IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received August 31, 2016, accepted September 11, 2016, date of publication September 21, 2016,
date of current version October 31, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2611863

Social-Aware Data Collection Scheme Through Opportunistic Communication in Vehicular Mobile Networks

ZHIPENG TANG[1], ANFENG LIU[1], AND CHANGQIN HUANG[2]
[1]School of Information Science and Engineering, Central South University, Changsha 410083, China
[2]School of Information Technology in Education, South China Normal University, Guangzhou 510631, China

Corresponding author: A. F. Liu (afengliu@mail.csu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 61379110, Grant 61073104, Grant 61370229, and Grant 61370178, in part by the National Basic Research Program of China (973 Program) under Grant 2014CB046305, in part by the National Key Technology R&D Program of China under Grant 2014BAH28F02, in part by the S &T Projects of Guangdong Province under Grant 2014B010103004, Grant 2014B010117007, Grant 2015A030401087, Grant 2015B010110002, and Grant 2016B010109008, and in part by GDUPS (2015).

ABSTRACT To enable the intelligent management of Smart City and improve overall social welfare, it is desirable for the status of infrastructures detected and reported by intelligent devices embedded in them to be forwarded to the data centers. Using ''SCmules'' such as taxis, to opportunistically communicate with intelligent devices and collect data from the sparse networks formed by them in the process of moving is an economical and effective way to achieve this goal. In this paper, the social welfare data collection paradigm SWDCP-SCmules data collection framework is proposed to collect data generated by intelligent devices and forward them to data centers, in which ''SCmules'' are data transmitters picking up data from nearby intelligent devices and then store-carry-forwarding them to nearby data centers via short-range wireless connections in the process of moving. Because of the storage limitations, ''SCmules'' need to weigh the value of data and select some less valuable data to discard when necessary. To quantify the value of data and find a well-performed selection strategy, the concept of priority is introduced to the SWDCP-SCmules scheme, and then, the simulated annealing for priority assignment SA-PA algorithm is proposed to guide the priority assignment. The SA-PA algorithm is a universal algorithm that can improve the performance of SWDCP-SCmules scheme by finding better priority assignment with respect to various optimization targets, such as maximizing collection rate or minimizing redundancy rate, in which priority assignment problem is converted into an optimization problem and simulated annealing is used to optimize the priority assignment. From the perspective of machine learning, the process of optimization is equal to automatically learn social-aware patterns from past GPS trajectory data. Experiments based on real GPS trajectory data of taxis in Beijing are conducted to show the effectiveness and efficiency of SWDCP-SCmules scheme and SA-PA algorithm.

INDEX TERMS Vehicular mobile networks, social welfare, data collection, opportunistic communication, oblivious data mules, simulated annealing, machine learning.

I. INTRODUCTION

Internet of Things (IoT) is experiencing extremely rapid development. There are several phenomena which can prove this trends: Firstly, the amount of data generated by IoT has been growing exponentially since past few years [1]–[9]. A Cisco's report reveals that the overall data traffic of IoT in 2014 has grown to 69 percent and is approximately 30 times the size of the entire global Internet in 2000 [3], [10]. Secondly, the number of connected devices have already exceeded the total number of people living on Earth since 2011. Connected devices have reached 9 billion and are expected to reach 24 billion by 2020 [11], [12]. Thirdly, in 2012, application systems based on IoT has contributed $4.8 trillion to revenue of international corporations and the number is still soaring since then, according to another report from Cisco [10], [13].

Due to the rapid development of IoT, the intelligent management of city becomes possible, which enables us to achieve the dream of Smart City.

In the process of constructing Smart City, sensors and actuators will definitely find their place because of their low costs and energy consumption [14]–[17]. Intelligent devices, which are equipped with various sensors and actuators, can be embedded into infrastructures of Smart City to enhance their productivity and functionality, such as detecting and reporting the status of infrastructures [14]. For example: garbage cans with intelligent devices can monitor their trash level, if they are filled up with trashes, the intelligent devices will detect it and send request for cleaning; Street lights with intelligent devices can detect their status, such as the power supply and the condition of bulbs, and send request for maintaining. More applications can be found in the fields of electricity facility, communication system, contamination monitoring, sanitary engineering, etc [18].

Although the form of applications is distinct, they all share a common abstract paradigm: detecting status of infrastructures and forwarding it to related municipal departments, which will then take measures to maintain the infrastructures. This paradigm realizes the intelligent management of city infrastructures, which will make a more comfortable city environment and improve social welfare significantly [17], [18]. In this paper, this paradigm is called Social Welfare Data Collection Paradigm (SWDCP Paradigm).

To implement SWDCP paradigm, how to establish network connections between the large amount of intelligent devices distributed in the whole city and municipal departments is a core problem [17].

In fact, the way to establish such kind of network connections really depends on the requirements of communication:

For real time communications, there are two feasible approaches: the first one is to equip each intelligent device with a SIM card [17], the second one is to build permanent base stations to relay data/messages generated by intelligent devices located in their coverage area [19]. However, they both have their own shortcomings. For the first approach, the high costs and energy consumption sharply reduce the advantage of this approach [20], [21]. Besides that, this approach potentially increases the number of devices occupying radio frequency (RF): concretely speaking, RF is a kind of valuable limited communication resource. Devices occupying RF have already reached 9 billion and are expected to reach 24 billion by 2020 [22]. The first approach will significantly increase such devices and thereby make the problem of the scarcity of RF much more serious [1], [3], [8]. For the second approach, as new infrastructures are built and old infrastructures are removed in the process of city construction, the distribution of intelligent devices in the city also constantly change, which makes building permanent base stations covering all intelligent devices impossible [17].

However, for most applications in SWDCP paradigm, real time communication is not necessary, That is to say, the data/messages generated by devices are not necessary to be report to municipal departments immediately, some delay is acceptable. For example, when the bulb of a street light doesn't work, the intelligent device embedded in this street light doesn't need to report this status to maintenance department immediately since one street light's malfunction doesn't influence the overall illumination of the city, several hours' data transmission delay is acceptable. Similarly, when a garbage can is filled with trashes, the intelligent device embedded in this garbage can doesn't need to report this status to sanitary department immediately since there are other available garbage cans nearby, several days' data transmission delay is acceptable [17]. Based on this feature, Bonola *et al.* think this type of network connections shares a lot of common features with Delay Tolerant Network (DTN) [17], [23].

Bonola *et al.* further proposed a data collection scheme for this kind of network connection based on Oblivious Data Mules [17]. There are three entities in their scheme: oblivious data mules, intelligent devices and data centers. Oblivious data mules (hereinafter referred to as mules), which are inspired by the work of Shah *et al.* [19], are mobile IoT nodes moving in Smart City and can opportunistically communicate with intelligent devices via short-range wireless communication to pick up data/messages and buffer them in their own storage. When mules move to the neighborhood of data centers, the buffered data/messages will be forwarded to data centers via short-range wireless communication. Intelligent devices are incorporated into the infrastructures of Smart City, they are capable of detecting the status of infrastructures and sending the data/messages to mules via short-range wireless communication. Data centers are special computing and processing nodes distributed in fixed locations of Smart City, they can connect with cloud tier via high-speed network to forward data/messages, their major responsibility is receiving data/messages buffered by nearby mules and forwarding them to cloud tier, which is accessible to corresponding municipal departments. Once municipal departments receive these data/messages, they will take measures to maintain related infrastructures. Most data centers are deployed in downtown area of Smart City to boost collection efficiency.

Here is a concrete instance: many objects can be regarded as mules in Smart City, such as taxis, buses and pedestrians holding portable communication devices [24], [25]. Taxis are the most ideal among them since they have several great features: they move independently according to customers' demands and can cover an extensive area of Smart City with somewhat random paths (with respect to buses that move only on main streets) and work 24/7. Taxis equipped with transceivers, which are capable of communicating via short-range wireless networks and exchanging data with neighboring intelligent devices and data centers [19], can be regarded as mobile IoT nodes, i.e. mules. The transceivers are incorporated into the chips of taxis and can automatically incidentally pick up data/messages sent by nearby intelligent devices when taxis are moving in Smart City to send passengers to their destinations. The data/messages incidentally collected in the process of motion are buffered and then dumped to data centers when taxis are passing near them. Data centers then simply filter out duplicated data/messages and forward the remaining information to cloud tier via the high-speed network connection. The relative municipal departments can access these data/messages from cloud tier and then take measures to maintain corresponding infrastructures. Note that the major task of taxis is serving passengers, the process of data/messages collection is incidental, which is automatically manipulated by transceivers, rather than drivers, that's why we call data mules oblivious.

The scheme proposed by Bonola *et al.* is cost-saving and energy-efficient [17]. However, this model can still be refined because it assumes the storage of mules is infinite, that is to say, mules can store all data/messages picked up along the course of moving and don't need to worry about the situation that their storage is full. Conversely, in reality, infinite storage is of impossibility and mules can only store limited data/messages. We call this refined model of mules as Storage-constrained oblivious data mules (hereinafter referred to as SCmules). Due to the storage limitation, SCmules have to discard some less important data/messages stored in buffer to make some free space in the situation that their storage is full and new more valuable data/messages are picked up. This situation occurs frequently because of the large number of intelligent devices installed in city. The selection strategy of how to choose more valuable data/messages to store and less valuable data/messages to discard plays a vital role in improving data/messages collection efficiency and overall social welfare of Smart City [26]. To prove the vitality of the selection strategy, we continue to use the aforementioned example to illustrate it: If the selection strategy is simply discarding new picked-up data/messages when the storage is full, a bad phenomenon will occur: data centers will receive a lot of duplications of data/messages from the downtown areas of a city, i.e. hotspot areas, but few data/messages from other areas, especially the remote areas, are collected. This is because the hotspot areas are always the transportation hubs of the city. The frequency that taxis pass by there is much higher than the frequency of other areas. Therefore, the probability that data/messages from hotspot areas are picked up by taxis is much higher than the probability of other areas, i.e. data/messages from hotspot areas have higher probability of occupying the storage and then being forwarded to data centers. Conversely, when taxis pass through remote area, it is likely to have no more storage to pick up new data/messages from there. So this example illustrates a bad selection strategy will lead to low data/messages collection efficiency and poor social welfare.

Now that the selection strategy is of vitality, how can we find an effective and efficient selection strategy? To answer this question, two tasks have to be finished: the first one is to introduce a universal way to formally describe selection strategy, the second one is to find a well-performed selection strategy.

A natural way to formally describe selection strategy is to introduce the concept of priority: We assign each intelligent device with a unique priority. Every registered SCmule owns a priority table, which records the priorities of all intelligent devices. The priority table is used to guide how to weigh the importance of data/messages. In detail, when the storage of a SCmule is not full, this SCmule will greedily buffer data/messages as much as possible; but when the storage is full, this SCmule will use the priority table to weigh the priority of new picked-up data/messages with the priorities of those stored in buffer. If the new picked-up one's priority is less than the priorities of all data/messages buffered, it will be simply discarded; otherwise, it will replace the buffered data/message with the least priority. The main principle of this process is to maximize the sum of priorities of data/messages buffered, which is referred as Greedy Selection Principle.

Obviously, the priority model is a natural and effective way to formally describe the selection strategy. In the example of illustrating the vitality of selection strategy, the poor selection strategy we used can be regarded as assigning each intelligent devices with equal priorities.

