlet placement method based on improved Affinity Propagation (AP) algorithm is proposed to solve the above problems. More specially, the improved AP algorithm can obtain the least number of cloudlets while covering the largest number of MDs. In addition, the load balancing strategy is used to ensure that the load of each cloudlet maintains a balanced state. Last but not the least, our proposed method can be used in scenarios where users move.

INDEX TERMS Mobile edge computing, cloudlet placement, affinity propagation algorithm, load balancing.

I. INTRODUCTION

Cyber-Physical-Social Systems (CPSS) integrates the cyber, physical and social spaces together. One of the ultimate goals of CPSS is to make our lives more convenient and intelligent by providing prospective and personalized services for users [1]–[6]. In CPSS, there are a large number of mobile devices (MDs) and some applications running in MDs may have various constraints, (e.g., time constraint, sequential restriction), which requires to be served in low latency [7]–[13]. Meanwhile, wireless sensor network and RFID system are also generating massive data, which further increases processing pressure for cloud [14]–[23].

Fortunately, the new computing architecture named Mobile Edge Computing (MEC) that can compute the offloaded tasks at the edge of the network is proposed to reduce the latency [24]–[29]. Generally, MEC can be simply divided into three layers (i.e., cloud, cloudlets and MDs) and the core of which is edge server, which can provide various types of services [30]. In addition, cloudlet is a typical edge server with advantages in economy and utility, which makes the placement of cloudlets in network much easier than large servers [31], [32]. On the one hand, cloudlet provides fast and low-latency service for the mobile users. On the other hand, it also releases the burden of the remote cloud as it performs a large number of user tasks as an intermediate node between users and the cloud. This architecture of numerous nodes provides better computing capacity and is more suitable to...
be applied to large-scale networks, for example, wireless metropolitan area networks (WMAN) [33], [34].

However, WMAN is a wide area network consisting of a large number of wireless access points for users to obtain service they need, and the distribution of these users is often complex and unpredictable. Thus, we must put a large number of cloudlets into the network to ensure most of the users can be served effectively. But how to place these cloudlets reasonably faces big challenges.

The placement of the cloudlets in WMAN has been well studied by authors in [35], [36]. It is assumed that cloudlets and MDs are both fixed. However, in reality, MDs are usually in mobile state. Therefore, the static placement method may not be directly used for the dynamic environment.

To solve this problem, researchers have proposed some feasible methods for cloudlet placement in the dynamic environment (to list some here [37], [38]). However, how to minimize the number of cloudlets while serving more MDs, meanwhile, making each cloudlet workload balanced is a very important issue.

In view of this, in this paper, we focus on how to deploy the cloudlets with efficiency. We propose a method called Affinity Propagation with Optimal Preference (APOP) for cloudlet placement. Our main contribution is summarized as follows.

1) We construct the set of the candidate preference to select the optimal preference. More specifically, the selected preference of AP algorithm can obtain the least number of cloudlets while covering the largest number of MDs.

2) The load balancing strategy is proposed to equilibrium the number of MDs covered by cloudlets to ensure that the load of each cloudlet maintains a balanced state.

3) Compared to other methods, such as K-Means and Mini Batch K-Means, our proposed APOP method is more effective and efficient.

The remaining parts of this paper are organized as follows. In section II, the related work is presented. Then, the network model and the problem formulation of cloudlet placement, cloudlet movement and load balancing are introduced in section III. The APOP algorithm for the cloudlet placement is presented in section IV. Section V gives an experimental evaluation and discussion. Section VI gives the conclusion of this paper and future work. Some key terms and descriptions are shown in TABLE 1 to get a more concise and better expression.

### II. RELATED WORK

In recent years, due to the convenience of MEC services, an increasing number of CPSS applications have been deployed on the edge side of network. Then, mobile users can get more high-quality services through cloudlets in CPSS. However, how to place the cloudlet reasonably is still an important but difficult problem.

Some of researchers focus on the user allocation of cloudlets. Lai et al. [39] formulated the user allocation problem into a bin packing problem, and then proposed an optimal method to solve the problem in the basis of the Lexicographic Goal Programming technique. Guo et al. [40] focused on the edge clouds at the candidate locations and allocated the mobile users to these clouds. Additionally, a novel decentralized algorithm for finding a Nash equilibrium based on game theory was proposed to address the edge user allocation problem by He et al. [41].

Wang et al. [30] formulated the placement of edge servers as a multi-objective optimization problem. Not only the delay of MDs is optimized, but also the workload of cloudlets is balanced. Chen et al. [34] designed two efficient methods for different situations, and the placement of the cloudlet in their study are determined based on historical statistics. Jia et al. [35] focused on the problem of cloud placement and user allocation in WMAN. They proposed a new method to decide how to deploy the cloudlets and assign the users to the cloudlets. The time consumption of MDs can be effectively reduced by using their method. Xu et al. [36] formulated the problem as integer linear programming and solved the problem of capacitated cloudlet placement and dynamic request assignment. However, the above placement methods may not be applied to the dynamic environment directly because the mobile users are often under moving state. That means the mobility of users should be taken into consideration for the cloudlet placement method.

