Edge Server Quantification and Placement for Offloading Social Media Services in Industrial Cognitive IoV

Xiaolong Xu, Bowen Shen, Xiaochun Yin, Mohammad R. Khosravi*, Huaming Wu, Lianyong Qi, Shaohua Wan

Abstract—The automotive industry, a key part of Industrial Internet of Things (IIoT), is now converging with cognitive computing (CC) and leading to industrial Cognitive Internet of Vehicles (CIoV). As the major data source of industrial CIoV, social media has a significant impact on the quality of service (QoS) of the automotive industry. To provide vehicular social media services with low latency and high reliability, edge computing is adopted to complement cloud computing by offloading CC tasks to the edge of the network. Generally, task offloading is implemented based on the premise that edge servers (ESs) are appropriately quantified and located. However, the quantification of ESs is often offered according to empirical knowledge, lacking analysis on real condition of Intelligent Transportation System (ITS). To address the above-mentioned problem, a collaborative method for the quantification and placement of ESs, named CQP, is developed for social media services in industrial CIoV. Technically, CQP begins with a population initializing strategy by Canopy and K-medoids clustering to estimate the approximate ES quantity. Then non-dominated sorting genetic algorithm III (NSGA-III) is adopted to achieve solutions with higher QoS. Finally, CQP is evaluated with a real-world ITS social media dataset from China.

Index Terms—Industrial Cognitive Internet of Vehicles; Edge Computing; Server Placement; Multi-objective Optimization

1 Introduction

As commuting and traveling have become indispensable parts of the daily routine, residents are calling for a better quality of service (QoS) provided by the Intelligent Transportation System (ITS). Recently, the Cognitive Internet of Vehicles (CIoV), aiming at improving the traffic conditions, is drawing much attention from the automotive industry [1]. In industrial CIoV, vehicles capture and share surrounding information through V2V (vehicle to vehicle) communications, then make corresponding adjustments in their speed or route to avoid accidents and congestion [2]. In addition, smart infrastructures (e.g., smart traffic lights) can provide vehicles with signals including weather impact warning and red-light violation warning through V2I (vehicle to infrastructure) communications. Simultaneously, pedestrians with smart phones or wearable devices like smart watches can communicate with vehicles through V2P (vehicle to pedestrian) communications to receive safety warnings. Such features provided by the industrial CIoV have huge potential in improving the traffic condition and raising the QoS of the ITS [3].

Generally, the above-mentioned features in industrial CIoV are based on big data-driven cognitive computing (CC). As the major data source of industrial CIoV, the social media data (e.g., dash-cam videos, voice commands) are key components of industrial manufacturing and are collected by various multimedia devices [4]. To mine out the value from raw social media data, applications including computer vision (CV), automatic speech recognition (ASR), service recommendation are conducted [5]. For example, CV enables vehicles to perform object detection to analyze road conditions and avoid pedestrians and other vehicles automatically by cognitively analyzing the video data [6]. Simultaneously, with ASR and semantic understanding, drivers can give commands to and receive feedback from their vehicles which record and extract key information from audio data, instead of looking for right buttons [7].

However, those CC applications based on social media usually go beyond the local computing capacity of a single vehicle for civil use. Thus, a powerful external computing and administrating system is needed by the automotive industry. Currently, the massive social media data produced

- *X. Xu and B. Shen are with the School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing 210044, China.*
  Email: njuxlxu@gmail.com, bwshen@nuist.edu.cn
- *X. Xu is also with the Facility Horticulture Laboratory of Universities in Shandong, WeiFang University of Science & Technology, ShouGuang, China, the Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science and Technology, Nanjing 210044, China, the Jiangsu Collaborative Innovation Center of Atmospheric Environment and Equipment Technology (CICAEET), Nanjing University of Information Science and Technology, Nanjing 210044, China and the Engineering Research Center of Digital Forensics, Ministry of Education, Nanjing 210044, China.*
- *X. Yin is with the Facility Horticulture Laboratory of Universities in Shandong, WeiFang University of Science & Technology, ShouGuang, China.*
  Email: xiaochunyin@wfust.edu.cn
- *M. Khosravi is with the Shiraz University of Technology, Shiraz, Iran.*
  E-mail: m.khosravi@sutech.ac.ir
- *H. Wu is with the Center for Applied Mathematics, Tianjin University, Tianjin 300072, PR China.*
  E-mail: whming@tju.edu.cn
- *L. Qi is with the School of Information Science and Engineering, Qufu Normal University, China.*
  E-mail: lianyongqi@gmail.com
- *S. Wan is with the School of Information and Safety Engineering, Zhongnan University of Economics and Law, Wuhan, China.*
  Email: shaohua.wan@ieee.org

Manuscript received xx xx, 20xx; revised xx xx, 20xx.

by traffic flow are transmitted through the roadside units (RSUs) to the cloud data center for analysis [8]. Generally, the cloud data center is located where construction cost is lower than urban areas. As a result, there is usually a long distance between terminal devices and cloud data center, and the transmission of the collected data is usually time-consuming. In addition, network congestion is likely to occur as redundant data are transmitted to the cloud data center. Thus, the response time of the social media services provided by the traditional cloud computing is usually unbearable, and can hardly meet the real-time demand as transporting system continues to develop [9].

Edge computing, an emerging computing paradigm contrary to the centralized cloud computing, has the potential to provide real-time CIoV services and mitigate the demand in bandwidth. Instead of transmitting data to the remote cloud, edge computing calls for terminal devices to transmit their data to ESs in close proximity [10]. Generally, most of the data are processed at the edge, and small amount of vital data are transmitted to the cloud after being preprocessed by ESs. Based on those vital data, further operations requiring global information and more computing resources are conducted on the cloud [11]. In such a collaboration of cloud and edge, the communication burden and the strict requirements of response time can be alleviated [12].

As the first step of implementing edge computing, the quantification and placement of ESs have a critical influence on the QoS of social media services provided by edge computing. If ESs are insufficiently deployed, they will be assigned with excessive workload. Consequently, the response time would significantly rise as the computing capacity of ESs are much smaller than the centralized cloud data center [13]. To minimize the service response time in the industrial CIoV, an ES is supposed to be co-located with each RSU. However, the quantity of deployed RSUs in ITS is usually large. In such an ideal placement, the construction cost and energy consumption of ESs are unaffordable [14]. Thus, the balance between QoS and the total cost of ESs is crucial to the planning of the industrial CIoV.