By combing SCmules, the model of priority and greedy selection principle, we propose a new concrete implementation of SWDCP paradigm for DTN, which is called Social Welfare Data Collection Paradigm based on Storage-Constrained Oblivious Data Mules (SWDCP-SCmules scheme).

To implement SWDCP-SCmules, we still have to address the second task which is how to assign priority to make an effective and efficient selection strategy. We refer to this problem as Priority Assignment Problem.

However, the priority assignment problem is really a challenge since there are three potential difficulties:

(1) The meaning of effectiveness and efficiency is ambiguous, different scenarios have different interpretations and correspond to different priority assignment methods. So it is troublesome to design new algorithms for every concrete interpretation of effectiveness and efficiency. For example, as will be described in section 3.2, in most cases, the definition of effectiveness and efficiency corresponds to maximizing collection rate and minimizing redundancy rate, but there are always some new demands, such as reducing collection delay of some important infrastructures, emerging with the advent of new applications. The algorithm designed for original demands fails to optimize these new demands and require a complete redesign. It is too troublesome to design new algorithm for every new demand.

(2) Even if we have an accurate definition of effectiveness and efficiency, i.e. optimization target, it is still too hard to design an appropriate algorithm that can optimize the performance of priority assignment according to this definition, not to mention a universal algorithm that can solve all unpredictable optimization targets.

For example, if the optimization target is to maximize collection rate and minimize redundancy rate, finding an effective pattern that can support the design of algorithm needs vast amounts of manual observation and thinking. In fact, it is impossible for human intelligence to perceive all the potential patterns lying in essence. Even if we have found an important pattern, e.g. decreasing the priorities of intelligent devices near data centers and increasing the priorities of those far from data centers can increase collection rate and decrease redundancy rate, you will still probably ignore the fact that this pattern doesn't work when the intelligent devices locate in the end point of one-way streets (In section 5.1, we will give a detailed possible explanation of how this pattern works and why it fails in that case).

(3) The most fatal problem is that patterns are not static. As time goes on, new patterns emerge, old patterns disappear and some patterns change periodically. For example, in hot summer, district near natatoriums will become hotspot areas and share the common features of downtown areas, but in other seasons, these districts will be ordinary areas.

According to the analysis above, we can infer that it is difficult to solve priority assignment problem using traditional approach, which is finding effective patterns and then using these patterns to design algorithms. However, we can consider to use the idea from machine learning, that is, to use learning algorithm to automatically learn social-aware patterns or other information from data and then utilize them to address this challenging problem.

The basis of why learning algorithm works is that there are a lot of social-aware patterns, which are patterns that can reflect the social preferences of citizens in Smart City, lying in the GPS trajectory data of SCmules. Concretely speaking, the GPS trajectory data of SCmules can reflect the overall social preferences of SCmules (i.e. the social preferences of citizens in Smart City). To prove this statement, we briefly illustrate an instance in this section (In section 5.1, we will explore the meaning of social-awareness more deeply): the GPS trajectory data of taxis, one of the typical examples of SCmules, can implicitly show the social preferences of citizens in the Smart City, such as the hotspot areas of Smart City in certain time interval, the preferred destinations of passengers in certain time interval, the preferred routes that drivers take in certain time interval and many other social labels. Therefore, we can use learning algorithm to automatically mine for these potential social-aware patterns lying in the data without any manual teaching and use them to make a better priority assignment to improve the performance of SWDCP-SCmules scheme.

In this paper, we try to convert the priority assignment problem to an optimization problem based on the idea from machine learning and then use simulated annealing metaheuristic to design a universal algorithm that can automatically find a priority assignment of good quality with respect to various optimization targets based on the social-aware patterns lying in the GPS trajectory data of SCmules in the past time. Concretely speaking, first, we need to specify the meaning of effectiveness and efficiency, that is to say, we need to use a performance measure to quantify the meaning of effectiveness and efficiency. The performance measure we used is the optimization target function $\mathcal{J}$ of priority assignment $\vec{\mathcal{P}}$; then, due to the future shares similar social-aware patterns with the past, we can use the past GPS trajectory data of SCmules to train a priority assignment $\vec{\mathcal{P}}$ that can optimize the optimization target function $\mathcal{J}$ and predict that $\vec{\mathcal{P}}$ will also achieve good performance in the future. Because of the complexity of the computation of $\mathcal{J}$ with respect to $\vec{\mathcal{P}}$ based on the past GPS trajectory data, we apply simulated annealing metaheuristic to search for appropriate $\vec{\mathcal{P}}$ in search space that can optimize $\mathcal{J}$. We refer to this method of solving priority assignment problem as Simulated Annealing for Priority Assignment Algorithm (SA-PA algorithm).

Based on real GPS trajectory dataset of taxis during the period of Feb. 2 to Feb. 8, 2008 in Beijing, the effectiveness of SA-PA algorithm can be proved in section 6.

The main contributions of this paper can be described as three points:

(1) We propose a Social Welfare Data Collection Paradigm based on Storage-Constrained Oblivious Data Mules (SWDCP-SCmules scheme) in which storage-constrained oblivious data mules can incidentally pick up data/messages generated by intelligent devices embedded in infrastructures of Smart City in the process of motion and store-carry-forward them to data centers, which will then notify related department to maintain corresponding infrastructures. Unlike the previous research, in SWDCP-SCmules scheme, SCmules are storage-constrained, which is much more real than the previous model. SWDCP-SCmules scheme enables the intelligent management of city to be possible and leads to the improvement of overall social welfare.

(2) Because of the storage limitation, SCmules have to weigh the importance of data/messages and discard some less important data/messages stored in the situation that their storage is full and new valuable data/messages are picked up. Therefore, the selection strategy is extremely important in SWDCP-SCmules. In this paper, the concept of priority is introduced to model the selection strategy to the priority assignment. Every intelligent device is assigned with a unique priority, which forms the priority table. Based on priority table, SCmules use greedy selection principle, which is maximizing the sum of priority of data/messages buffered, to guide the storing of more important data/messages and the discarding of less important data/messages.

(3) To find an optimized priority assignment, Simulated Annealing for Priority Assignment

Algorithm (SA-PA algorithm) is proposed to learn social-aware patterns from GPS trajectory data of SCmules. The SA-PA algorithm converts the priority assignment problem to an optimization problem based on the idea from machine learning and makes a simulated annealing-based universal algorithm that can automatically find priority assignments of good quality with respect to various optimization targets based on the GPS trajectory data of SCmules in the past time.

Below is the organization of the rest of this paper: In section 2, related works are reviewed. In section 3, the system model and problem statement of SWDCP-SCmules scheme and SA-PA algorithm are described. In section 4, SWDCP-SCmules scheme is proposed in detail. In section 5, SA-PA algorithm is described in detail. In section 6, experimental results are given to show the performance of SWDCP-SCmules scheme and SA-PA algorithm. In section 7, the conclusion of this paper is described.

II. RELATED WORK

A. DATA COLLECTION FRAMEWORK OF IoT

We first observe the structure of SWDCP-SCmules scheme from the perspective of data collection tasks. As illustrated in Figure 1, IoT data collection framework can be divided into 2 tiers: the data collection tier (DCT) and the cloud tier (CT). The main function of DCT, which is the base tier of IoT data collection framework, is sensing and collecting data [27]. The main function of CT, which is the core tier of IoT, is refining data collected by DCT and convert them to the elements of service [10].

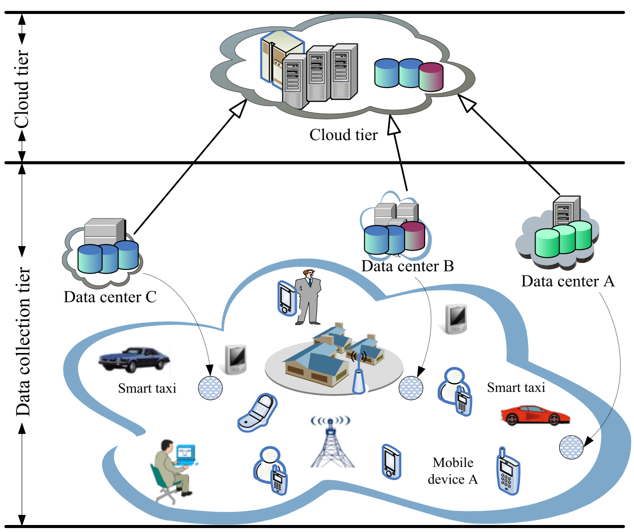


FIGURE 1. The data collection framework of IoT.

For example, the network constructed in SWDCP-SCmules scheme can be seen as a kind of vehicular mobile networks, a concrete form of the IoT data collection framework, in a narrow sense. In DCT, sensors are incorporated into infrastructures of the city [28] to sense the status of them. Vehicles opportunistically communicate with nearby sensors to pick up data/messages generated by the nearby sensors and then store-carry-forward to nearby data centers when passing the nearby data centers [17]. Data centers then send the received data/messages from DCT to CT via high-speed network connections to notify related departments to maintain corresponding infrastructures.

In this paper, we focus on the DCT part of the IoT data collection framework. According to the difference of applications, DCT can be categorized as different networks, such as wireless sensor networks [5], [6], [14], [15], [26], [28], participatory sensing networks [18], [26], [29], crowd sensing networks [29], vehicular mobile networks [17] and delay tolerant networks [23]. The researches on wireless sensor networks (WSNs) start comparatively earlier [30]. In WSNs, a large amount of wireless sensor nodes are deployed in geographic position to construct a fine-grained network connections. WSNs can be used to monitor certain objects or indexes in geographic areas, such as local temperatures [31]. Unlike WSNs, the main participant objects in participatory sensing networks and crowd sensing networks are people holding mobile smart phones which are equipped with sensing devices. As people move around city, those mobile smart phones automatically sense and collect data using sensing devices and communicate via cell networks [18], [26], [29]. The two main features of participatory sensing networks and crowd sensing networks are that the number of sensing devices is large and these sensing devices can move around to sense and collect data. Vehicular mobile networks are another form of networks in DCT. The basis of its network connection is the motion of vehicles and their opportunistic communication [17]. In delay tolerant networks (DTNs), all sensors are moving around and opportunistically communicate with each other to exchange data [23]. The major feature of DTNS is that transmission delay is not an important measure of network performance.

In this paper, the networks constructed by the SWDCP-SCmules scheme can be regarded as a synthesis of these five networks mentioned above [17]. In the synthesized networks, the geographic locations of intelligent devices, which are responsible for sensing and collecting data, are fixed, which is similar to sensors in WSNs. But SCmules, which are responsible for collecting and transferring data, are moving around the city to pick up data and store-carry-forward to data centers, which is same as vehicular mobile networks. The data collection process of the synthesized networks relies on the extensive participation of SCmules, which is similar to participatory sensing networks and crowd sensing networks. The transmission delay of data collection is not that important in the synthesized networks, which is similar to DTNs.

In the rest of this subsection, we will introduce these five networks in detail, i.e. wireless sensor networks, participatory sensing networks and crowd sensing networks, vehicular mobile networks, delay tolerant networks and their relations to the SWDCP-SCmules scheme respectively.