Lai et al. [42] took into consideration the dynamic Quality of service (QoS) levels for mobile users and turn it into a dynamic QoS edge user allocation problem. Correspondingly, an optimal approach is proposed to find solutions which maximizes mobile users’ overall Quality of Experience. In addition, clustering algorithms are widely used for cloudlet deployment. Liang and Li [43] proposed a location-aware service deployment method based on K-Means for cloudlets which divided MDs into multiple MD clusters according to the geographical location of MDs and then deployed service instances onto the cloudlets closest to the centers of MD clusters. Shen et al. [37] proposed a dynamic method for cloudlet placement with clustering algorithm. They used K-Means algorithm to get the initial position of cloudlets. Then, the movement strategy is used in their placement method. Zhang et al. [38] proposed an adaptive clustering method based on the location of MDs. Additionally, they used tracking method to collect the initial position and the target

### TABLE 1. Key terms and descriptions.

| Terms | Description |
|-------|-------------|
| \( n \) | The total number of MDs |
| \( K \) | The number of cloudlet |
| \( n_c \) | The number of candidate preference |
| \( K_i \) | The number of MDs covered by cloudlet \( cl_i \) |
| \( M \) | The set of MDs \( M = \{m_1, m_2, ..., m_n \} \) |
| \( CL \) | The set of cloudlets, \( CL = \{cl_1, cl_2, ..., cl_k \} \) |
| \( p_{d_i}(t) \) | The position of \( d_i \) at time \( t \) |
| \( p_{cl_i}(t) \) | The position of \( cl_i \) at time \( t \) |
| \( R(t) \) | The coverage radius for \( cl_i \) at time \( t \) |
| \( D_j(t) \) | The set of MDs covered by \( cl_j \) at time \( t \) |
position of the cloudlet to realize the dynamic changes. The access latency of MDs and the number of deployed cloudlet servers are optimized.

Different from the existing researches, in this paper, we focus on cloudlet placement for CPSS in WMAN based on improved AP algorithm. Both the mobility of the users and load balancing of cloudlets are taken into consideration, our objective is to find the best number of cloudlets, which can cover the largest number of MDs while keeping the load balancing of the cloudlets. More specifically, the load balancing strategy has been considered in the initial state and the state after moving.

III. PRELIMINARY
In this section, the system model in WMAN is presented. Then, the problem formulation of how to place the cloudlet for optimizing the CPSS is described in details.

A. THE SYSTEM MODEL
The structure of MEC can be simply divided into three layers (i.e., cloud, cloudlets and MDs), which is shown in Fig. 1. Compared to the cloud, the cloudlet is placed on the edge of the network, which is closer to users. And users can interact with the cloudlet through wireless access points. Then, with the shorter physical distance and better computing capacity, the service latency can be reduced greatly. Furthermore, when the cloudlet is placed in a more critical position, its resource utilization may be improved significantly, which means the placement method of the cloudlet may play a vital role in CPSS to provide the users with high-quality service [6], [8].

FIGURE 2. The dynamic placement process of the cloudlet.

In Fig. 2, due to the numerous MDs in CPSS, the density of MDs should be taken into consideration. The density of MDs can be broadly divided into three categories, the Dense Cluster of MDs (DCMDs), the Sparse Cluster of MDs (SCMDs) and the Discrete MD (DMD). After changing the location of the cloudlet according to the distribution of MDs, more MDs will be covered by the cloudlet. Then, not only more users can enjoy high-quality services, but also the resource utilization of the cloudlet will be improved seriously.

Thus, it is of considerable significance to propose a feasible algorithm that can place the cloudlet in the optimal position according to the movement locus of MDs.

B. PROBLEM FORMULATION
In this paper, we mainly focus on how to make the cloudlet serve the MDs with less latency and improve the resource utilization of the cloudlet concurrently. The method for the cloudlet placement is presented firstly. Then, considering the users’ mobility and the random assignment of tasks, a movement strategy and a load balancing method are presented subsequently.
1) CLOUDLET PLACEMENT METHOD
It’s assumed that the cloudlet service scope is circular area and is denoted as \( R_j(t) \). The MD in the circular area means that the MD is provided with better service by the cloudlet. The positions of the cloudlet and MDs at initial time \( t \) are denoted as \( P_{cl_j}(t) \) and \( P_{d_i}(t) \) respectively. Thus, the set of MDs covered by the \( j_{th} \) cloudlet is denoted as \( D_j(t) = \{ d_{j1}(t), d_{j2}(t), \ldots, d_{jt}(t) \} \), which are computed by using (1) and the number of MDs in \( D_j(t) \) is denoted \( k_j \).

\[
D_j(t) = \{ d_{j(t)} \mid \text{dis}(P_{d_i}(t), P_{cl_j}(t)) \leq R_j(t) \} \tag{1}
\]

where \( d_{j(t)} \) denotes the \( i_{th} \) MD covered by the \( j_{th} \) cloudlet, the distance between MD and cloudlet is denoted as \( \text{dis}(P_{d_i}(t), P_{cl_j}(t)) \), which is computed as

\[
\text{dis}(P_{d_i}(t), P_{cl_j}(t)) = \sqrt{(x_{d_i} - x_{cl_j})^2 + (y_{d_i} - y_{cl_j})^2} \tag{2}
\]

The total number of MDs covered by the cloudlets at initial time \( t \) is denoted as \( T_d(t) \), which is computed by using (3).