To achieve the balance, the actual traffic flow of the city needs to be studied to find out the specific quantity and locations of ESs. In this paper, a collaborative method for the quantification and placement of ESs, named CQP, is proposed. In contrast to existing studies which estimate and locate ESs based on empirical conclusions, this paper is the first to quantify and locate the ESs collaboratively. Specifically, the key contributions of this paper are as follows:

- Formalize the ES placement problem as a multi-objective optimization problem with three objectives, namely the minimized ES quantity, the minimized latency and the most balanced workload.
- Design a population initializing strategy based on Canopy and K-medoids clustering to avoid CQP falling into local optimal solutions.
- Adopt the non-dominated sorting genetic algorithm III (NSGA-III) [15] to search for a set of ES placement with low latency and balanced workload as well as a proper ES quantity.
- Conduct experiments based on a real-world ITS social media dataset collected from Nanjing to evaluate the effectiveness of CQP.

The rest of this paper is organized as follows. In section 2, the related work of this paper is summarized. In section 3, the model of ES quantification and placement are described. In section 4, details of CQP are presented. Then, comparison experiments based on real datasets are conducted in section 5. And in section 6, the achievements of this paper are concluded and future works are discussed.

2 Related Work

At the early stage of ITS, vehicles can communicate with each other through vehicular ad-hoc networks (VANET). With the development of IoT, automotive industry enabled the connection between smart vehicles and the Internet and gradually formed IoV [16]. Afterwards, CC, which can assist drivers or control vehicles, is combined with IoV and leads to the novel idea of CIoV. Generally, CC is conducted based on the massive social media data collected in the ITS. For instance, the intelligent vehicle is equipped with over one hundred sensors to ensure the vehicle safety and enhance passenger comfort [17]. Those sensors and cameras capture massive social media data from both inside and outside the vehicle, then applications including object detection, vehicle tracking and classification are conducted based on the data and CC technology [18]. As the cognitive social media data processing usually requires large computing capacity, cloud computing is adopted to provide automotive industry with Platform as a Service (PaaS) [19]. For example, Ali et al. [20] addressed the weakness of on-board computing device in terms of computing capacity and bandwidth, then proposed a dynamic priority-based resource allocation scheme in multimedia cloud computing to provide vehicular media services with high QoS.

Although cloud computing has made great contributions to cognitive social media data processing, Shi et al. [21] stressed the weakness of the conventional cloud computing paradigm, and showed the advantages and promising future of the edge computing. Technically, in the edge computing, instead of transmitting data to the remote cloud, terminal devices transmit their data to ESs nearby, and most of the data are stored and processed at the edge [22]. Such computing paradigm is lowering the latency by simplifying data transmission, as well as mitigating the demand of bandwidth by offloading tasks [23]. Generally, edge computing proves to be promising in CIoV.

So far, there are three major implementations of edge computing, namely fog computing, cloudlets and mobile edge computing (MEC). Rimal et al. [24] reduced the offloading delay, while extending the battery life of edge device through a cloudlet-aware resource scheduling. However, cloudlets are usually incapable of storage and computing capacity. Thus, cloudlets cannot achieve large-scale computation and analysis in the industrial CIoV. To process vehicular applications, Hou et al. [25] utilized vehicles as infrastructures, and proposed vehicular fog computing (VFC). However, the scale of decentralized devices is really large in fog computing, effective centralized control becomes hard to achieve. To overcome the weakness of fog computing and cloudlets as mentioned above, MEC is proposed. So far, optimal solutions to the problem of task offloading in

MEC have been proposed and adopted. Aiming at achieving efficient computation offloading, Chen et al. [26] proposed a distributed computation offloading method in the multi-user computation offloading game and derive the Nash equilibrium. Mach et al. [27] provided an overview of offloading principles and illustrate the application scenarios of edge computing, e.g., smart city and vehicle applications.

There are studies on cloudlet placement in recent years. For example, Zhao et al. [28] proposed a method to minimize average access delay through SDN techniques in cloudlets placement. However, considering the difference between cloudlets and MEC, the placement of cloudlets cannot be simply adopted in the MEC. In recent years, achievements related to the ES placement in the MEC have been made. For instance, Wang et al. [29] studied the ES placement while giving consideration to load balancing as well as access delay. From another starting point, Li et al. [30] devised an ES placement which can minimize the energy consumption of ESs while ensuring an acceptable latency. However, these researches mainly focused on the placement of a known quantity of ESs. To the best of our knowledge, the predefined server numbers are mostly based on empirical conclusions, and there is no method to collaboratively quantify and locate the ESs.

3 System Model and Problem Definition

In this section, the system model of the cloud-edge computing for the industrial CIoV is designed. To quantify the ESs and locate them, three models, i.e., the edge utilization, load balancing and latency model are proposed.

3.1 Cloud-edge Computing for Social Media Services in Industrial CIoV

In the industrial CIoV, RSUs, denoted by set $R = \{r_1, r_2, \cdots, r_N\}$ ($|R| = N$), collect social media data from vehicles. As the distribution of traffic flow in a city is relatively stable, the workload of RSUs can be indicated by the average size of collected data per offloading period, denoted by set $DS = \{ds_1, ds_2, \cdots, ds_N\}$. For the data processing capacity of RSUs is generally insufficient, ESs are arranged to some certain areas to process the massive social media data collected by RSUs. The ESs are denoted by set $E = \{e_1, e_2, \cdots, e_K\}$ ($|E| = K$). In the CIoV, an RSU can communicate with other RSUs, ESs and cloud access points (APs) in its range. To ensure that the ESs are placed where suitable for construction, they are prescribed to be co-located with some certain RSUs. Generally, the framework of cloud-edge computing for social media service in industrial CIoV is shown in Fig. 1.

To simplify the model, the coverage of each ES is assumed to be the same and denoted by R_c, so every ES and RSU can be respectively denoted by

$$e_i\,(x_i, y_i, R_c), \quad 1 \leqslant i \leqslant K, \tag{1}$$
$$r_j\,(\tilde{x}_j, \tilde{y}_j, ds_j), \quad 1 \leqslant j \leqslant N, \tag{2}$$

where (x_i, y_i) and $(\tilde{x}_j, \tilde{y}_j)$ represent the latitude and longitude of the $i-th$ ES and the $j-th$ RSU respectively, and ds_j represents the social media data size of RSU r_j.