1) WIRELESS SENSOR NETWORKS

In early days, WSNs are put forward to monitor certain objects or indexes in local areas by deploying a large amount of wireless sensor nodes to form fine-grained networks [2], [5], [6], [9], [16], [21], [31]. This is somewhat similar to SWDCP-SCmules scheme's intelligent devices embedded in infrastructures. But a major difference between them is the deployment of intelligent devices is planless, sporadic and geographically sparse, which means they will not form self-organized networks like WSNs. Therefore, the data collection process relies on the motion and store-carry-forward mechanism of SCmules, rather than the fine-grained networks formed by wireless sensors.

There are two major elements in WSNs: sensors and sinks. Sensors are ordinary wireless sensor nodes which is responsible for sensing data. Besides sensors, there are some special wireless sensor nodes called sinks deployed in WSNs, which can be regarded as the destination of data collection.

Routing algorithm is the algorithm that routes data from sensors to sinks [2], [3], [31]. In WSNs, the key problem of data collection is to find an efficient routing algorithm, that is to say, to find a routing path from sensors to sinks with high quality of service (QoS). Some typical measures of QoS are energy consumption, network lifetime, transmission delay and packet loss rate. Usually, there exist some tradeoffs between different measures of QoS, for example, reducing transmission delay and packet loss rate will lead to higher energy consumption and shorter network lifetime [2], [3], [9], [26], [31]. Unlike WSNs, routing problem is not a problem in SWDCP-SCmules scheme, because the intelligent devices cannot form fine-grained networks and data transmission relies on the opportunistic communications among SCmules, intelligent devices and data centers.

Another relevant networks is wireless sensor actor networks (WSANs). WSANs consist of a large amount of inexpensive sensor nodes and a small amount of expensive powerful mobile actor nodes (e.g. robots) [32]. After receiving messages (e.g. fire alarm) sent by a certain sensor node, actor nodes (e.g. robots) will move to the position of this sensor node and deal with the emergency reported in the messages (e.g. putting out the fire). In WSANs, actor nodes move constantly, therefore how to effectively and reliably route the messages generated by sensor nodes to moving actor nodes becomes the central problem in WSANs.

In some forms of WSNs, mule nodes are adopted to relay data/messages [24], [25]. In most cases, there are some hotspot areas, in which energy consumption is much higher than other areas, in WSNs. Mule nodes relay data between sinks and hotspot areas to reduce energy consumption in hotspot areas, thereby increasing network lifetime.

2) PARTICIPATORY SENSING NETWORKS & CROWD SENSING NETWORKS

With the rapid development of portable devices, participatory sensing networks and crowd sensing networks have great potential in collecting and sharing sensing information through smartphones and other portable devices [18], [26], [29].

The main feature of participatory sensing networks and crowd sensing networks is that a large amount of sensing devices participate in monitoring and sensing the same type of data to accomplish the same task. For example, when the task is monitoring PM 2.5 of a city, it is too costly to establish a lot of PM 2.5 monitor stations distributed in the whole city. In addition, due to the limited number of PM 2.5 monitors, the accuracy of results is also limited. But when using participatory sensing networks and crowd sensing networks, citizens' mobile phones can sense and report their location's PM 2.5 spontaneously, which significantly reduce the costs and increase the accuracy of results [18], [29].

In participatory sensing networks and crowd sensing networks, portable devices can directly connect to Internet to share data, therefore the focus of participatory sensing networks and crowd sensing networks is not how to forward the data collected by portable devices to centralized station, but how to motivate users to participate in the process of collecting sensing data. A practical adaptive incentive strategy to motivate the enthusiasm of users is to offer rewards to data reporters [18]. In this strategy, data are linked to rewards (e.g. money). If we want to collect more data of a certain part (e.g. the PM 2.5 of a certain area), we can increase the rewards to reporters who report this part of data. Conversely, if we want to collect less data of another certain part, we can decrease the rewards to reporters who reports this part of data. The adaptive adjustment of rewards will finally level off to market equilibrium, which will increase the overall efficiency.

In the SWDCP-SCmules scheme, we don't adopt this adaptive incentive strategy. The strategy we adopt is that fixed rewards are given to users who are willing to install transceivers and participate in the collection of sensing data, i.e. SCmules. In this strategy, the existence of transceivers is nearly transparent to users. Transceivers automatically sense and collect data according to unified arrangement from data collector. Users don't need to worry about the running of transceivers, to get rewards, what they need to do is to carry the transceivers when they are moving around the city. This unified strategy can increase the overall efficiency and avoid Prisoner's dilemma.

There is a problem in the participatory sensing networks and crowd sensing networks. That is, there are some hotspot areas in the networks where more portable devices locate. Data from hotspot areas is much easier to be collected than data from other areas. If all portable devices report the same amount of data, the amount of data we collected from hotspot areas will be excessive, but the amount of data we collected from other areas will be not enough [26]. SWDCP-SCmules scheme also have similar problem. Tham *et al.* proposed a data collection scheme based on Quality of Contribution which can balance the collection of sensing data in its covered area [29]. Besides this, Quality of Information is also proposed to address this problem [33].

3) VEHICULAR MOBILE NETWORKS

In vehicular mobile networks, vehicles spontaneously connect with each other via wireless networks to exchange data [34]. A wide range of applications can be realized by vehicular mobile networks, such as traffic jam reporting [35]. At present, the research of vehicular mobile networks focus on the communication of vehicles to vehicles (V2V) and vehicles to devices (V2D).

SWDCP-SCmules scheme can be implemented by vehicular mobile networks. Taxis are typical SCmules that can be used in this scheme. However, unlike traditional vehicular mobile networks, the main communication objects of this scheme are intelligent devices and data centers.

4) DELAY TOLERANT NETWORKS

SWDCP-SCmules scheme can be regarded as a special case of DTNs from the perspective of transmission delay requirements [17]. In traditional DTN, both the data generator and receiver are mobile, when two mobile nodes encounter with each other, they can exchange and relay their data to achieve the goal of data diffusion [23]. The major feature of DTNs is that data's transmission delay and the aim of data diffusion has no strict requirement. In most cases, the delay of data transmission is not an important measure of network performance and the aim of data transmission is just diffusing data as large as possible.

However, in the SWDCP-SCmules scheme, the positions of data generators (i.e. intelligent devices) are fixed, the positions of data transmitters (i.e. SCmules) are mobile and the positions of data receivers (i.e. data centers) are fixed. The aim of data transmission is to obliviously transmit data from data generators to data receivers by data transmitters. To achieve this aim, we don't put any management strategy on data transmitters, the process of data collection is totally incidental and oblivious.

B. MACHINE LEARNING

Tom Mitchell gives a received definition of machine learning in 1998: A computer program is said to learn from experience E with respect to some task T and some performance measure P, if its performance on T, as measured by P, improves with experience E [36]. As will be illustrated in section 5.3, SA-PA algorithm can be categorized into machine learning program according to this definition.

One typical work flow of machine learning program is to find an appropriate hypothesis from hypothesis set and use learning algorithm to learn the parameters of hypothesis based on training set and generate the final hypothesis [37]. The SA-PA algorithm conforms to this work flow according to the illustration of section 5.3.

C. SIMULATED ANNEALING METAHEURISTIC

Optimization problems are the problems of finding the best solution from all feasible solutions [38], the priority assignment problem is a typical optimization problem.

Metaheuristics are one of the practical techniques to solve optimization problems. Typical metaheuristics include simulated annealing, tabu search, ant colony optimization and so on. Although the quality of solution provided by metaheuristics have no guarantee, for most computationally complicated optimization problems, metaheuristics can lead to better solution than other algorithm design method when given limited time [38]. So metaheuristics are perfect candidates for solving the priority assignment problem due to the computationally complication.

Simulated annealing is a metaheuristic proposed by Suman and Kumar [39]. It can be regarded as an improvement of local search algorithm. The aim of local search algorithm, such as hill climbing method [39], is to find local optimums by moving from solution to solution in search space using local changes. Simulated annealing introduces acceptance probability into local search algorithm to allow inferior moves to be accepted. A move will be accepted in the following acceptance probability $\mathfrak{X}$:

$$\mathfrak{X} = \begin{cases} 1 - e^{\frac{\Delta}{T_0}}, & \Delta < 0 \\ 1, & \Delta \geq 0 \end{cases}$$

where Δ is the performance of the move, or more simply speaking, the difference between the height after taking the move and the height before taking the move, T_0 is the temperature parameter, which controls the probability of taking an inferior move [39].

To apply simulated annealing metaheuristic to solve optimization problems, three elements need to be specified: configuration, evaluation function and neighborhood function. Configuration is the representation of feasible solution, all configurations form the search space. Evaluation function is the measure of optimization target. Neighborhood function defines the way moving from solution to solution.

III. THE SYSTEM MODEL AND PROBLEM STATEMENT

A. THE SYSTEM MODEL

In SWDCP-SCmules scheme, intelligent devices are embedded in the infrastructures of city. Assume there are m intelligent devices in total, which consist of set $\mathcal{S} = \{S_1, S_2, \ldots, S_m\}$. These intelligent devices are self-powered, which is possible nowadays due to the development of energy harvesting technology [40]. The intelligent devices have two main functions: detecting the status of infrastructures and sending data/messages via short-range wireless communications (due to the limitation of energy, the communication range is limited). For example, intelligent devices deployed in garbage cans can detect the level of trashes periodically and, when SCmules pass by them, they can forward the data/messages of the trash level to SCmules via wireless communications.

Every intelligent device is assigned with a unique priority, which is a measure of the importance of the data/messages generated by this intelligent device. For intelligent device S_i, its corresponding priority is P_i. Suppose the set of priority is

$\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ and its corresponding m-dimensional vector is $\vec{\mathcal{P}} = [P_1, P_2, \ldots, P_m]$. There is a one-to-one correspondence between $\mathcal{P}$ and $\vec{\mathcal{P}}$. We will not explicitly distinguish $\mathcal{P}$ and $\vec{\mathcal{P}}$ in following context since both $\mathcal{P}$ and $\vec{\mathcal{P}}$ can be used to express priority assignments, i.e. priority table. Obviously, from the perspective of linear algebra, all possible priority assignments form a subset of m-dimensional space.

Besides intelligent devices, there are k data centers distributed in fixed locations of Smart City, especially the downtown areas. The i-th data center is h_i, all data centers form the set $\mathcal{H} = \{h_1, h_2, \ldots, h_k\}$. As described in section 1, data centers are special computing and processing nodes which can connect with cloud tier via high-speed networks to forward data/messages. Their major responsibilities are receiving data/messages buffered by nearby SCmules and then forwarding them to cloud tier, which is accessible to corresponding municipal departments. Loosely speaking, data centers can be viewed as the destination of transmission: data/messages generated by intelligent devices are eventually forwarded to data centers via SCmules. Data centers may receive a lot of duplicated information since several SCmules may buffer the same data/messages when passing by the same intelligent devices in the same time period. This phenomenon is unwanted because it wastes energy and occupies precious limited storage of SCmules.