Considering the computing resource of cloudlet is limited, we define the largest number of MDs that one cloudlet can provide service to as \( L \) and the largest number of cloudlets as \( B \). Then, the capacity of cloudlet to accommodate MDs should satisfy the following (4) and (5).

\[
T_d(t) = \sum_{j=1}^{m} k_j(t) \tag{3}
\]

\[
k_j \leq L \tag{4}
\]

\[
T_d(t) \leq B \times L \tag{5}
\]

2) THE NUMBER OF CLOUDLET
If the MDs are served by too many or too few cloudlets, the computing resource will be unmatched and the latency of service may be in a high standard. Specifically, when the number of cloudlets is too small, many MDs may get a poor service, which will lead to intolerable latency. And when too many cloudlets are placed at WMAN, MDs can get better service, but the cost of the cloudlet will become expensive and the resource utilization of the cloudlet will be under a low standard. Thus, how to compute the optimal number of the cloudlet according to MDs is still important.

AP method can generate the optimal number of clusters with the preference. Once we get the optimal preference, the AP algorithm can compute the optimal number of cloudlets according to the number of MDs, which ensures the least number of cloudlets can cover MDs as many as possible.

3) THE CLOUDLET MOVEMENT STRATEGY
At time \( t \), the cloudlet \( cl_j \) placed at the initial position \( P(x, y) \) after the clustering algorithm is denoted as \( P_{cl_j}(t) \). After the movement of MDs, the set of MDs covered by the \( P_{cl_j}(t') \) is denoted as \( P_{d_i}(t') \) at time \( t' \). Additionally, the potential cloudlet position \( P'(x', y') \) computed by clustering algorithm satisfies the strategy of cloudlet placement and the set of MDs covered by the \( P'_{cl_j}(t') \) is denoted as \( D_j(t') \). It’s assumed that in the region with \( P \) as the center and \( T \) as the radius, if the cloudlet \( cl_j \) at position \( P' \) can serve more MDs than initial position \( P \), the cloudlet is suitable to move from \( P \) to \( P' \).

4) THE LOAD BALANCING STRATEGY FOR CLOUDLET
The number of MDs covered by different cloudlets is often different, which leads to a significant waste of the resource. Furthermore, once the task load exceeds the capability of the cloudlet, the latency will become intolerable. In order to settle this problem, the load balancing strategy is proposed.

IV. AFFINITY PROPAGATION WITH OPTIMAL PREFERENCE (APOP) FOR CLOUDLET PLACEMENT
In this section, the method for the problem of the cloudlet placement is described in details. In order to select the position of cloudlet, the Affinity Propagation with Optimal Preference (APOP) algorithm is proposed firstly. Unlike K-Means [44], the Affinity Propagation (AP) can compute the optimal number of the cloudlet, which is determined by the preference. Thus, we propose a method to select the optimal preference by creating the set of candidate preference. Then, the load balancing method is introduced. Finally, a dynamic optimization strategy considering the movement of MDs is introduced.

A. PRELIMINARY
The process of our proposed method APOP can be summarized as follows and the execution order of algorithms corresponds to the process of APOP.

1) PROCESS 1: DECIDE THE NUMBER OF CLOUDLETS
The AP algorithm has an important parameter: preference, which decides the number of clusters. And the optimal preference is selected from the candidate set of preference, which generates the least number of clusters. More specifically, the optimal number of cloudlets can not only improve resource utilization, but also provide the low latency service. If the AP algorithm uses the random preference, the number of cloudlets may be huge, which is unreasonable. Thus, it is significant to choose the optimal preference.

2) PROCESS 2: GET THE CENTER POSITION OF MDs
The AP algorithm depends on the two messages of availability and responsibility to decide which data point as exemplar. In this way, the center position of MDs where can deploy the cloudlet is computed.

3) PROCESS 3: THE STRATEGY OF LOAD BALANCING
When cloudlet is deployed, we consider the efficiency of the cloudlet due to the limit of computing resources. That is to say, we should balance the number of MDs covered by one cloudlet ensuring the efficiency of cloudlet. When the position of the cloudlet is changed, the load balancing strategy should be used again.
4) PROCESS 4: THE STRATEGY OF MOVEMENT
In reality, the MDs are moving randomly. If the position of cloudlet is fixed, some MDs after moving may lose the service provided by the cloudlet and have the intolerable latency. So, we use the strategy of movement to update the position of cloudlet based on the movement of devices.

B. THE BASIC STEP OF APOP
AP algorithm is a high-performance clustering algorithm, which was first proposed in 2007 [45]. Compared to the popular clustering algorithms, the AP can compute the number of clusters based on the data set and preference. But, the original AP algorithm is hard to get an optimal preference for the data set. So, the algorithm of APOP is proposed to have a better clustering result for solving the problem of cloudlet placement.