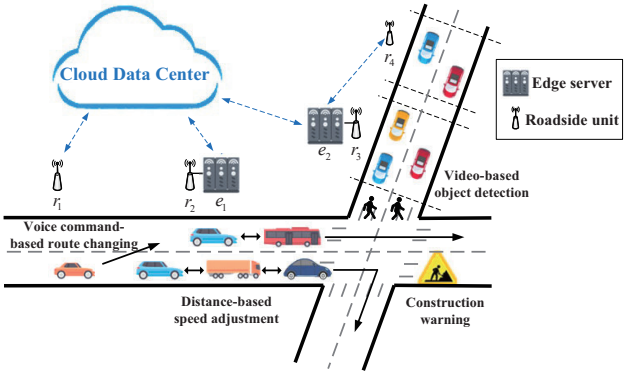

Fig. 1. A framework of cloud-edge computing for social media services in industrial CIoV.

Based on the latitude and longitude, the horizontal distance between each ES and RSU can be calculated. As the difference in height between the RSUs and ESs is usually negligible, the distance between an ES e_i and an RSU p_j is calculated by the horizontal Euclidean distance as

$$d\,(e_i, r_j) = \sqrt{(x_i - \tilde{x}_j)^2 + (y_i - \tilde{y}_j)^2}. \tag{3}$$

3.2 Edge Utilization Analysis in Industrial CIoV

By observing the location and data size of RSUs, it is easy to find that there is a high correlation between the density of RSUs and the social media service requests in certain areas. Generally, RSUs are densely placed in the areas with large amount of service requests like city center, whereas away from the city center, RSUs are sparsely placed. Thus, it is wasteful to place many ESs to cover RSUs in suburban areas, instead, more servers are supposed to be placed in core areas.

To achieve better collaboration of the cloud and edge, we decide to offload the majority of computing tasks to the ESs, while a few tasks with relative low demand for latency and bandwidth can be processed directly by the cloud. The collaboration can reduce the quantity of ESs without bringing network congestion to the cloud. Thus, we introduce the edge utilization to represent the amount of service requests processed by the ESs.

We assume that the RSU r_j is covered by the edge when it is in the coverage of at least one ES, and all the service requests of r_j are processed by the edge. Thus, the probability of ξ_j is defined as

$$P(\xi_j) = \begin{cases} 1\,, & \exists e_i \in E,\ d\,(e_i, r_j) \leqslant R_c \\ 0\,, & otherwise \end{cases}, \tag{4}$$

where ξ_j represents the event that r_j is covered by the edge. Then, we can define the edge utilization as the ratio of edge-processed service requests to total requests, calculated as

$$f_{util} = \left[\sum_{j=1}^{N} P\,(\xi_j) \cdot ds_j\right] \Big/ \left(\sum_{j=1}^{N} ds_j\right). \tag{5}$$

3.3 Load Balancing of Servers in Industrial CIoV

Provided that K ESs are deployed, we can classify the set of RSUs into $K+1$ subsets, i.e., $\{PE_1, PE_2, \cdots, PE_{K+1}\}$,

where no intersections exist in those $K+1$ subsets. RSUs in the set $PE_i = \{r_{i1}, r_{i2}, \cdots, r_{ik}\}(1 \leqslant i \leqslant K)$ is assigned to ES e_i, which means that those RSUs offload their social media services to e_i, while those in the set $PE_{K+1} = \{r_{i1}, r_{i2}, \cdots, r_{ik}\}(i = K+1)$ offload their social media services directly to the cloud. Based on the classification, the load balancing model is proposed as follows.

Since the computing capacity of ESs is limited comparing with the cloud, optimal placement aims at making better use of them. For it would never be an optimal solution when some of the ESs are overloaded while others are in idle state, we are supposed to assign similar workload to each server. Specifically, the workload of each server is denoted by

$$B_i = \begin{cases} \frac{1}{w_{edge}^{th}} \sum\limits_{r_j \in PE_i} ds_j, & 1 \leqslant i \leqslant K \\ \frac{1}{w_{cloud}^{th}} \sum\limits_{r_j \in PE_i} ds_j, & i = K+1 \end{cases}, \tag{6}$$

where w_{edge}^{th} and w_{cloud}^{th} respectively represents the maximum workload threshold of an edge or cloud server. Then, to formalize the load balancing, standard deviation is applied. As a statistic which can measure the dispersion of a set of values, lower standard deviation indicates that the values are closer to each other. Therefore, we can use the standard deviation of workload to indicate load balancing.

Specifically, let $\overline{B}$ represents the means of workload, and the standard deviation of workload is calculated as

$$\sigma_B = \left[\frac{1}{K+1} \sum_{i=1}^{K+1} \left(B_i - \overline{B}\right)^2 \right]^{\frac{1}{2}}. \tag{7}$$

Smaller σ_B indicates that the workload is more balanced, where the resources of the edge can be better utilized to avoid ESs being overloaded.

3.4 Network Latency for Offloading Social Media Services in Industrial CIoV

By adopting edge computing, we aim at complementing the shortcomings of cloud computing in real-time tasks. As a great obstacle to real-time processing, the network latency should be minimized as much as possible. Generally, network latency is caused by transmission, propagation, processing, and queuing. However, the processing latency is usually considered negligible unless complex encryption is performed by the RSUs. Thus, in this paper, we mainly focus on the transmission latency, propagation latency and queuing latency between the RSUs and servers. Then, the total latency of r_j transmitting data to e_i, denoted by T_i^j, is the sum of transmission latency, propagation latency and queuing latency, calculated as

$$\begin{aligned} T_i^j &= T_i^{trans} + T_i^{prop} + T_i^{queue} \\ &= \frac{ds_j}{\lambda_{trans}} + \frac{d\left(e_i, r_j\right)}{\lambda_{prop}} + \frac{u_i}{\mu_i - \lambda_{proc}}, \end{aligned} \tag{8}$$

where the amount of data transmitted from r_j to e_i is ds_j, the average queue length of destination server is u, package arriving rate is μ , the transmission rate is λ_{trans}, propagation rate is λ_{prop}, and package processing rate of edge and cloud is λ_{queue} and λ'_{queue} respectively. For those RSUs that transmit their data directly to the cloud, the latency can be calculated as

$$\begin{aligned} T_{cloud}^j &= T_{cloud}^{trans} + T_{cloud}^{prop} + T_{cloud}^{queue} \\ &= \frac{ds_j}{\lambda_{trans}} + \frac{d\left(cloud, r_j\right)}{\lambda_{prop}} + \frac{u_{cloud}}{\mu_{cloud} - \lambda'_{proc}}. \end{aligned} \tag{9}$$