There are n total SCmules which consist of set $\mathcal{V} = \{V_1, V_2, \ldots, V_n\}$. SCmules, which can be regarded as mobile IoT nodes, move independently in Smart City. They are equipped with short-range transceivers. In the process of motion, when SCmules pass near intelligent devices, transceivers can automatically detect the active intelligent devices within their communication range, pick up the data/messages which these active intelligent devices are sending and then buffer them into their storage. Similarly, when SCmules pass near data centers, the transceiver can detect the existence of these data centers within their communication range, dump all the data/messages buffered in their storage to data centers and then clear their storage. To simplify the model, we assume the data transmission processes between intelligent devices and SCmules and the data transmission process between data centers and SCmules are instant. This is feasible in most cases due to the rapid development of new radio technologies such as Ultra-Wideband (UWB) [41], even if the worst situation occurs, such as packet loss or unfinished packet transmission, SCmules can simply discard these damaged packets since the oblivious and incidental data collection feature of SWDCP-SCmules scheme.

The storages of SCmules are limited. Suppose SCmule V_i can store Q_i data/messages at most and all Q_i form set $\mathcal{Q} = \{Q_1, Q_2, \ldots, Q_m\}$. When the storage is full and new valuable data/messages are picked up. SCmules will use greedy selection principle, which is described in section 1 and section 4.4, to maximize the sum of priorities of data/messages buffered.

We refer to the quintuple $\mathbb{C} = (\mathcal{S}, \mathcal{P}, \mathcal{H}, \mathcal{V}, \mathcal{Q})$ as a configuration, which is a concrete deployment of SWDCP-SCmules scheme in Smart City.

B. PROBLEM STATEMENTS

Simply speaking, the biggest problem of SWDCP-SCmules scheme is how to make it effective and efficient. For a certain configuration $\mathbb{C} = (\mathcal{S}, \mathcal{P}, \mathcal{H}, \mathcal{V}, \mathcal{Q})$ in Smart City, the only element that can be manually changed to improve performance is priority assignment $\mathcal{P}$. After all, intelligent devices, data centers and transceivers are deployed in advance and cannot be changed arbitrarily. That is to say, the priority assignment $\mathcal{P}$ is the only factor that can be easily altered to improve the performance of SWDCP-SCmules scheme as illustrated in section 1.

However, the meaning of effectiveness and efficiency is ambiguous since different scenarios have their own different concrete meaning. To specify the meaning of effectiveness and efficiency, we introduce a universal optimization target function $\mathcal{J}$ to quantify them. $\mathcal{J}$ can be thought as a quantified performance measure of SWDCP-SCmules scheme. The larger $\mathcal{J}$ is, the more effective and efficient SWDCP-SCmules scheme is. We convert the problem of making SWDCP-SCmules scheme effective and efficient to an optimization problem of $\mathcal{J}$ with respect to $\mathbb{C}$ (In fact, $\mathbb{C}$ is dominated by $\mathcal{P}$ as other elements cannot be controlled easily).

For example, in most cases, the interpretation of effectiveness and efficiency corresponds to high collection rate and low redundancy rate. Collection rate, redundancy rate and their synthesized rate are formally defined as follow:

1) COLLECTION RATE $\mathcal{C}_{\mathcal{T}}^{\mathbb{C}}$

Collection rate is the number of distinct data/messages (excluding duplicated data/messages) collected by data centers in a given time interval. It is obvious that the higher collection rate is, the less data/messages loss in the process of data transmission of SCmules. Therefore, we should maximize collection rate.

For a given time interval $\mathcal{T} = [t_x, t_y)$ where t_x and t_y are timestamps, assume the total number of data/messages generated by all intelligent devices is $\mathcal{G}_{\mathcal{T}}^{\mathbb{C}}$, the total number of distinct data/messages collected by data centers is $k_{\mathcal{T}}^{\mathbb{C}}$, we can define the collection rate $\mathcal{C}_{\mathcal{T}}^{\mathbb{C}}$ as

$$\mathcal{C}_{\mathcal{T}}^{\mathbb{C}} = \frac{k_{\mathcal{T}}^{\mathbb{C}}}{\mathcal{G}_{\mathcal{T}}^{\mathbb{C}}}.$$

2) REDUNDANCY RATE $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$

Redundancy rate is the ratio of the number of duplicated data/messages collected by data centers to the number of data/messages collected by data centers in a given time interval. It is obvious that the more duplicated data/messages data centers collect, the lower storage utilization efficiency SCmules are and the more energy the whole system wastes. Therefore, we should minimize redundancy rate.

For a given time interval $\mathcal{T} = [t_x, t_y)$ where t_x and t_y are timestamps, the total number of data/messages collected by data centers is $\mathcal{K}_{\mathcal{T}}^{\mathbb{C}}$, the total number of duplicated data/messages collected by data centers is $\mathcal{K}_{\mathcal{T}}^{\mathbb{C}} - \ell_{\mathcal{T}}^{\mathbb{C}}$, we can define the redundancy rate $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$ as

$$\mathcal{R}_{\mathcal{T}}^{\mathbb{C}} = \frac{\mathcal{K}_{\mathcal{T}}^{\mathbb{C}} - \ell_{\mathcal{T}}^{\mathbb{C}}}{\mathcal{K}_{\mathcal{T}}^{\mathbb{C}}}$$

In the following context, we will also use effective collection rate $\overline{\mathcal{R}}_{\mathcal{T}}^{\mathbb{C}}$ as a substitution of $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$, which is defined as

$$\overline{\mathcal{R}}_{\mathcal{T}}^{\mathbb{C}} = 1 - \mathcal{R}_{\mathcal{T}}^{\mathbb{C}} = 1 - \frac{\mathcal{K}_{\mathcal{T}}^{\mathbb{C}} - \ell_{\mathcal{T}}^{\mathbb{C}}}{\mathcal{K}_{\mathcal{T}}^{\mathbb{C}}} = \frac{\ell_{\mathcal{T}}^{\mathbb{C}}}{\mathcal{K}_{\mathcal{T}}^{\mathbb{C}}}$$

Obviously, the higher $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$ is, the lower $\overline{\mathcal{R}}_{\mathcal{T}}^{\mathbb{C}}$ is. Conversely, the lower $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$ is, the higher $\overline{\mathcal{R}}_{\mathcal{T}}^{\mathbb{C}}$ is.

By the way, it is worth to mention that the way we define redundancy rate is similar to the definition of repurchase rate in the field of management, which proves the definition of redundancy rate is reasonable.

Based on the definitions of collection rate and redundancy rate, we can define three distinct optimization target functions:

(1) Collection rate optimization target function $\mathcal{J}_{\mathcal{C}}$

$$\mathcal{J}_{\mathcal{C}_{\mathcal{T}}}^{\mathbb{C}} = \mathcal{C}_{\mathcal{T}}^{\mathbb{C}}$$

The optimization target of $\mathcal{J}_{\mathcal{C}}$ is to maximize the overall collection rate.

(2) Effective collection rate optimization target function $\mathcal{J}_{\overline{\mathcal{R}}}$

$$\mathcal{J}_{\overline{\mathcal{R}}_{\mathcal{T}}}^{\mathbb{C}} = 1 - \mathcal{R}_{\mathcal{T}}^{\mathbb{C}} = \overline{\mathcal{R}}_{\mathcal{T}}^{\mathbb{C}}$$

The optimization target of $\mathcal{J}_{\overline{\mathcal{R}}}$ is to minimize redundancy rate or maximize effective collection rate.

(3) Synthesized optimization target function $\mathcal{J}_{\mathcal{S}}$

We can synthesize the two optimization targets, i.e. maximizing collection rate $\mathcal{C}_{\mathcal{T}}^{\mathbb{C}}$ and minimizing redundancy rate $\mathcal{R}_{\mathcal{T}}^{\mathbb{C}}$, to form a synthesized rate and make a concrete synthesized optimization target function $\mathcal{J}_{\mathcal{S}}$:

$$\mathcal{J}_{\mathcal{S}_{\mathcal{T}}}^{\mathbb{C}} = \lambda_1 \mathcal{C}_{\mathcal{T}}^{\mathbb{C}} + \lambda_2 \left(1 - \mathcal{R}_{\mathcal{T}}^{\mathbb{C}}\right)$$

where $\lambda_1 + \lambda_2 = 1$, $\lambda_1 \geq 0$ and $\lambda_2 \geq 0$. Maximizing $\mathcal{J}_{\mathcal{S}}$ is equivalent to maximizing $\mathcal{C}$ and minimizing $\mathcal{R}$ in the same time. The higher $\mathcal{J}_{\mathcal{S}}$ is, the higher $\mathcal{C}$ is and the lower R is.

$\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$ will be used in the following context as examples of optimization target functions to prove the effectiveness and efficiency of SWDCP-SCmules scheme and SA-PA algorithm.

In summary, we introduce a universal optimization target function $\mathcal{J}$ to measure the performance of a certain configuration of SWDCP-SCmules scheme. In different applications, the meaning of effectiveness and efficiency is different, we can use different $\mathcal{J}$ to quantify them. Optimizing $\mathcal{J}$ with respect to configuration $\mathbb{C}$ (In fact, $\mathbb{C}$ is dominated by $\mathcal{P}$ as other elements cannot be controlled easily) can acquire an optimized configuration that significantly improve the performance of SWDCP-SCmules scheme. Therefore, the formal problem we have to tackle with is

$$\max_{\mathbb{C}} \mathcal{J}$$

We propose the SA-PA algorithm to solve this problem in section 5.

TABLE 1. Main notations.

Symbol	Description
m	The number of intelligent devices in Smart City
k	The number of data centers in Smart City
n	The number of SCmules in Smart City
$\mathcal{S}$	The set of intelligent devices
S_i	The i-th intelligent device in $\mathcal{S}$
$\mathcal{P}$	The set of priorities of intelligent devices
$\vec{\mathcal{P}}$	The vector form of $\mathcal{P}$
P_i	The priority of intelligent device S_i
$\mathcal{H}$	The set of data centers
h_i	The i-th data center in $\mathcal{H}$
$\mathcal{V}$	The set of SCmules in Smart City
V_i	The i-th SCmule in $\mathcal{V}$
$\mathcal{Q}$	The set of storage limits of SCmules
Q_i	The storage limit of SCmule V_i
$\mathbb{C}$	The configuration of Smart City
$\mathcal{J}$	The optimization target function
$\mathcal{C}$	Collection rate
$\mathcal{R}$	Redundancy rate
$\overline{\mathcal{R}}$	Effective collection rate
$\mathcal{J}_{\mathcal{S}}$	Synthesized optimization target function
$\mathcal{J}_{\mathcal{C}}$	Collection rate optimization target function
$\mathcal{J}_{\overline{\mathcal{R}}}$	Effective collection rate optimization target function

IV. SWDCP-SCMULES SCHEME

A. OVERVIEW

To realize the intelligent management of city and improve the overall social welfare, we can deploy intelligent devices to infrastructures of city and make them monitor and report the status of these infrastructures to corresponding departments, of which responsibility is to maintain the infrastructures' normal function. As illustrated in section 1, we refer to this abstract paradigm as Social welfare data collection paradigm (SWDCP paradigm). Many applications can be realized by this paradigm, such as the intelligent monitoring of the conditions of street lights.

To give a concrete, feasible and economical scheme to implement SWDCP paradigm, many related researches have been done as introduced in section 1. In this paper, we propose a Social Welfare Data Collection

Scheme based on Storage-Constrained Oblivious Data Mules (SWDCP-SCmules scheme), which can be regarded as a refined model proposed by Bonola *et al.* [17], to collect data/messages distributed in the sparse network formed by intelligent devices. The main feature of this scheme is that oblivious data mules are storage-constrained, which is much more realistic than previous researches. In the following context, we refer to this kind of oblivious data mules as SCmules.