In this algorithm, each data point can be seen as the potential exemplar, which represents the center point of the clusters. \( X = \{x_1, x_2, \ldots, x_i\}, i \in \{1, 2, \ldots, n\} \) represents the set consisting of \( n \) data points. Firstly, the similarity matrix \( S = [S_{ij}]_{mn}, i, j \in \{1, 2, \ldots, n\} \) is constructed, where \( S_{ij} \) shows the level of similarity between \( X_i \) and \( X_j \) [46]. If \( i = j \), the similarity \( S_{ij} \) of data point \( x_i \) is represented by the preset exemplary preference. The similarity is computed by Euclidean distance (6):

\[
S_{ij} = -|x_i - x_j|^2
\]

Then, AP algorithm takes each data points as potential exemplar and recursively transmits the two types of messages, responsibility \( r(i, j) \) and availability \( a(i, j) \) to find high-performance exemplars. The exemplar preference \( p \) is the similarity \( S_{ij}, i = j \), (i.e., \( S_{ii} = p \)). Here come the important parts of AP algorithm:

1) RESPONSIBILITY
In order to find the exemplar, the data point \( x_i \) sees all the other data points as potential exemplars and \( x_i \) sends the responsibility information \( r(i, j) \) to the candidate exemplar \( x_j \) [45]. And its process diagram is shown in Fig. 3. More specifically, the responsibility information \( r(i, j) \) describes the fitness of data point \( x_j \) as the clustering center of data point \( x_i \) and \( x_j \) is selected as the candidate exemplar. The responsibility information \( r(i, j) \) is computed as follows [45], [47]:

\[
r(i, j) \leftarrow s_{ij} - \max\{a(i, \hat{j}) + s_{ij}, \forall i, j\}
\]

If the value of \( r(i, j) \) is higher, it means that \( x_j \) is more likely to serve as the exemplar of \( x_i \) and its performance may be better.

2) AVAILABILITY
When \( x_i \) sends the responsibility information \( r(i, j) \) to \( x_j \), \( x_j \) also sends availability information \( a(i, j) \) to \( x_i \) to confirm if \( x_i \) will choose \( x_j \) as its exemplar [45]. And its process diagram is shown in Fig. 4. If \( x_i \) selects other data points \( x_j, i \neq j \), the availability \( a(i, j) \) is computed by using (8):

\[
a(i, j) \leftarrow \min\{0, r(i, j)\} + \max\{0, r(i, k)\}, i \neq j
\]

And its process diagram is shown in Fig. 4. If \( x_i \) selects other data points \( x_j, i \neq j \), the availability \( a(i, j) \) is computed by using (8):

\[
a(i, j) \leftarrow \min\{0, r(i, j)\} + \max\{0, r(i, k)\}, i \neq j
\]

In all, the availability \( a(i, j) \) and responsibility \( r(i, j) \) is updated recursively by (7), (8) and (9) accordingly. The value of availability \( a(i, j) \) and responsibility \( r(i, j) \) getting higher means data point \( x_j \) is more likely to be the candidate exemplar and the probability of \( x_i \) belonging to the clustering with \( x_j \) as the clustering center is also greater. According to this recursive computing process, the candidate exemplars are selected finally.

3) THE EXEMPLARS WITH HIGH-PERFORMANCE
The set of candidate exemplar \( c = \{c_1, c_2, \ldots, c_m\} \) represents the center points of the clustering. According to the evaluation of responsibility \( r(i, j) \) and availability \( a(i, j) \), the data point \( x_j \) is selected as the exemplar of \( x_i \), denoted as \( c_i = j \). In order to select the high-performance exemplar from the set of candidate \( c \), a way to estimate the performance of exemplar for the clustering is needed. The sum of the similarities between each data point can estimate the effect of candidate exemplars \( c \) for the whole clustering, and we call it net similarity [45]. It is represented by the \( CluSim(c) \), which is computed as

\[
CluSim(c) = \sum_{i=1}^{n} S_{ic_i} + \sum_{j=1}^{n} \xi_{jc_i}
\]
where $S_{ij}$ represents the data point $x_i$ choosing data point $x_j$ as an exemplar. In order to avoid the condition of $x_i$ choosing data point $x_j$ as an exemplar, $\text{CluSim}(c)$ is constrained by an exemplar-consistency constraint $\xi_{ij}$, (11),

$$
\xi_{ij} = \begin{cases} 
-\infty & \text{if } c_j \neq j, \text{but } \exists i : c_i = j \\
0 & \text{otherwise}
\end{cases}
$$

(11)

In (11), it means that if $c_j = j$, the data point $x_j$ must choose itself as exemplar. If not, the $-\infty$ is used to punish. Then, the set of high-performance exemplars are selected by maximizing the net similarity $\text{CluSim}(c)$.

4) SELECT THE OPTIMAL PREFERENCE

The preference of AP algorithm controls the number of clusters in the clustering. So, how to select an optimal preference according to the number of data set is important. Firstly, the set of candidate preferences is constructed, which is denoted as $P$, $P = \{p_1, p_2, \ldots, p_m\}$, where $m$ represents the preset parameter of the number of candidate preferences. According to [46], [48], during the process of updating the two types of information $r(i, j)$ and $a(i, j)$, the exemplar preference is also changing between different data points like similarity. So, the optimal preference should be chosen between $[p_{\text{min}}, p_{\text{max}}]$ and consider the bounds of the value of the similarity matrix.

$$
p_{\text{min}} = \min_{i \neq j} S_{ij}, \quad p_{\text{max}} = \max_{i \neq j} S_{ij}
$$

(12)

The preferences $p_{\text{min}}$ and $p_{\text{max}}$ lead to a small number of clusters and a big number of clusters accordingly. In order to select the optimal candidate preference, it’s computed as follows,

$$
p_i = p_{\text{min}} + \frac{p_{\text{max}} - p_{\text{min}}}{m-1} \times (i-1), \quad i = 1, \ldots, m
$$

(13)

The optimal preference is selected by analyzing the number of clusters with applying the candidate preference to the AP. Once the problem of selecting preference is solved, the AP algorithm will have the best clustering effect.