Then, the average latency of RSUs transmitting their data to destination servers is calculated as

$$\overline{T} = \frac{1}{N} \left(\sum_{i=1}^{K} \sum_{r_j \in PE_i} T_i^j + \sum_{r_j \in PE_{K+1}} T_{cloud}^j \right). \tag{10}$$

One of our aims is to find the ES placement with minimum $\overline{T}$ to reduce the latency and improve the QoS of CC services provided by edge computing.

3.5 Definition of Collaborative ES Quantification and Placement for Social Media Services in Industrial CIoV

Latency, load balancing and edge utilization are important indicators to the performance of cloud-edge collaboration. Based on the models above, the problem of collaborative ES quantification and placement is formulated as

$$\min \sigma_B,\ \min \overline{T},\ \min K, \tag{11}$$

$$s.t. \quad f_{util} \geqslant f_{th}, \tag{12}$$

$$\forall i \in [1, K+1],\ 0 \leqslant B_i \leqslant 1, \tag{13}$$

where f_{th} represents the minimum edge utilization that can be accepted. Unlike a coverage problem, the edge utilization is not an objective to be maximized but rather a constraint as the cloud-edge collaboration enables the absence of ESs in remote areas. When (12) and (13) are satisfied, the majority of data are processed at the edge, whereas only a few data are processed at the cloud, and none of the servers is overloaded. In this way, the burden of cloud can be alleviated and the risk of network congestion can be reduced.

4 CQP for Social Media Services in Industrial CIoV

In this section, CQP is designed to achieve collaborative quantification and placement of ESs for social media services in industrial CIoV. First, a population initializing strategy based on clustering algorithms is proposed. With the initial population that obtained, NSGA-III is adopted to find out the optimal quantity and placement of the ESs.

4.1 Clustering-based Population Initialization

In the planning of industrial CIoV, since the distribution of RSUs is known before placing ESs, better usage on the information can be made. By conducting a noise-robust clustering operation, the number of clusters and the centroids of RSUs can be obtained. Thus, the number of ESs and their positions can be preliminarily estimated. In this way, NSGA-III can avoid the blindness of randomly generating the initial population, finally contribute to faster convergence and smaller probability of falling into local optimal solutions.

Centroid-based clustering algorithms usually generate clusters of similar sizes, and they tend to associate points with the nearest centroid. These two features make these

algorithms have satisfying performances in the facility location problem. Generally, there are three representative centroid-based clustering algorithms, namely K-means, K-median and K-medoids. According to their name, K-means and K-median calculate the centroids based on mean and median of coordinates respectively, while K-medoids would only choose real points as centroids. As ESs are prescribed to be co-located with RSUs, K-medoids is considered the most suitable solution to such a placement problem. If K-means or K-median is adopted to replace K-medoids, an additional process will be need to select an RSU near the centroid as the initial population which is likely to lower the overall accuracy of clustering. In addition, the robustness of the three algorithms follows K-means $\leqslant$ K-median $\leqslant$ K-medoids in case of noises and outliers [31].

Although K-medoids has advantages in the problem, there are two limitations need to be noted. One is that K-medoids requires the cluster number K given as a constant before algorithm begins. The other is that it has a total time complexity of $O((n-k)^2kt)$, where n and k are the number of data points and clusters respectively, and t represents the iteration times. That means it is time-consuming to use K-medoids when the amount of data is large.

To obtain an approximate value of K, and shorten the run time of K-medoids, Canopy clustering is used as a preprocession. Canopy is a rough clustering algorithm, which can obtain the number of clusters and their centroids with lower accuracy [32]. Since its result is influenced by noises, we need additional operations to filter out the noise clusters. As shown in Algorithm 1, those clusters with few data points ($point\_number_i < minPts$) are viewed as noises and abandoned. Then, the number of remaining clusters becomes the parameter K, and closest RSUs to centroids of those clusters become the initial medoids of K-medoids.

As the initial medoids are already close to the final solution, iteration times (denoted by t in the time complexity) needed for K-medoids to reach convergence can be reduced. To draw a conclusion, the population initializing strategy not only exerts advantage of K-medoids in accuracy and robustness, but also overcomes its drawbacks in uncertain parameter K and long executing time.

Algorithm 1 Clustering-based population initialization

Input: RSU set R, population size P, $minPts$, $K \leftarrow 0$
Output: Initial population

for $p = 1$ to P **do**
  Canopy clustering
  for $c_i \in$ Canopy centroids list C **do**
    if $point\_number_i \geqslant minPts$ **then**
      $K \leftarrow K + 1$
      $m \leftarrow \arg\min_{r \in R} d(c_i, r)$
      Add m to initial medoids list M
    end if
  end for
  K-medoids clustering with initial medoids list M
  Encode the chromosome base on clustering result
  Add the chromosome to the initial population
end for
return Initial population

Algorithm 1 elaborates how initial population is generated by clustering algorithms. An example of the clustering-based population initialization is shown in Fig. 2, in which 8 dots are marked from r_1 to r_8 to represent 8 RSUs that collect social media data in industrial CIoV. After conducting Canopy, the eight RSUs are divided into three clusters, each has a centroid represented by a red triangle. However, as cluster 3 has only one RSU, it is considered as noise thereby being removed. Thus, the number of remaining clusters $K = 2$ and the nearest RSU to each centroid are passed to K-medoids as the starting condition. Finally, two RSUs, r_1 and r_6, are selected by K-medoids, and their serial numbers are encoded for further operations in the NSGA-III.