Below we describe this scheme on the whole:

Intelligent devices are embedded into the infrastructures of Smart City to detect their status. Once the status meets a certain condition, the communication function of intelligent devices will be activated and tries to report data/messages to nearby SCmules via short-range wireless communication. SCmules are mobile IoT nodes moving in Smart City, they are equipped with transceivers which enable them to opportunistically communicate with nearby nodes. When SCmules pass near active intelligent devices which are trying to report data/messages, they will detect the existence of them and pick up data/messages sent by them obliviously and incidentally. As the storage is limited, when receiving new data/messages, SCmules will determine whether the new data/messages will be buffered or not according to the remaining storage space they have and the priority of the intelligent devices generated these data/messages. The selection principle is to greedily maximize the sum of priorities of data/messages stored. We refer to this principle as greedy selection principle. Besides intelligent devices and SCmules, there are many data centers deployed in fixed locations of Smart City, especially downtown areas. Data centers can be viewed as the destination of data collection, they are special computing and processing nodes that can connect with cloud tier via high-speed network to forward data/messages. When SCmules pass near them, they will dump all data/messages buffered in their storage to them and then clear their buffer. Data centers received these data/messages will forward them to cloud tier, which is accessible to municipal departments. Once corresponding municipal departments see these data/messages, they will take measures to maintain the related infrastructures.

To illustrate the scheme in a concrete way, below is an example of the application of monitoring the health status of allée trees using SWDCP-SCmules scheme (see Figure 2): Intelligent devices are deployed on allée trees to monitor their health status, once these devices detect that these allée trees are in unhealthy condition, such as water shortage, pest threaten, the communication function of these intelligent devices will be activated immediately and try to report the messages via short-range wireless communication. Taxis equipped with transceivers, which are typical types of SCmules, move in Smart City to send passengers to destinations, when taxis pass near these active intelligent devices, which are trying to report messages, transceivers will detect the existence of them, pick up the messages obliviously and incidentally and then buffer them according to greedy selection principle. In the process of sending passengers to destinations, taxis may also pass near some data centers,

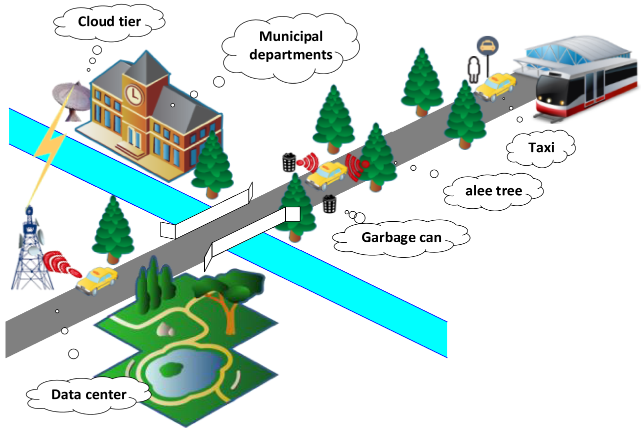


FIGURE 2. A concrete example of SWDCP-SCmules scheme.

which are distributed in fixed locations of Smart City. When a certain data center is nearby, transceivers will detect the existence of it and dump all data/messages buffered in storage to it. After receiving these messages, data centers will simply process them, such as filter out duplicated messages and then forward them to cloud tier via high-speed network connections, gardening department will receive this message from cloud tier and then send staffs to water the allée trees.

In rest of this section, we will give a detailed introduction of how intelligent devices, data centers and SCmules work in section 4.2, 4.3 and 4.4, respectively.

B. INTELLIGENT DEVICES

Various intelligent devices are embedded in the infrastructures of Smart City to enhance their functionality and productivity. Most of them are self-powered. They have two main functions: detecting the status of the infrastructures and communicating via short-range wireless networks.

To illustrate the running process of intelligent devices, we divide their running process into three states: detecting state, latency state and transmission state. In detecting state, the detecting function of intelligent devices is activated to detect the status of infrastructures. If a certain condition is satisfied (e.g. the infrastructures turn into bad condition or the current time is the arranged time of reporting), the latency state is activated. In latency state, the communication function of intelligent devices is active and waits for the appearance of SCmules within communication range to transmit data/messages to them. This state may consume unnecessary energy since intelligent devices have no idea of when SCmules will pass near them, many researches have been done to solve this problem in ad hoc networks, which can also be used to address the problem in this situation [42], [43]. Once a certain SCmule appears in the communication range of intelligent devices, the intelligent devices will turn into transmission state imediately, it will then communicate directly with the SCmule to send data/messages to it. Figure 3 illustrates the running process of intelligent devices.

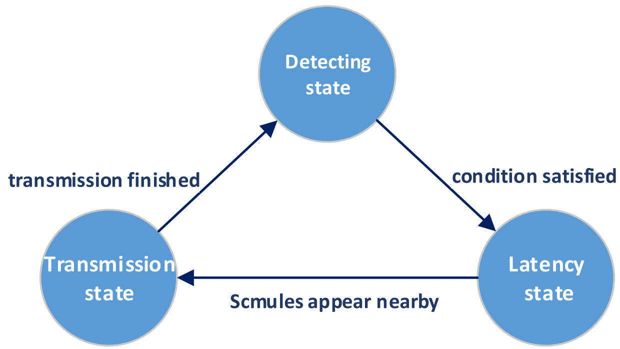


FIGURE 3. The state diagram of intelligent devices.

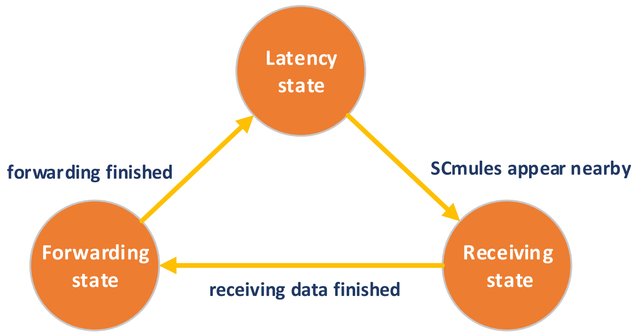


FIGURE 4. The state diagram of data centers.

Algorithm 1 The Running Process of Intelligent Devices

```
While true
    switch into detecting state
    detect the infrastructure's status
    If infrastructure's status satisfies a certain condition
        switch into latency state
        While true
            If SCmule V_i enters the communication range
                switch into transmission state
                report the infrastructure's status to V_i
            End
        End
    End
End
```

Algorithm 1 is the pseudo-code of the running process of intelligent devices.

C. DATA CENTERS

Data centers are special computing and processing nodes distributed in fixed locations of Smart City. They are connected with cloud tier using high-speed networks to forward data/messages and can detect the existence of SCmules within their communication range and receive data/messages from them via short-range wireless communication. They have sufficient stable power supply because they are installed in fixed locations, so their communication range is usually larger than intelligent devices.

The running process of data centers can be divided into three states (see Figure 4): latency state, receiving state and forwarding state. Unlike intelligent devices, data centers are always in latency state to detect the appearance of SCmules within their communication range. Once SCmules pass near them, they will switch into receiving state in which they communicate with SCmules' transceivers to receive all data/messages buffered in their storage. After receiving data/messages from SCmules, data centers will send ACK as a feedback to SCmules to tell them the data/messages are received and they can safely clear their buffer and then switch into forwarding state in which they process

Algorithm 2 The Running Process of Data Centers

```
While true
    switch into latency state
    detect SCmules which enter communication range
    If SCmule V_i enters communication range
        wait for communication request from V_i
        switch into receiving state
        receive data/messages dumped by V_i
        switch into forwarding state
        process received data/messages
      send received data /messages to cloud tier
    End
End
```

the received data/messages, such as filtering out duplicated information, and then forward them to cloud tier. Municipal departments can access the data/messages stored in cloud tier. Once they check these data/messages in cloud tier, they will take measures to maintain related infrastructures. Usually, data centers are deployed in the downtown areas of Smart City, i.e. hotspot areas, where SCmules travel more frequently than other areas, to boost the efficiency of data collection.

Algorithm 2 is the pseudo-code of the running process of data centers.

D. SCmules

SCmules are mobile IoT nodes moving in Smart City, they are equipped with transceivers which enable them to opportunistically communicate with nearby nodes (including intelligent devices and data centers). Typical SCmules include taxis, private vehicles, buses and people holding smart mobile phone. When SCmules pass near intelligent devices which are in latency state, they will detect the existence of them and pick up data/messages they are reporting obliviously and incidentally. As the storage is limited, SCmules have to face with the problem whether to buffer the new picked-up data/messages or not. More specifically, in the situation that their storage is full and new more valuable data/messages are picked up, they have to discard some less important data/messages buffered in storage to make room for them. SCmules use greedy selection principle, which means greedily maximize the sum of priorities of data/messages buffered, to guide the selection. That is to say, when the storage is not full, SCmules will store data/messages as much as possible, but when the storage is full, SCmules will compare the priority of the intelligent devices generating these new picked-up data/messages with the priorities of intelligent devices generating the data/messages buffered in SCmules' storage according to priority table and then make decision: if the priority of these new picked-up data/messages is less than any priorities of the data/messages buffered, the new picked-up data/messages will be discarded directly, otherwise, the data/messages in storage with the least priority will be discarded to make room for the new picked-up data/messages and then the new more valuable picked-up data/messages will be buffered in the storage. When SCmules pass near data centers, SCmules will send request for dumping data/messages to data centers, if data centers approve of the request, they will dump all data/messages buffered in their storage to data centers and then clear the storage.

The running process of SCmules, or more accurately the transceivers deployed in SCmules can be divided into three states: latency state, buffering state and dumping state (see Figure 5). In latency state, SCmules are detecting the existence of intelligent devices or data centers within their communication range. When detecting that intelligent devices are nearby, SCmules rapidly switch into buffering state to pick up new data/messages and then use greedy selection principle to determine whether to buffer them or not. When detecting that data centers are nearby, SCmules will switch into dumping state and then dump all the data/messages stored to data centers.

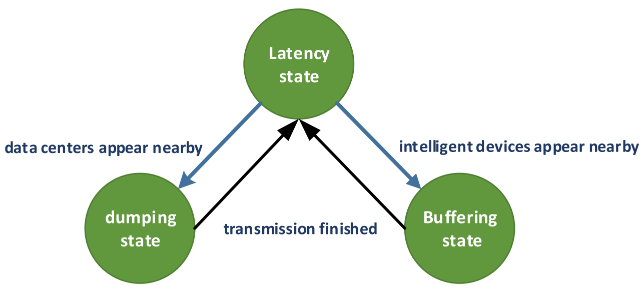


FIGURE 5. The state diagram of SCmules.

Algorithm 3 is the pseudo-code of the running process of SCmules, or more accurately, the transceivers deployed in SCmules.