C. THE STRATEGY OF LOAD BALANCING

The position of cloudlet and the set of covered MDs are determined using the APOP algorithm, which means that the service range of cloudlet and providing service to which MD are also determined. But, the APOP merely gets a good result of clustering. If too many MDs are served by one cloudlet, the computing load of cloudlet is too heavy, which results in the response latency. So, the load balancing strategy of balancing the number of MDs served by one cloudlet is proposed to optimize the method, which is shown in Algorithm1. Algorithm1 is used to optimize the effect of cloudlet, so it is used usually after the position of cloudlet is decided, that is to say, Algorithm1 is executed after Algorithm2 and Algorithm3. Here comes the detailed introduction for this Algorithm.

When one cloudlet is under the burden of heavy load, the load balancing strategy is used to move MDs from the cloudlet with heavy burden to the nearby cloudlet with rich computing resources left. Firstly, the load of cloudlet is evaluated via the number of MDs covered by the same cloudlet. When the number of MDs covered by one cloudlet more than a certain amount, the cloudlets are working in the heavy burden state, which lead to the high-latency service. Therefore, it is quite significant to select the optimal number of MDs as the evaluation criterion of maximum load capacity of cloudlet ensuring the average as well. And the reasonable load range

Algorithm 1 The Strategy of Load Balancing

**Input:** The set of cloudlets $CL$, the set of MDs $M$, the position of MDs $p_{di}(t)$ and cloudlets $p_{clj}(t)$

**Output:** The set of MDs covered by the cloudlet at time $t$

```
D_j(t)
```

1: Compute the distance $d$ between cloudlet $cl_j$ and $cl_k$
2: for $i = 1$ to $K$ do
3: \hspace{1cm} for $i = 1$ to $n$ do
4: \hspace{2cm} if $j' = k$ then
5: \hspace{3cm} if number($D_j(t) - average > 0.05 * average$ and $average - number(D_k) < 0.05 * average$ then
6: \hspace{4cm} if abs($dis_{ij} - d/2) < d/9$ and $dis_{jk} < 1.4 * d$ then
7: \hspace{5cm} the cloudlet that covers $d_i$ changes from $cl_j$ to $cl_k$
8: \hspace{5cm} if number($D_k(t) < average$ then
9: \hspace{6cm} Continue
10: \hspace{11cm} end if
11: \hspace{10cm} end if
12: \hspace{9cm} end if
13: \hspace{8cm} end if
14: \hspace{7cm} end for
15: \hspace{6cm} end for
16: return $D_j(t)$
```

Algorithm 2 Cloudlet Movement Strategy

**Input:** The movable path of cloudlets $G_f$

**Output:** The moving trace of all cloudlets