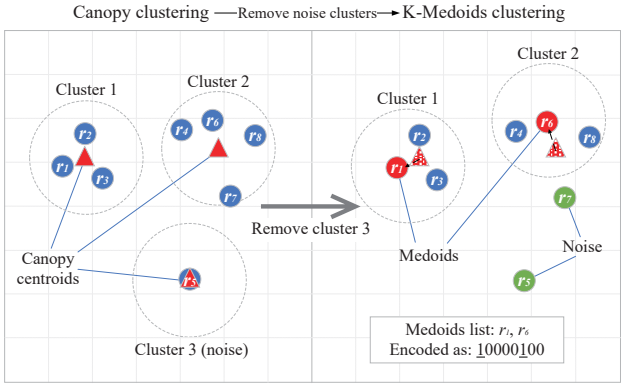

Fig. 2. An example of clustering-based population initialization with 8 RSUs (i.e., $r_1 \sim r_8$).

4.2 Collaborative Quantification and Placement of ESs Using NSGA-III

Facility location problems, including the placement of the ESs, are usually considered to be NP-hard multi-objective problems. Accordingly, multi-objective optimization algorithms are widely adopted in this series of problems [33]. So far, several multi-objective optimization algorithms have been proposed, e.g., Particle Swarm Optimization (PSO) and Genetic Algorithm (GA). PSO has its advantage in continuous problems, whereas GA has a better performance in discrete problems. Since CQP of ESs for social media services in industrial CIoV is a discrete optimization, GA is chosen to solve the problem. Among the existing genetic algorithms, NSGA-III proves to be effective in the CQP, since it has an outstanding performance in multi-objective optimizing problems with three or more objectives. The processes of NSGA-III adopted in CQP is as follows.

4.2.1 *Encoding Scheme*

Since each RSU has and only has the following two states, an ES placed next to it, or not placed next to it, so binary coding is adopted to encode the state of each RSU. Take Fig. 2 as an example. If there are eight RSUs in total, each chromosome is considered to be made up of eight bits of gene. As r_1 and r_6 are supposed to be located with an ES, the chromosome is encoded as $\underline{1}0000\underline{1}00$.

4.2.2 *Crossover and Mutation*

Crossover operation can generate new offspring by recombining two parental chromosomes. At first, two individuals

are picked from population with a probability of p_c. Then the two parental chromosomes exchange the right part of a crossover point randomly selected. This operation generates two offspring, each carrying genes from both parents.

After iterations of the GA, chromosomes would become similar to each other. To preserve genetic diversity and reduce the risk of algorithm to fall into local optimal solutions, the mutation operation is introduced. In the operation, one individual is selected with a probability of p_m, then a random bit of it is changed.

4.2.3 Fitness Function

In this operation, the chromosomes are decoded and transformed into ES placements. Since the binary coding scheme is adopted, the quantity of ESs on a chromosome is calculated as $K = \sum_{i=1}^{N} gene(i)$. Then, to minimize the latency of cognitive social media services, each RSU covered by the edge is assumed to transmit its data to the nearest ES. Based on the assumption, edge utilization, workload of each server, standard deviation of workload and average latency of each chromosome are calculated. Those results calculated in this operation will be the key to the selection operation.

4.2.4 Selection

The selection operation screens better individuals to generate the next population. During the selection operation of each generation, the parental chromosomes and offspring chromosomes (both have the size of P) are combined into one population with the size of $2P$. Then the best P individuals are selected by fast non-dominated sorting approach with reference points niching [15]. The best P individuals with the minimum quantity of ESs, minimum latency and most balanced workload, will make up the new generation. After the selection, NSGA-III goes into the next iteration.

4.3 CQP Overview

Generally, CQP is designed on the logical basis shown in Fig. 3. As an iterative algorithm which continuously updates the current solution, the final solution of GA is affected by the initial population. Inappropriate initial population may increase the iteration times to reach convergence, or increase the risk of falling into local optimal solution. However, existing genetic algorithms usually initialize population randomly. This strategy cannot guarantee the quality of the initial population, in consequence, influences the performance of genetic algorithms. From this perspective, existing genetic algorithms have its blindness. To avoid the blindness and ensure more accurate solutions, the clustering-based population initializing strategy is conducted before NSGA-III as an alternative to the random initializing operations.

In a multi-objective optimization problem, the feasible solutions are usually more than one. Specifically, the output of NSGA-III is a Pareto front with multiple non-dominated solutions. To obtain the final quantification and placement of ESs, a representative solution with the best load balancing is selected from the Pareto front to be the final output.

5 Exprimental Evaluation

In this section, we implement CQP and conduct the experiments based on the real-world ITS social media dataset collected in Nanjing. Then, we compare the results of CQP, K-medoids and random choice. The experimental evaluation shows that CQP is effective in the ES placement for the social media services in industrial CIoV.