Algorithm 3 The Running Process of SCmules' Transceivers

While true
  switch into latency state
  detect the existence of intelligent devices or data centers
  If intelligent device S_i in latency state enters communication range
    switch into buffering state
    pick up data/message $\mathfrak{d}$ which is sent by S_i
    If SCmule's storage is full
      select data/message $\mathfrak{e}$ with the least priority $\wp$ from SCmule's storage
      If $P_i < \wp$
        discard data/message $\mathfrak{d}$
      Else
        discard data/message $\mathfrak{e}$ from storage
        store $\mathfrak{d}$ into SCmule's storage
      End
    Else
      store $\mathfrak{d}$ into SCmule's storage
    End
  Else If data center h_i enters communication range
    switch into dumping state
    send dumping request to h_i
    If h_i approve of the request
      dump all data/messages stored to h_i
    End
  End
End

V. SA-PA ALGORITHM

A. OVERVIEW

Section 4 gives an almost comprehensive introduction to SWDCP-SCmules scheme, but sidestep the problem of priority assignment. As illustrated in section 1, the priority assignment problem is key to SWDCP-SCmules scheme since it almost determines the final performance of this scheme. In other words, as illustrated in section 3.2, the performance of SWDCP-SCmules scheme is quantified as an optimization target function $\mathfrak{J}$, the priority assignment almost determines the final value of $\mathfrak{J}$. However, the traditional approach, which is finding effective patterns and then using these patterns to design algorithms, to solve the priority assignment problem is unfeasible because of the ambiguity of the meaning of effectiveness and efficiency, the difficulty of finding patterns and the variability of patterns as described in section 1 [44].

To solve the priority assignment problem, we try to convert the priority assignment problem to an optimization problem based on the idea from machine learning and use simulated annealing metaheuristic to design a universal algorithm that can automatically find a well-performed priority assignment with respect to various optimization targets, which are modeled by optimization target function $\mathfrak{J}$, based on the social-aware patterns lying in the GPS trajectory data of SCmules in the past time.

Concretely speaking, first, we quantify the performance of the SWDCP-SCmules scheme as a universal optimization target function $\mathfrak{J}$ of priority assignment $\vec{\mathcal{P}}$; then, we use the past GPS trajectory data of SCmules to train a priority assignment $\vec{\mathcal{P}}$ that can optimize $\mathfrak{J}$ with respect to $\vec{\mathcal{P}}$ and predict that it will also achieve good performance in the future because the future shares similar patterns with the past. Because of the complexity of the computation of $\mathfrak{J}$, we use simulated annealing metaheuristic to search for $\vec{\mathcal{P}}$ that can optimize $\mathcal{J}$, in search space. We refer to this solution to priority assignment problem as Simulated Annealing for Priority Assignment Algorithm (SA-PA algorithm).

Why a priority assignment perform well in the past can also perform well in the future? Because the running process of SWDCP-SCmules scheme in the future is similar to the running process in the past under the same configuration $\mathbb{C}$. In other words, they have similar social-aware patterns. Usually, similar social-aware patterns will lead to similar results if the configuration $\mathbb{C}$ is fixed. We use a manually observed social-aware pattern to illustrate this statement:

As illustrated in section 1, the frequency that SCmules pass by downtown areas, i.e. hotspot areas, is much higher than the frequency of other areas, especially remote areas. If we deploy data centers in hotspot areas, then the data/messages generated by intelligent devices which are near data centers are more likely to be collected than those generated by intelligent devices located in other areas. This lead to a bad phenomenon that data centers may receive many duplicated data/messages generated by the intelligent devices located near them, but receive a few data/messages from intelligent devices located in other areas. Therefore, we should decrease the priorities of intelligent devices near data centers and increase the priorities of intelligent devices far from data centers: Decreasing the priorities of intelligent devices near data centers will not interfere with the data/messages collection of those intelligent devices, on the contrary, it can decrease the duplicated data/messages collected by data centers (i.e. redundancy rate) and thereby decreasing unnecessary energy consumption. Increasing the priorities of intelligent devices far from data centers will decrease data/messages loss and thereby increasing collection rate. Due to the pattern that the locations of hotspot areas in the future are usually the same as the locations of hotspot areas in the past, assigning priorities according to the priority assignment method mentioned above can achieve good results both in the past and in the future.

However, we can't just rely on the patterns simply acquired by our observation to design algorithm for priority assignment problem. As described in section 1, the pattern mentioned above is just one of the huge number of patterns lying in essence. In fact, most of patterns are difficult for human intelligence to observe and even understand, not to mention that different definitions of effectiveness and efficiency (i.e. optimization target function $\mathcal{J}$) need the support of different patterns, which means a former observed pattern may become useless in new applications. Besides that, some patterns may change occasionally or periodically. Based on the above inference, we could conclude that if we simply use the patterns acquired by our observation to design algorithm solving the priority assignment problem, which is the traditional approach for algorithm design, there must be some unforeseen problems lying in the algorithm that are not taken into consideration.

We follow the above example to illustrate two unforeseen problems lying in priority assignment method proposed in that example:

The performance of the priority assignment method in that example is based on the assumption that all data centers are located in hotspot areas of Smart City. This strong assumption is not that realistic. In fact, it is impossible to deploy data centers for all hotspot areas since there may be temporary hotspot areas in a certain time period, e.g. the districts near natatoriums in hot summer (see section 1). For hotspot areas without the deployment of data centers, the priority assignment method proposed in that example is inefficient because intelligent devices in those hotspot areas are assigned with high priorities since they are far from data centers, but their data/messages may be repeatedly collected by SCmules since they are in hotspot areas. Therefore, these hotspot areas will contribute a lot of redundant data/messages to data centers and thus increase redundancy rate significantly.

Even if the assumption is satisfied, i.e. all hotspot areas are deployed with data centers, some data/messages generated by intelligent devices located near data centers may still have difficulties to be collected. For example, if the data center is deployed near the starting point of the one-way street but the intelligent device is deployed in the end point. Taxis, a typical kind of SCmules, moving along the one-way street will first pass by the data center and dump buffered data/messages to it and then pass by the intelligent devices and pick up data/messages from it. If the one-way street lead taxis from a downtown area to a remote area, although this intelligent device locates in a hotspot area and near a data center, we still can't guarantee that the data/messages from this intelligent device can be collected rapidly just like those generated by other ordinary intelligent devices located near data centers. However, according to the priority assignment method proposed in that example, this intelligent device will be assigned with low priority, its data/messages are easily replaced by data/messages generated by intelligent devices located in remote areas. This will lead to the loss of data/messages generated by this intelligent device (see Figure 6).

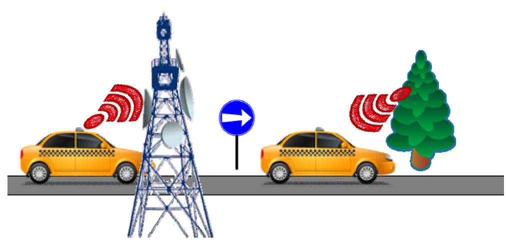


FIGURE 6. The one-way street in counterexample.

Unlike traditional algorithm design approach, machine learning can automatic find social-aware patterns, which are patterns that can reflect the social preferences of citizens in Smart City, by training hypothesis using training data. Using the idea from machine learning, we propose the SA-PA algorithm which is a universal algorithm for priority assignment problem. No matter what optimization target $\mathcal{J}$ is, how difficult the observation of potential social-aware patterns are and how social-aware patterns lying in the GPS trajectory data change constantly, SA-PA algorithm can always automatically learn social-aware patterns from training data (i.e. the GPS trajectory data of SCmules) without manual teaching and then use these social-aware patterns to guide priority assignment.

An important attribute of SA-PA algorithm is its social-awareness since it can automatically learn citizens' social behaviors, which can be formally expressed by social-aware patterns, from GPS trajectory data of SCmules in Smart City. As illustrated in section 1, the basis of learning is that there are a lot of social-aware information lying in the data. This is obviously true because a lot of instances can be provided. For example, if SCmules are taxis, the preferred places citizens in Smart City usually go to, i.e. the hotspot areas, can be analyzed from the frequencies that taxis travel to a certain place with; if SCmules are private vehicles, the places the owners of the private vehicles prefer to go to can be acquired from the trajectories of these private vehicles. The key to the success of SA-PA algorithm is that it can automatically discover these social-aware information from data, express them in the form of hypothesis (the meaning of hypothesis is hard to explain and sometimes even beyond the understanding of human beings) and then use them to improve the priority assignment. This attributes to the magic of machine learning.

In section 5.2, we formalize the priority assignment problem. In section 5.3, we introduce SA-PA algorithm in detail.

B. FORMAL PROBLEM DESCRIPTION

Informally speaking, the problem solved by SA-PA algorithm can be described as follow:

Given the following four elements

- the information of the intelligent devices (e.g. positions, communication range);
- the information of data centers (e.g. positions, communication range);
- the list of all registered SCmules and their configurations (e.g. storage size, communication range of transceivers);
- GPS trajectory data of SCmules in past time interval $[t_1, t_2)$ where t_1 and t_2 are timestamps and $t_1 < t_2$.

We want to find a priority assignment that can optimize the optimization target (e.g. maximize collection rate, minimize redundancy rate) in the future time interval $[t_2, t_3)$ where t_2 and t_3 are timestamps and $t_2 < t_3$.

To introduce SA-PA algorithm, we need formalize the problem description above: As defined in section 3.1, the intelligent devices and data centers can be expressed as $\mathcal{S}$ and $\mathcal{H}$ respectively. All registered SCmules and their storage size can be expressed as $\mathcal{V}$ and $\mathcal{Q}$ respectively. To formalize the GPS trajectory of SCmule V_i in a given time interval, we define trajectory function $\mathcal{D}$ as follow:

$$\mathcal{D}(V_i, t_x) = (x, y)$$

The output of $\mathcal{D}$ is the geographic coordinate (x, y) of SCmule V_i in timestamp t_x. Therefore, the meaning of knowing the GPS trajectories of SCmule V_i in time interval $[t_1, t_2)$ can be formalized as knowing the value of $\mathcal{D}(V_i, t_x)$ where $V_i \in \mathcal{V}$ and $t_x \in [t_1, t_2)$.

Next, we formalize the notation of priority assignment. We follow the way we express priority assignment in section 3.1. That is, all priorities of intelligent devices form set $\mathcal{P} = \{P_1, P_2, \ldots, P_m\}$ where m is the number of intelligent devices and P_i is the priority assigned to intelligent device S_i. The corresponding vector form is $\vec{\mathcal{P}} = [P_1, P_2, \ldots, P_m]$. Obviously, there is a one-to-one-to-one correspondence among priority assignment, $\mathcal{P}$ and $\vec{\mathcal{P}}$. It is worth to mention all possible priority assignments form a subset of m-dimensional space if $\vec{\mathcal{P}}$ is used to express priority assignments.

In the informal problem description, the meaning of optimization target is not specified, which means any concrete target, such as maximize collection rate or minimize redundancy rate, can be embedded into the problem and then be solved by SA-PA algorithm. As illustrated in section 3.2, we use a universal optimization target function $\mathcal{J}$ to model the unspecified optimization target. Based on $\mathcal{J}$, if $\mathcal{P}$ is known, the configuration $\mathbb{C} = (\mathcal{S}, \mathcal{P}, \mathcal{H}, \mathcal{V}, \mathcal{Q})$ is determined and we can directly compute the value of optimization target function in time interval $[t_2, t_3)$, which is $\mathcal{J}^{\mathbb{C}}_{[t_2,t_3)}$.