```
1: Run the Algorithm 3
2: Get the position of cloudlets $P_{clj}(t)$
3: while $i < k$ do
4: \hspace{1cm} $W$ ← the distance between $P_{clj}(t)$ and $P_{clj}(t(t'))$
5: \hspace{2cm} end while
6: for $i = 1$ to $k$ do
7: \hspace{1cm} for $j = 1$ to $k'$ do
8: \hspace{2cm} if $P_{clj}(t')$ matches to $P_{clj}(t)$ then
9: \hspace{3cm} Using Dijkstra Algorithm to get the shortest trace
10: \hspace{4cm} Get the $x$ from $P_{clj}(t)$ and $P_{clj}(t')$
11: \hspace{5cm} end if
12: \hspace{6cm} end for
13: \hspace{7cm} end for
14: Get the moving trace of all cloudlets
```

34318

VOLUME 8, 2020
of cloudlets is computed with experiments, which is shown in the following situation. Then the distance between \( c_l \) and \( c_j \) is denoted as \( d \), which evaluates the MDs that can be moved from one cloudlet to another. More specifically, in order to ensure the quality of service, the MD that is moved from the cloudlet full of heavy burden should be served within the service range of another cloudlet. Normally, the MD is moved to the nearby cloudlet, so the distance of MD and two cloudlets is compared to evaluate whether the MD is suitable to be moved. And the potential range of MDs that can be moved is shown in situation 3). Here come the related situations that will change the MD belong to which cloudlet.

1) The number of \( D_j \) minus the average number of MDs is more than five percent of the average.
2) The number of \( D_k \) minus the average number of MDs is less than five percent of the average.
3) The distance \( d_{ij} \) between MD \( d_{ij} \) and cloudlet \( c_l \) minus \( d/2 \) is farther than \( \beta \) percent of \( d \), where \( \beta \) is the experimental test value.

When the above situation is satisfied, the MD \( d_j \) will be migrated from cloudlet \( c_l \) to the cloudlet \( c_k \). In this way, the MD can get better and faster service provided by the cloudlet with the load balancing.

D. THE STRATEGY OF MOVEMENT

The MDs in CPSS have become smaller and easier for users to carry around, thus the movements of MDs are also more frequent. In the time period from \( t \) to \( t' \), the position of cloudlet \( c_l \) at time \( t \) denoted as \( p_{c_l}(t) \) may not be suitable to serve the MDs at time \( t' \). Therefore, the position of MDs should be changed according to the MDs before and after moving. We use the Algorithm 3 to get the cloudlet position at time \( t \) and \( t' \). Then, the Algorithm 2 is used to make the cloudlet at initial position arriving to the position of time \( t' \) with the shortest path. So a method based on Dijkstra algorithm is proposed to move cloudlet ensuring the resource utilization of cloudlet.

The directed acyclic graph \( G_j = (V, \xi, W) \) is used to compute the shortest path. \( V = \{v_{11}, v_{12}, \ldots, v_{mm}\} \) is the vertex of the path graph and it represents the position of cloudlet, which is shown in Algorithm 2. \( \xi = \{\xi_1, \xi_2, \ldots\} \) is the edge of the path graph and it represents the potential movement of cloudlet. \( W \) is the position weight and it represents the distance between the original position of cloudlet and its potential position at time \( t' \). And, the cloudlet position before moving is set as the initial point. We get the shortest path for cloudlet movement by looping to compare the distance between the initial position of one cloudlet and other potential positions. Finally, the directed acyclic graph \( G_j = (V, \xi, W) \) is output, which shows the moving paths for cloudlets. However, it remains a question that the number of cloudlets before and after moving may be different. And here comes the solution. If the number of original cloudlet \( k \) is less than the number of potential cloudlet \( k' \), the original cloudlet is matched to the potential cloudlet firstly with the shortest path and then the extra cloudlets will be added. If \( k \) is greater than \( k' \), the redundant cloudlets having a larger position weight will be disabled.

E. THE PLACEMENT FOR CLOUDLET BASED ON APOP

The set of MDs is presented by \( M = \{m_1, m_2, \ldots, m_n\} \) and it’s a two-dimensional data set containing the position of MD \( d_j \). And the position of MDs satisfies the Gaussian distribution \( G \). The position of cloudlets is equal to the position of clustering centers, which is based on the APOP. With an optimal preference, the AP algorithm can generate a high-performance clustering result. That is to say, we need fewer cloudlets to place in WMAN but most MDs still can be served by cloudlet, which is shown in Algorithm 3.

V. EXPERIMENTAL EVALUATION AND DISCUSSION

In order to validate the performance of the algorithm APOP, it is compared with other methods using K-Means algorithm.
or Mini Batch K-Means algorithm, which have excellent
effect and have been widely used.

K-Means, as an unsupervised clustering algorithm, is one
of the most typical clustering algorithms and has a good
clustering effect, so it is quite suitable as comparative
method. When using the K-Means for a given sample set, the
number of clusters is determined as K in advance, so that the samples
within the cluster are distributed as closely as possible and
the distance between the clusters is as large as possible. More
specifically, K-Means is unsupervised learning. When giving
the number K of clusters, this algorithm will randomly choose
K points as the center of clusters. The point closest to the
center of one cluster is marked as belonging to that cluster.
Then, the K points are selected again through the computing
and repeat the previous step until the position of the cluster
center no longer changes. However, the selection of K has
no reference and requires a lot of experiments. Moreover,
when the number of sample set is more than one hundred
thousand, the convergence rate will slow down. So, the Mini
Batch K-Means selecting a subset of the sample set to do
with the traditional k-Means is proposed for dealing with the
numerous sample set, which makes the convergence speed
of the algorithm is greatly accelerated, but there are some
decrease in the accuracy of clustering.

Using these two excellent algorithms as comparison,
the performance of our method based APOP can be fully