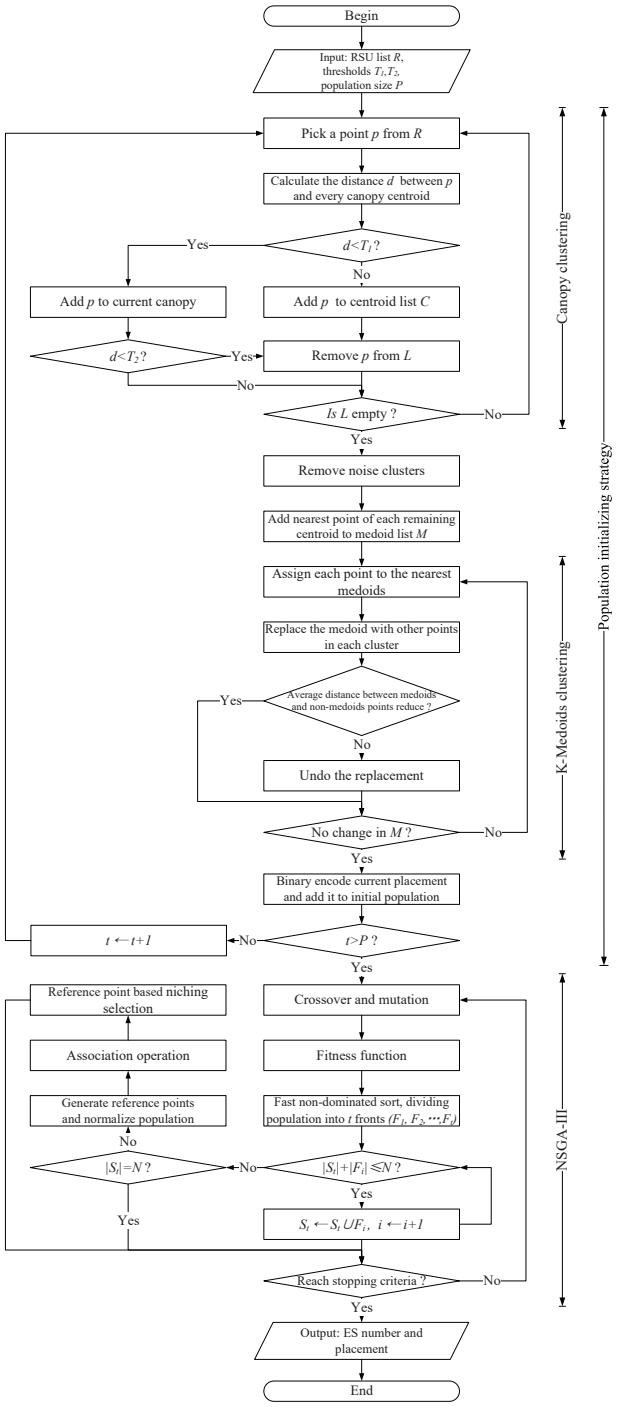

Fig. 3. The programming flowchart of CQP.

5.1 Experiment Setup

There are two real-world datasets applied in the experimental evaluation. One dataset contains details of 436 activated RSUs in Nanjing, including their latitude and longitude values. The other dataset contains vehicular social media service requests of each RSU in 30 consecutive days (00:00:00

Sept. 1st $\sim$ 23:59:59 Sept. 29th). The total number of service requests exceeds 160 million. With these data as a sample, good universality and authenticity are ensured. In Fig. 4, RSUs are marked with blue dots on the map of Nanjing. This gives us an intuitive understanding of the RSUs' locations.

Fig. 4. Distribution of 436 activated RSUs in the datasets.

5.2 Comparative Algorithms

5.2.1 K-medoids

The K-medoids algorithm can find the centroid of each cluster from real data points with high robustness to noise. When the number of clusters, denoted as K, is known a prior, K-medoids have good performance in facility location problems. From the Pareto front obtained by CQP, a representative result with the best load balancing is chosen as the final result of CQP. Then, the maximum ES number is used as the parameter K of K-medoids algorithm.

5.2.2 Proportional Selection

In the proportional selection, a probability is assigned to each RSU r_i, calculated as $p_i = ds_i / \sum_{j=1}^{N} ds_j$. Then, K RSUs are selected based on the probability sequence to co-locate ESs. This strategy exhibits priority to RSUs with large scale of social media service requests.

5.2.3 Random Selection

The random selection method can generate an unoptimized ES placement by randomly select K RSUs to co-locate ESs.

5.3 Comparison Analysis

5.3.1 Comparison of Social Media QoS

The QoS of social media services by CQP, K-medoids, proportional selection and random selection are evaluated from aspects of average latency and load balancing. The results are shown in Fig. 5 and Fig. 6, where the hollow marks represent data points of holidays (including weekends). According to the analysis on the original data sets, we found that the service requests would usually have a rise on the day before holidays, and drop significantly during holidays. This phenomenon explains the anomalous rising and dropping effects in both figures. This phenomenon will not interfere with the correctness of our experiments.

As shown in Fig. 5, the average latency follows CQP $<$ K-medoids $<$ Proportional $<$ Random. According to the statistics, CQP is 1.86% lower in latency when $K = 42$ than K-medoids in average. As K-medoids is a relatively robust and accurate clustering algorithm based on distance, it already has a good performance in lowering the latency of data transmission. Thus, the slight advantage can also verify the effectiveness of CQP. Fig. 6 indicates that the load balancing follows CQP $<$ Proportional $<$ Random $<$ K-medoids through the standard deviation of workload. According to the statistics, the standard deviation of workload by CQP is in average 21.76% lower than by K-medoids and 11.45% lower than by proportional selection. It is evident that CQP can better balance the workload of each server and prevent servers from being overloaded.

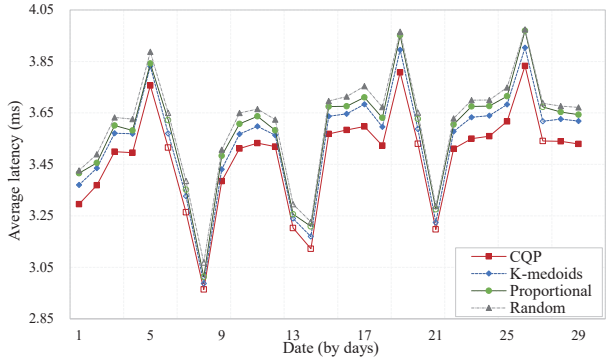

Fig. 5. Comparison of latency when $K = 42$.

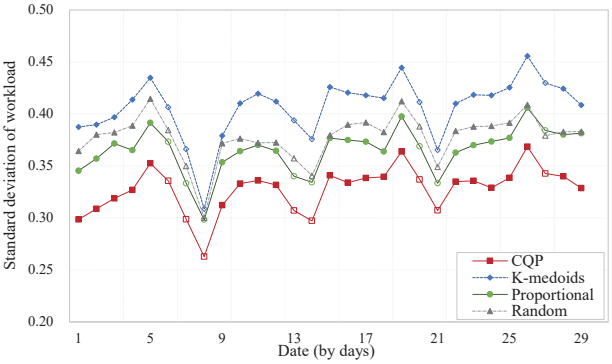

Fig. 6. Comparison of load balancing when $K = 42$.

5.3.2 Comparison of ES Location

To have a more intuitive understanding on the difference between CQP and K-medoids placing strategies, the placements obtained by CQP and K-medoids are visualized. As shown in Fig. 7 and Fig. 8, the ESs are placed at points with a red square server icon, blue points represent RSUs which have their data processed at the edge, while green points indicate that data of those RSUs are processed at the cloud.