To illustrate the meaning of $\mathcal{J}$, three concrete examples of $\mathcal{J}$ are introduced in section 3.2, i.e. $\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$, which will be used in section 6 to conduct experiments.

To simplify expressions, we introduce the simplified notation of optimization target function. Before doing that, we first introduce the simplified function notations of collection rate and redundancy rate:

$$\mathcal{C}(\mathcal{P}, \mathcal{T}) = \mathcal{C}^{(\mathcal{S},\mathcal{P},\mathcal{H},\mathcal{V},\mathcal{Q})}_{\mathcal{T}}$$
$$\mathcal{R}(\mathcal{P}, \mathcal{T}) = \mathcal{R}^{(\mathcal{S},\mathcal{P},\mathcal{H},\mathcal{V},\mathcal{Q})}_{\mathcal{T}}$$

where $\mathcal{P}$ is priority assignment and $\mathcal{T}$ is time interval. $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{Q}$ are known elements of $\mathbb{C}$.

The simplified notation of optimization target function is

$$\mathcal{J}(\mathcal{P}, \mathcal{T}) = \mathcal{J}^{(\mathcal{S},\mathcal{P},\mathcal{H},\mathcal{V},\mathcal{Q})}_{\mathcal{T}}$$

where $\mathcal{P}$ is priority assignment and T is time interval. $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{Q}$ are known elements of $\mathbb{C}$.

As a result, the simplified notation of $\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$ are listed below:

$$\begin{aligned}
\mathcal{J}_{\mathcal{C}}(\mathcal{P}, \mathcal{T}) &= \mathcal{C}(\mathcal{P}, \mathcal{T}) \\
\mathcal{J}_{\overline{\mathcal{R}}}(\mathcal{P}, \mathcal{T}) &= 1 - \mathcal{R}(\mathcal{P}, \mathcal{T}) \\
\mathcal{J}_{\mathcal{S}}(\mathcal{P}, \mathcal{T}) &= \lambda_1 \mathcal{C}(\mathcal{P}, \mathcal{T}) + \lambda_2 (1 - \mathcal{R}(\mathcal{P}, \mathcal{T})) \\
&\qquad (\lambda_1 + \lambda_2 = 1, \lambda_1 \geq 0 \text{ and } \lambda_2 \geq 0)
\end{aligned}$$

In summary, the formal description of the problem solved by SA-PA algorithm can be stated as follow:

Given $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$, $\mathcal{Q}$ and the value of $D(V_i, t_x)$ where $V_i \in \mathcal{V}$ and $t_x \in [t_1, t_2)$, find a $\mathcal{P}_{best}$ which can maximize $\mathcal{J}(\mathcal{P}_{best}, [t_2, t_3))$.

C. FORMAL DESCRIPTION OF SA-PA ALGORITHM

To address the problem in section 5.2, we can use the GPS trajectory data in past time interval $[t_1, t_2)$ of all SCmules in $\mathcal{V}$ as training set and find a priority assignment which can achieve good performance in training set by automatically learning the social-aware patterns lying in training set. Due to the future shares similar social-aware patterns with the past, we claim this priority assignment will also achieve good performance in future time interval $[t_2, t_3)$. This is the sketch of SA-PA algorithm.

Below details the sketch of SA-PA algorithm:

Assume $\mathcal{P}$ is given, according to the formal problem description given in last subsection, all elements of configuration $\mathbb{C} = (\mathcal{S}, \mathcal{P}, \mathcal{H}, \mathcal{V}, \mathcal{Q})$ are known. Besides these, we also know the GPS trajectory data of SCmules in time interval $[t_1, t_2)$, i.e. the value of $\mathcal{D}(V_i, t_x)$ where $V_i \in \mathcal{V}$ and $t_x \in [t_1, t_2)$. Based on the given information, we can estimate the value of $\mathcal{J}(\mathcal{P}, [t_1, t_2))$, which is a measure of performance as described in section 5.1, by simulating the running process of SWDCP-SCmules scheme in time interval $[t_1, t_2)$.

However, we don't know which $\mathcal{P}$ will make training set performs well, i.e. maximize $\mathcal{J}(\mathcal{P}, [t_1, t_2))$. To find such $\mathcal{P}$ that can optimize $\mathcal{J}$, we convert the original problem to a optimization problem:

Given $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$, $\mathcal{Q}$ and the value of $\mathcal{D}(V_i, t_x)$ where $V_i \in \mathcal{V}$ and $t_x \in [t_1, t_2)$, find a priority assignment $\wp$ which can maximize $\mathcal{J}(\wp, [t_1, t_2))$. We say $\wp$ is a good approximation of $\mathcal{P}_{best}$, which can maximize $\mathcal{J}(\mathcal{P}_{best}, [t_2, t_3))$.

Note that $\mathcal{J}$ cannot be simply expressed by closed formula. To compute $\mathcal{J}$, we need to simulate the running process of Smart City based on the past GPS trajectory data. That's why we cannot use iterative optimization algorithms such as gradient descent and Newton method [45]. Metaheuristics are perfect solution to optimizing this kind of computational complicated function as described in section 2.3. Therefore, we use simulated annealing metaheuristic to solve this optimization problem.

To apply simulated metaheuristic to this optimization problem, we need to define configuration, evaluation function and neighborhood function as described in section 2.3. For this optimization problem, we can naturally define configuration as $\wp$, i.e. a priority assignment and define evaluation function as $\mathcal{J}(\wp, [t_1, t_2))$. The definition of neighborhood function is not as direct as configuration and evaluation function. We propose a neighborhood function based on swapping the priorities of two intelligent devices:

$$\Re\left(\left[\wp_1, \ldots, \wp_i, \ldots, \wp_j, \ldots, \wp_m\right], i, j\right) = \left[\wp_1, \ldots, \wp_j, \ldots, \wp_i, \ldots, \wp_m\right]$$

where $1 \leq i < j \leq m$.

In section 5.1, we have given an intuitive explanation of why SA-PA algorithm takes effect. In section 6.3, we conduct experiments to prove its effectiveness.

Algorithm 4 is the pseudo-code of SA-PA algorithm.

Algorithm 4 SA-PA Algorithm

Input: T_0, α, $maxMv$, $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$, $\mathcal{Q}$ and $\mathcal{D}(V_i, t_x)$ $(V_i \in \mathcal{V}, t_x \in [t_1, t_2))$
Output: $\wp_{best}$
1: $nbMv := 0$
2: $\wp_{current}$:=generateRandomPriorityAssignment($\mathcal{S}$)
3: $\wp_{best} := \wp_{current}$
4: **While** $nbMv < maxMv$
5: pick random number i and j $(1 \leq i < j \leq m)$
6: $\wp_{new} := \Re(\wp_{current}, i, j)$
7: $\Delta := \mathcal{J}(\wp_{new}, [t_1, t_2)) - \mathcal{J}(\wp_{current}, [t_1, t_2))$
8: **If** $\Delta \geq 0$ **or** $e^{\Delta/T_0} \geq rand(0, 1)$
9: $\wp_{current} := \wp_{new}$
10: $nbMv := nbMv + 1$
11: **If** $\mathcal{J}(\wp_{current}, [t_1, t_2)) \geq \mathcal{J}(\wp_{best}, [t_1, t_2))$
12: $\wp_{best} := \wp_{current}$
13: **End**
14: **End**
15: **End**
16: **Return** $\wp_{best}$

Parenthetically, the conditional statements of line 8 is equivalent to the acceptance probability illustrated in section 2.3.

Notice SA-PA algorithm uses idea from machine learning. It conforms to the received definition of machine learning proposed by Tom Mitchell, which is described in section 2.2 [36]. In SA-PA algorithm, E is the GPS trajectory data in time interval $[t_1, t_2)$, $\mathcal{P}$ is the optimization target function $\mathcal{J}$ and T is to find a well-performed priority assignment. From the perspective of hypothesis, a configuration $\mathbb{C}$ is a hypothesis and $\mathcal{P}$ is the parameters of the hypothesis, we use training set, i.e. the GPS trajectory data in time interval $[t_1, t_2)$, to train the hypothesis to find a well-performed $\mathcal{P}$ as the parameters of the hypothesis.

VI. EXPERIMENTAL RESULTS

A. OVERVIEW OF DATASET

To prove the effectiveness of the SWDCP-SCmules scheme and SA-PA algorithm, we use T-Drive trajectory dataset, which is provided by MSRA [46], [47], to do experiments. The T-Drive trajectory dataset contains GPS trajectories of 10357 taxis during the period of Feb. 2 to Feb. 8, 2008, in Beijing. The number of GPS waypoints in this dataset reaches up to 15 million and the total distance of the trajectories reaches up to 9 million kilometers. Figure 7 illustrates the dataset visually, different colors reflect different density distribution of the GPS waypoints.

Figure 8, which is extracted from the instruments of the dataset, shows the distribution of time interval between two consecutive waypoints. It indicates, in most cases, GPS devices equipped in taxis sample geographical data every one second or five seconds. This means these discrete waypoints

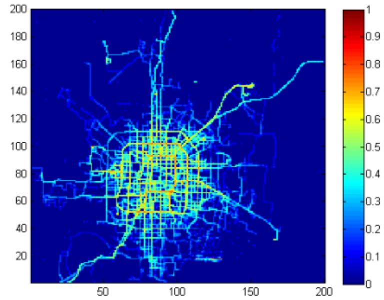


FIGURE 7. Visualization of T-Drive trajectory dataset.

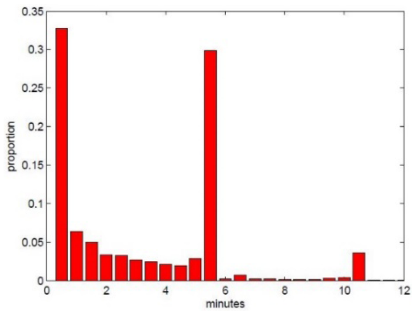


FIGURE 8. The distribution of sampling time interval [46], [47].

have enough information to observe the continuous trajectories of taxis.

Besides that, we find there exist some invalid waypoints in dataset, they deviate from the geographical position of the last valid waypoints in very short time interval. For example, in Figure 9, invalid waypoints are marked by red colors. These invalid waypoints usually occur when taxis are in the areas where GPS service quality is poor, e.g. tunnels. To guarantee the accuracy of experimental results, we need to filter out these invalid waypoints. Below is the simple filtering algorithm we propose:

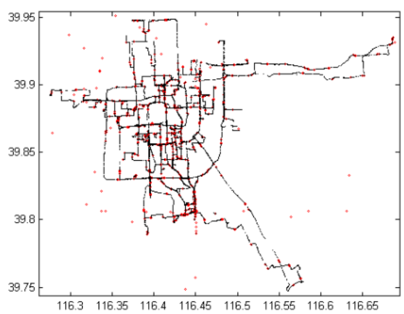


FIGURE 9. Invalid waypoints in dataset.

For a given waypoint, we check the distance between the last valid waypoint and this waypoint divides the time interval between them, if it is larger than the maximum speed limit in Beijing, which is 80 km/h in most places, we mark this waypoint as invalid waypoint and filter it out. Figure 10 illustrates the effectiveness of this filtering method.

FIGURE 10. Dataset after filtering out invalid waypoints.