tested. And the extensive simulation experiments are con-
ducted in this section. Before the experiments being con-
ducted, the experimental parameters are set firstly. Then,
the analysis based on the experimental result is followed.

A. DATASET AND EXPERIMENTAL SETTING

In the environment of WMAN, we set up a uniform standard
for cloudlet: every cloudlet is equipped with access points to
access to the network. We generate the data set that obeys
the Gaussian distribution $G = (\mu, \sigma)$ to simulate the MDs
and its parameters $\mu$ and $\sigma$ represent the mean and standard
deviation, respectively. The data sets that obey the Gaussian
distributions of three sets of different parameters $G_1$, $G_2$, $G_3$
represent the set of different number of MDs: 500, 1000,
1500 accordingly at time $t$. The Gaussian distributions are
set as follows:

- $G_1 = \{(700, 600), (1400, 500), (3200, 550), (2600, 560)\}$
- $G_2 = \{(800, 500), (1300, 520), (3000, 550), (2800, 540)\}$
- $G_3 = \{(700, 580), (1200, 520), (3100, 560), (2700, 540)\}$.

The Gaussian distributions at time $t'$ are as follow:

- $G_1' = \{(900, 560), (1200, 490), (3200, 550), (2400, 600)\}$
- $G_2' = \{(1000, 520), (1500, 540), (3300, 550), (2600, 560)\}$
- $G_3' = \{(900, 600), (1400, 540), (3300, 540), (2500, 560)\}$.

B. PERFORMANCE ANALYSIS

In this section, some parameters are used to evaluate the
clustering algorithms and the optimization of load balanc-
ing including the average coverage of cloudlet $\overline{Cov}$ and the
standard deviation of MDs $\sigma(N)$, which follow the (14),
(15) and (16):

\[
\overline{Cov} = \frac{\sum Cov_{i}(n)}{k} \quad (14)
\]

\[
Cov(i) = \frac{n_{cov}(i)}{n_i}, \quad i = 1, \ldots j, \ldots k \quad (15)
\]

where $Cov(i)$ represents the coverage of $i_{th}$ cloudlet, $n_{cov}(i)$
represents the number of MDs covered by the $i_{th}$ cloudlets
at time $t$ and $n_i$ represents the number of MDs served by
$i_{th}$ cloudlet. The standard deviation of MDs $\sigma(N)$ shows the
evenness of the number of MDs in each cluster, which reflects
the effectiveness of load balancing optimization algorithm.
And the $\sigma(N)$ is computed by using (16). The average number
of MDs covered by cloudlet is presented as $\mu$. Under the same
condition, the smaller the value of $\sigma(N)$, the more balanced
the number of MDs in each cluster.

\[
\sigma(N) = \sqrt{\frac{\sum_{i=1}^{j}(K_i - \mu)^2}{j}}
\]  

(16)

1) THE AVERAGE COVERAGE
The result of APOP is compared with the other two clustering algorithms, which can show the performance comparison more clearly. The important index to evaluate the performance of clustering is the coverage of cloudlet. That is to say, the higher \( \bar{Cov} \), the better performance of the algorithm and the superiority for dealing with the placement problem can be reflected. K-Means is a typical algorithm that has excellent clustering performance especially for the MDs less than one hundred thousand. The Mini Batch K-Means is proposed to deal with the numerous MDs that are more than one hundred thousand. So the average coverage of MDs using K-Means is higher than Mini Batch K-Means in Fig. 5, where Mini Batch K-Means is using Mbkms to represent.

Additionally, we can clearly see that no matter the number of MDs is 500, 1000 or 1500, the average coverage of MD with APOP is higher than K-Means and Mini Batch K-Means. Although when the number of MD \( n \) is 1000, the average coverage of different clustering algorithms is close, the performance of APOP algorithm is still better than others. Even after the MDs moving, the cloudlet still can provide service for more MDs compared to other two algorithms. For better efficiency of cloudlet, load balancing is used and its experiment result of coverage is shown in Fig. 6. In the histogram, the value of bar columns between original data and load balancing is quite close, which means the load balancing
not only improves the efficiency of cloudlet but ensures the quality of service for users.

During the process of the experiment, we find that the performance of AP algorithm is greatly influenced by preference. More specifically, when giving the optimal number of cloudlets, the AP algorithm can make cloudlet has higher coverage. If the unreasonable preference results in the small amount of cloudlets, the average coverage run by the AP algorithm decreases rapidly and may be lower than the other two clustering algorithms.

2) THE LOAD BALANCING
From Fig. 5, we can see that the cloudlet can provide more users with service using APOP method. In order to ensure the efficiency of cloudlet avoiding the heavy computing burden, the load balancing strategy is proposed to solve this problem. The load balancing reflecting the evenness of the amount of MDs in each cloudlet is shown in Fig. 7. In order to evaluate the effect of load balancing, the standard deviation before and after moving is tested for further comparison. After load balancing, the standard deviation decreases obviously in Fig. 7. More specifically, the lower the stand deviation, the more balanced of the amount of MDs in each cloudlet.

For estimating the partial load balancing, we take experiments for further comparison, which are shown in Fig. 8, Fig. 9 and Fig. 10. The load balancing strategy moves the MD from the cloudlet with a heavy load to other cloudlets with low resource utilization. From Fig. 8, the number of MDs served by the cloudlet is more balanced and becomes closer to the average number. Additionally, the number of cloudlets after moving add up to 6 and is more than the initial number because the cloudlet should ensure the quality of service provided to the more dispersed MDs.
Especially the condition after moving in Fig. 9, we can see that the number of cloudlet increases greatly compared to the condition before moving, this is because the distance between MDs and cloudlet is getting further and the service scope is limited, which needs more cloudlets to serve more discrete MDs. When the number of clouds increase, such as the condition after moving in Fig. 9 and Fig. 10, the number of MDs after load balancing changes slightly, but it is still more balanced than the initial result without load balancing. When the number of MDs in the cloudlet is much larger than the average, some MDs will be moved to other cloudlets. On the contrary, if the resource utilization of cloudlet is quite low, MDs are more likely to be served in this cloudlet. The reason why the number of MDs changes slightly is that the MDs covered by the cloudlet is in the best position to get less response latency, which can not be moved. Most of MDs are covered by the cloudlet and are accessed to the cloudlet in low latency, so the number of MDs which can be moved is quite small. More specifically, the load balancing strategy is implemented with coverage guaranteed, which ensures the global service.