The major difference between the result of K-medoids and CQP lies in the density of the ESs in different areas. K-medoids tends to make an even placement where some ESs are placed far from the high-demanding areas to cover RSUs in remote areas. Consequently, ESs in core urban areas are assigned with excessive workload, whereas those in remote areas are often in an idle state. This result explains the rationality of the worse-than-random load balancing of K-medoids in Fig. 6. In contrast, CQP places ESs more densely in core areas, and the edge abandons some remote

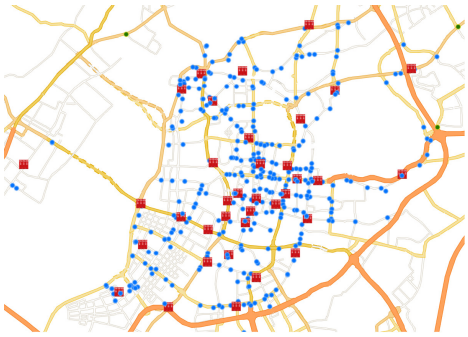

Fig. 7. Central part of the distribution map with 42 ESs by K-medoids.

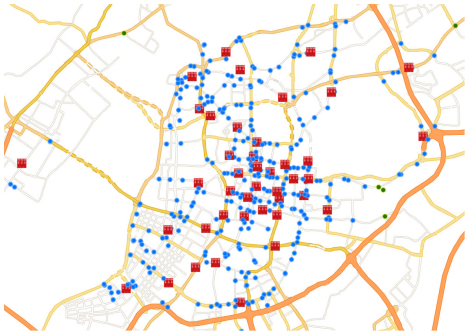

Fig. 8. Central part of the distribution map with 42 ESs by CQP.

RSUs so that it can focus on the high-demanding areas. Meanwhile, the social media services in remote areas are provided directly by the cloud. In such a placement by CQP, the synergy between edge and cloud can be achieved.

6 Conclusion and Future Work

In industrial CIoV, edge computing was adopted to provide vehicular social media services with high QoS. To implement edge computing, the ESs need to be placed appropriately. The ES placement problem was formalized as a multi-objective optimization problem with three objectives. Then, NSGA-III with a clustering-based population initialization strategy is designed and adopted in CQP, a collaborative method for the quantification and placement of ESs for industrial CIoV. The experiments are conducted based on the real-world ITS social media dataset collected in Nanjing, and the result proves that CQP is effective.

To simplify the model, the computing capacity of each ES was assumed to be equal in this paper. Actually, moderate adjustments can be adopted to obtain higher QoS. For instance, ESs in areas with heavy traffic are assumed to have stronger computing capacity. While for those in areas with light traffic, computing capacity could be reduced to lower the construction cost and energy consumption. In future works, a novel quantification and placement of ESs can be designed where the computing capacity of ESs are different.

Acknowledgment

This research is supported by the National Natural Science Foundation of China under grant no.61702277 and no.61872219. This research is also supported by the Priority Academic Program Development of Jiangsu Higher Education Institutions (PAPD) fund.

References

[1] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: A survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 1, pp. 383–398, 2018.

[2] A. H. Sodhro, Z. Luo, G. H. Sodhro, M. Muzamal, J. J. Rodrigues, and V. H. C. de Albuquerque, "Artificial intelligence based qos optimization for multimedia communication in iov systems," *Future Generation Computer Systems*, vol. 95, pp. 667–680, 2019.

[3] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "Safedrive: online driving anomaly detection from large-scale vehicle data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2087–2096, 2017.

[4] X. Xue, S. Wang, L. Zhang, Z. Feng, and Y. Guo, "Social learning evolution (sle): Computational experiment-based modeling framework of social manufacturing," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3343–3355, 2019.

[5] Y. Zhang, C. Yin, Q. Wu, Q. He, and H. Zhu, "Location-aware deep collaborative filtering for service recommendation," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, pp. 1–12, 2019.

[6] H. Gao, B. Cheng, J. Wang, K. Li, J. Zhao, and D. Li, "Object classification using cnn-based fusion of vision and lidar in autonomous vehicle environment," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 9, pp. 4224–4231, 2018.

[7] Z. Ma, H. Yu, W. Chen, and J. Guo, "Short utterance based speech language identification in intelligent vehicles with time-scale modifications and deep bottleneck features," *IEEE transactions on vehicular technology*, vol. 68, no. 1, pp. 121–128, 2018.

[8] Z. Xia, L. Lu, T. Qiu, H. Shim, X. Chen, and B. Jeon, "A privacy-preserving image retrieval based on ac-coefficients and color histograms in cloud environment," *Computers, Materials & Continua*, vol. 58, no. 1, pp. 27–44, 2019.

[9] H. B. Liaqat, A. Ali, J. Qadir, A. K. Bashir, M. Bilal, and F. Majeed, "Socially-aware congestion control in ad-hoc networks: Current status and the way forward," *Future Generation Computer Systems*, vol. 97, pp. 634–660, 2019.

[10] J. Guo, B. Song, S. Chen, F. R. Yu, X. Du, and M. Guizani, "Context-aware object detection for vehicular networks based on edge-cloud cooperation," *IEEE Internet of Things Journal*, 2019.

[11] Y. Lv, Y. Duan, W. Kang, Z. Li, and F.-Y. Wang, "Traffic flow prediction with big data: a deep learning approach," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2014.

[12] J. Ren, G. Yu, Y. He, and G. Y. Li, "Collaborative cloud and edge computing for latency minimization," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 5, pp. 5031–5044, 2019.

[13] A. K. Bashir, R. Arul, S. Basheer, G. Raja, R. Jayaraman, and N. M. F. Qureshi, "An optimal multitier resource allocation of cloud ran in 5g using machine learning," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, p. e3627, 2019.

[14] L. Yang, H. Zhang, M. Li, J. Guo, and H. Ji, "Mobile edge computing empowered energy efficient task offloading in 5g," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 7, pp. 6398–6409, 2018.

[15] K. Deb and H. Jain, "An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints," *IEEE Transactions on Evolutionary Computation*, vol. 18, no. 4, pp. 577–601, 2013.

[16] A. Ali, L. Feng, A. K. Bashir, S. H. A. El-Sappagh, S. H. Ahmed, M. Iqbal, and G. Raja, "Quality of service provisioning for heterogeneous services in cognitive radio-enabled internet of things," *IEEE Transactions on Network Science and Engineering*, 2018.