VI. CONCLUSION

In this paper, we focus on cloudlet placement for CPSS in WMAN. Technically, a new method named APOP method based on the improved Affinity Propagation is proposed. More specifically, APOP does not need to specify the K value in advance, but it can obtain the least number of cloudlets with the largest number of MDs. In addition, the load balancing strategy is used to ensure that the load of each cloudlet maintains a balanced state.

In our future work, we will investigate the cloudlet deployment or computation offloading problems over 5G network or other scenarios, such as Vehicular Networks [49]–[51]. Meanwhile, the privacy preservation and data caching will be considered in the cloudlet deployment process [52], [53].

FIGURE 10. The load balancing comparison of 1500 MDs.

REFERENCES

[1] X. Wang, L. T. Yang, X. Xie, J. Jin, and M. J. Deen, “A cloud-edge computing framework for cyber-physical-social services,” IEEE Commun. Mag., vol. 55, no. 11, pp. 80–85, Nov. 2017.
[2] Y. Jiang, H. Song, Y. Yang, H. Liu, M. Gu, Y. Guan, J. Sun, and L. Sha, “Dependable model-driven development of CPS: From stateflow simulation to verified implementation,” ACM Trans. Cyber-Phys. Syst., vol. 3, no. 1, pp. 1–31, Aug. 2018.
[3] H. Song, G. Fink, and S. Jeschke, Security and Privacy in Cyber-Physical Systems: Hoboken, NJ, USA: Wiley, 2017.
[4] X. Xu, X. Zhang, M. Khan, W. Dou, S. Xue, and S. Yu, “A balanced virtual machine scheduling method for energy-performance trade-offs in cyber-physical cloud systems,” Future Gener. Comput. Syst., vol. 105, pp. 789–799, Apr. 2020.
[5] X. Wang, W. Wang, L. T. Yang, S. Liao, D. Yin, and M. J. Deen, “A distributed HOSVD method with its incremental computation for big data in cyber-physical-social systems,” IEEE Trans. Comput. Social Syst., vol. 5, no. 2, pp. 481–492, Jun. 2018.
[6] L. Qi, Y. Chen, Y. Yuan, S. Fu, X. Zhang, and X. Xu, “A QoS-aware virtual machine scheduling method for energy conservation in cloud-based cyber-physical systems,” World Wide Web, pp. 1–23, May 2019, doi: 10.1007/s11280-019-00684-y.
[7] W. Gong, L. Qi, and Y. Xu, “Privacy-aware multidimensional mobile service quality prediction and recommendation in distributed fog environment,” Wireless Commun. Mobile Comput., vol. 2018, pp. 1–8, Apr. 2018.
[8] Y. Zhang, K. Wang, Q. He, F. Chen, S. Deng, Z. Zheng, and Y. Yang, “Covering-based Web service quality prediction via neighborhood-aware matrix factorization,” IEEE Trans. Services Comput., to be published.
[9] X. Xu, R. Mo, F. Dai, W. Lin, S. Wan, and W. Dou, “Dynamic resource provisioning with fault tolerance for data-intensive meteorological workflows in cloud,” IEEE Trans Ind. Informat., to be published.
[10] L. Qi, Q. He, F. Chen, W. Dou, S. Wan, X. Zhang, and X. Xu, “Finding all you need: Web APIs recommendation in Web of things through keywords search,” IEEE Trans. Comput. Social Syst., vol. 6, no. 5, pp. 1063–1072, Oct. 2019.
[11] Y. Zhang, G. Cui, S. Deng, F. Chen, Y. Wang, and Q. He, “Efficient query of quality correlation for service composition,” IEEE Trans. Services Comput., to be published.
[12] X. Sun, S. Wang, Y. Xia, and W. Zheng, “Predictive-trend-aware composition of Web services with time-varying quality-of-service,” IEEE Access, vol. 8, pp. 1910–1921, 2020.
[13] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, “Location-aware deep collaborative filtering for service recommendation,” IEEE Trans. Syst., Man, Cybern., Syst., to be published.
[14] X. Wang, L. T. Yang, H. Li, M. Lin, J. Han, and B. O. Apduhan, “NQA: A nested anti-collision algorithm for RFID systems,” ACM Trans. Embedded Comput. Syst., vol. 18, no. 4, pp. 1–21, Jul. 2019.

K. Peng et al.: New Cloudlet Placement Method Based on AP for CPSS in WMAN
K. Peng (Member, IEEE) received the Ph.D. degree in computer science from the State Key Laboratory of Network and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, China, in 2014. From 2017 to 2018, he was a Visiting Faculty Member (under the Stated Scholarship Fund) with the Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, BC, Canada. He is currently an Associate Professor with the College of Engineering, Huaqiao University, China. He has coauthored over 20 journal articles/conference papers. His research interests include service computing and mobile edge computing. He is a member of CCF and a TPC Member for more than ten conferences. He is also a Reviewer for more than ten journals.
XINGDA QIAN is currently pursuing the B.S. degree in Internet of Things engineering with Huaqiao University. His research interests include mobile edge computing, cloud computing, and the optimization of cloudlet placement.

BOHAI ZHAO is currently pursuing the B.S. degree in Internet of Things engineering with Huaqiao University. His research interests include mobile edge computing and the computing offloading of workflow applications.

KEJIA ZHANG received the Ph.D. degree in cryptography from the Beijing University of Posts and Telecommunications, in 2015. From 2016 to 2017, he was a Visiting Scholar with the Center of Quantum Techniques (CQT), National University of Singapore (NUS). He is currently an Associate Professor with Heilongjiang University, China. His current research interests include cybersecurity, cryptography (including post-quantum and quantum cryptography), machine learning, and information theory.

YANG LIU (Member, IEEE) received the M.S. degree in computer application from the University of Science and Technology Liaoning, China, in 1998. She is currently an Associate Professor with the Computer Science and Software Engineering, University of Science and Technology Liaoning. Her research interests include DM and AI.