[17] J. Yao, Y. Ni, J. Zhao, H. Niu, S. Liu, Y. Zheng, and J. Wang, "Data based violated behavior analysis of taxi driver in metropolis in china," *CMC Computer, Materials and Continua*, vol. 60, no. 3, 2019.

[18] A. Mehrish, A. V. Subramanyam, and M. Kankanhalli, "Multimedia signatures for vehicle forensics," in *2017 IEEE International Conference on Multimedia and Expo (ICME)*. IEEE, 2017, pp. 685–690.

[19] K. Djemame, R. Bosch, R. Kavanagh, P. Alvarez, J. Ejarque, J. Guitart, and L. Blasi, "Paas-iaas inter-layer adaptation in an energy-aware cloud environment," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 2, pp. 127–139, 2017.

[20] A. Ali, H. Liu, A. K. Bashir, S. El-Sappagh, F. Ali, A. Baig, D. Park, and K. S. Kwak, "Priority-based cloud computing architecture for multimedia-enabled heterogeneous vehicular users," *Journal of Advanced Transportation*, vol. 2018, 2018.

[21] W. Shi, J. Cao, Q. Zhang, Y. Li, and L. Xu, "Edge computing: Vision and challenges," *IEEE Internet of Things Journal*, vol. 3, no. 5, pp. 637–646, 2016.

[22] D.-Y. Kim and S. Kim, "A data download method from rsus using fog computing in connected vehicles," *CMC-COMPUTERS MATERIALS & CONTINUA*, vol. 59, no. 2, pp. 375–387, 2019.

[23] C. Long, Y. Cao, T. Jiang, and Q. Zhang, "Edge computing framework for cooperative video processing in multimedia iot systems," *IEEE Transactions on Multimedia*, vol. 20, no. 5, pp. 1126–1139, 2017.

[24] B. P. Rimal, D. P. Van, and M. Maier, "Cloudlet enhanced fiber-wireless access networks for mobile-edge computing," *IEEE Transactions on Wireless Communications*, vol. 16, no. 6, pp. 3601–3618, 2017.

[25] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[26] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2015.

[27] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[28] L. Zhao, W. Sun, Y. Shi, and J. Liu, "Optimal placement of cloudlets for access delay minimization in sdn-based internet of things networks," *IEEE Internet of Things Journal*, vol. 5, no. 2, pp. 1334–1344, 2018.

[29] S. Wang, Y. Zhao, J. Xu, J. Yuan, and C.-H. Hsu, "Edge server placement in mobile edge computing," *Journal of Parallel and Distributed Computing*, vol. 127, pp. 160–168, 2019.

[30] Y. Li and S. Wang, "An energy-aware edge server placement algorithm in mobile edge computing," in *2018 IEEE International Conference on Edge Computing (EDGE)*. IEEE, 2018, pp. 66–73.

[31] P. Arora, S. Varshney *et al.*, "Analysis of k-means and k-medoids algorithm for big data," *Procedia Computer Science*, vol. 78, pp. 507–512, 2016.

[32] Q. Xu and G. Tao, "Traffic accident hotspots identification based on clustering ensemble model," in *2018 5th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud)/2018 4th IEEE International Conference on Edge Computing and Scalable Cloud (EdgeCom)*. IEEE, 2018, pp. 1–4.

[33] L. Zhen, W. Wang, and D. Zhuge, "Optimizing locations and scales of distribution centers under uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 11, pp. 2908–2919, 2016.

Xiaolong Xu received the Ph.D. degree in computer science and technology from Nanjing University, China, in 2016. He was a Research Scholar with Michigan State University, USA, from April 2017 to May 2018. He is currently an Associate Professor with the School of Computer and Software, Nanjing University of Information Science and Technology. He has published more than 80 peer-review articles in international journals and conferences. His research interests include edge computing, the Internet of Things (IoT), cloud computing, and big data.

Bowen Shen is currently working towards his B.S. degree in Computer Science and Technology at School of Computer and Software in Nanjing University of Information Science and Technology. His research interests include edge computing and IoT.

Xiaochun Yin received the B.S. degree in education and technology from Qufu Normal University, Qufu, China in 2004, and received the M.S. degree in education and technology from Nanjing Normal University, Nanjing, China in 2007, and received the Ph.D. from Dongseo University, Korea in 2015. She is now working as an associate professor in Weifang University of Science & Technology China. Her research interests include network security, IoT security, authentication protocol and agricultural intelligence systems. She has published over 20 research papers in international journals and international conferences.

Mohammad Khosravi is now with the Department of Computer Engineering, Persian Gulf University, Bushehr, Iran, and has been with Department of Electrical and Electronic Engineering, Shiraz University of Technology, Shiraz, Iran. His main interests include statistical signal and image processing, medical bioinformatics, radar imaging and satellite remote sensing, computer communications, industrial wireless sensor networks, underwater acoustic communications, information science and scientometrics. Mohammad has studied electrical engineering with expertise in communications and signal processing.

Huaming Wu is currently an associate professor in the Center for Applied Mathematics, Tianjin University. He received the B.E. and M.S. degrees from Harbin Institute of Technology, China in 2009 and 2011, respectively, both in electrical engineering. He received the Ph.D. degree with the highest honor in computer science at Freie Universität Berlin, Germany in 2015. He is currently an associate professor in the Center for Applied Mathematics, Tianjin University. His research interests include model-based evaluation, wireless networks, edge/cloud computing, deep learning and complex networks.

Lianyong Qi received his PhD degree in Department of Computer Science and Technology from Nanjing University, China, in 2011. Now, he is an associate professor of the School of Information Science and Engineering, Chinese Academy of Education Big Data, Qufu Normal University, China. He has already published more than 100 papers. His research interests include services computing, big data and IoT.

Shaohua Wan received the joint Ph.D. degree from the School of Computer, Wuhan University and the Department of Electrical Engineering and Computer Science, Northwestern University, USA in 2010. He is currently an Associate Professor at the School of Information and Safety Engineering, Zhongnan University of Economics and Law. His research interests include deep learning for IoT and Cyber-Physical Systems. He is a senior member of IEEE and has over 60 peer-reviewed research papers.