B. EXPERIMENTAL METHODOLOGY

As SA-PA algorithm need repeatedly simulate the running process of training set with respect to different priority assignment $\wp$ to compute $\mathcal{J}(\wp, [t_1, t_2))$, which is a extremely time-consuming process, we should simplify the running process to accelerate the algorithm:

We separate the map of Beijing by compact square grid cells, the area of each grid cell is about $40 \times 40m^2$, which is the typical coverage range of low power wireless technologies like 802.15.4 and Bluetooth in free space. We assume the communication range of every intelligent device, data center or the transceiver is the area covered by the grid cell it locates in. Although this approximation will introduce some computational error, it can significantly accelerate the simulation.

Since we have separated the map of Beijing by compact grid cells, we can also compress the T-Drive trajectory dataset to filter out unnecessary information based on the grid cells. Because we no longer concern about the exact geographical positions of taxis if we use grid cells as the substitution of original maps, Instead, what we only need to concern is which grid cells these taxis locate in. Therefore, we can compress the dataset and only retain the trajectory of grid cells rather than trajectory of exact longitudes and latitudes. By compressing the dataset, the amount of data needed to be processed significantly reduces, and thereby accelerate the simulation.

We then discuss the geographical deployment of intelligent devices and data centers since they determine two important elements of configuration $\mathbb{C} = (\mathcal{S}, \mathcal{P}, \mathcal{H}, \mathcal{V}, \mathcal{Q})$, i.e. $\mathcal{S}$ and $\mathcal{H}$, respectively.

For data centers $\mathcal{H}$, we follow the suggestion in section 4.3, that is, data centers are usually deployed in downtown areas of Smart City, i.e. hotspot areas. To find hotspot areas, we count the frequency of being travelled by taxis of each grid cell, select the grid cells with high frequency as hotspot areas and then deploy data centers to these grid cells.

By contrast, the deployment of intelligent devices $\mathcal{S}$ is comparatively free, we randomly select grid cells and deploy intelligent devices to these grid cells. These grid cells need to be travelled by taxis because taxis can usually cover most of places of the city except the non-constructive lands, such as rivers and forests. Non-constructive land usually have no infrastructures deployed. If a grid cell has no taxis passing by, it is likely to be non-constructive land. In addition, random selection can reduce bias since infrastructures of all areas have probability to be selected to deploy intelligent devices.

For the other three elements $\mathcal{V}$, $\mathcal{Q}$ and $\mathcal{P}$ of $\mathbb{C}$. $\mathcal{V}$ is determined by dataset. $\mathcal{Q}$ can be controlled by ourselves. Both $\mathcal{V}$ and $\mathcal{Q}$ are easy to be handled in experiments.

$\mathcal{P}$ need to be trained by SA-PA algorithm with respect to various optimization target function $\mathcal{J}$. Three concrete optimization target functions, which are introduced in section 3.2, are used in the experiments: $\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$.

Based on the research paradigm of machine learning, we separate T-Drive trajectory dataset into two parts: training set and test set. Training set is used to train the hypothesis, i.e. optimize $\mathcal{J}$ with respect to $\mathcal{P}$ to get a well-performed $\mathbb{C}$ (or more accurately, $\wp$), using SA-PA algorithm. Test set is used to report the performance of the trained hypothesis, i.e. the performance of $\mathbb{C}$. Because the T-Drive trajectory dataset provided by MSRA is a kind of spatial-temporal data, we separate it according to time: The trajectory data in period of Feb. 2 to Feb. 6 is training set and the trajectory data in period of Feb. 7 to Feb. 8 is test set.

C. PERFORMANCE ANALYSIS

1) PERFORMANCE WITH RESPECT TO $\mathcal{J}_{\mathcal{C}}$

As illustrated in section 6.2, three concrete optimization target functions are used to do experiments: $\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$. In this subsection, $\mathcal{J}_{\mathcal{C}}$ is used as optimization target function to illustrate the performance of SWDCP-SCmules scheme and SA-PA algorithm.

We first try to prove the effectiveness of SA-PA algorithm experimentally. To prove its effectiveness, we fix elements $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{Q}$ of configuration $\mathbb{C}$, and observe if the value of $\mathcal{J}_{\mathcal{C}_{[t_2,t_3)}}^{\mathbb{C}}$ increases with the increase of $\mathcal{J}_{\mathcal{C}_{[t_1,t_2)}}^{\mathbb{C}}$ where $[t_1, t_2)$ is the time interval of training set and $[t_2, t_3)$ is the time interval of test set. If $\mathcal{J}_{\mathcal{C}_{[t_1,t_2)}}^{\mathbb{C}}$ and $\mathcal{J}_{\mathcal{C}_{[t_2,t_3)}}^{\mathbb{C}}$ have a positive correlation, it proves the social-aware patterns in the past are similar to those in the future and these similarities lead to similar performance, which proves the basis of SA-PA algorithm.

In Figure 11, the blue line roughly tilts up with the increase of horizontal axis, which means that the priority assignment $\wp$ that can optimize $\mathcal{J}_{\mathcal{C}}(\mathcal{P}, [t_1, t_2))$ will also roughly optimize $\mathcal{J}_{\mathcal{C}}(\mathcal{P}, [t_2, t_3))$, i.e. $\mathcal{J}_{\mathcal{C}}(\mathcal{P}, [t_1, t_2))$ and $\mathcal{J}_{\mathcal{C}}(\mathcal{P}, [t_2, t_3))$ have a positive correlation.

Next we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{C}_{[t_2,t_3)}}^{\mathbb{C}}$ with respect to the increase of storage limit in test set.

In Figure 14, the black line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{random}$ is a randomly generated priority assignment. assignment. The red line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{1000}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{1000}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 1000$ based on $\wp_{random}$. From Figure 14, we can see that no matter which

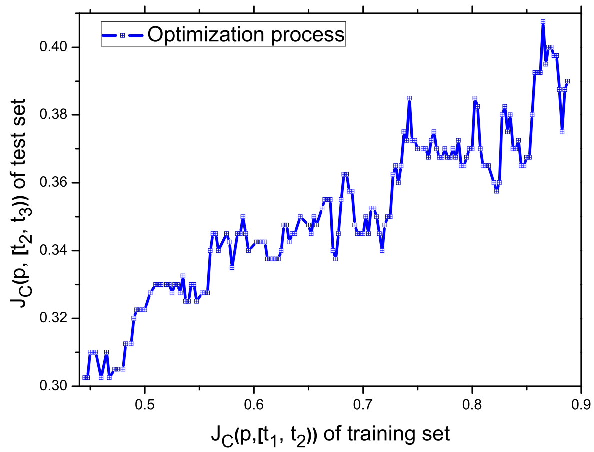


FIGURE 11. The optimization process of $\mathcal{J}_{\mathcal{C}}$.

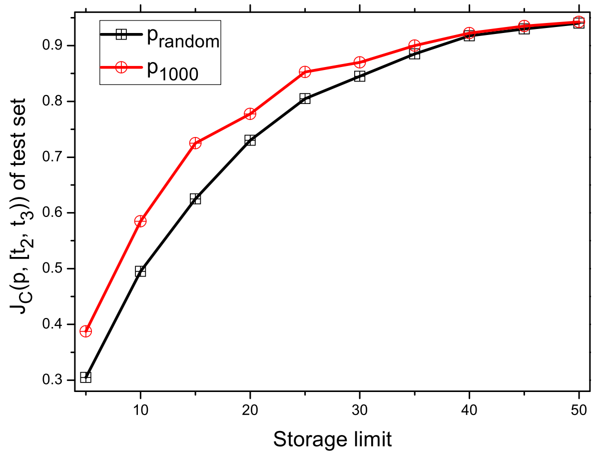


FIGURE 12. The change of $\mathcal{J}_{\mathcal{C}}$ with respect to storage limit.

priority assignment is used, the collection rate of SWDCP-SCmules scheme increases with the increase of storage limit, which conforms to intuition. Besides that, we can also see that $\wp_{1000}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the collection rate of red line is higher than the collection rate of black line, line, especially when the storage limit is not large. This proves the effectiveness of SA-PA algorithm. However, when storage limit is large, the effectiveness of SA-PA algorithm is not significant, this is because the storage is large enough to get rid of the limitation of storage.

Then we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{C}_{[t_2,t_3)}}^{\mathbb{C}}$ with respect to the increase of time in test set.

In Figure 13, the black line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of time where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{200}, [t_2, t_3))$ with respect to the the increase of time where $\wp_{200}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 200$ based on $\wp_{random}$. From Figure 13, we can see that no matter which priority assignment is used, the

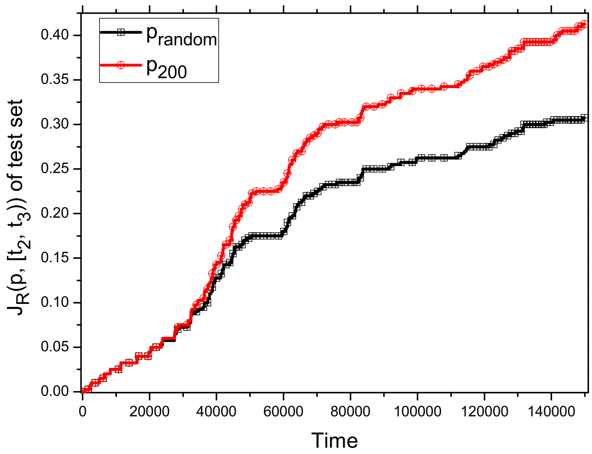


FIGURE 13. The change of $\mathcal{J}_{\mathcal{C}}$ with respect to time.

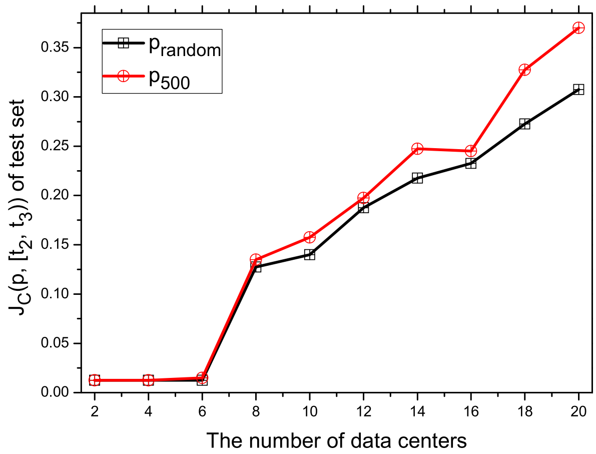


FIGURE 14. The change of $\mathcal{J}_{\mathcal{C}}$ with respect to the number of data centers.

of SWDCP-SCmules scheme increases with the increase of time, which conforms to intuition. Besides that, we can also see we can also see that $\wp_{200}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the collection rate of $\wp_{200}$ is larger than the collection rate of $\wp_{random}$ in every time. This proves the algorithm.

Finally, we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{C}}{}^{\mathbb{C}}_{[t_2,t_3)}$ with respect to the increase of data centers deployed in Smart City in test set.

In Figure 12, the black line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of the number of data centers where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the increase of $\mathcal{J}_{\mathcal{C}}(\wp_{500}, [t_2, t_3))$ with respect to the increase of the number of data centers where $\wp_{500}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 500$ based on $\wp_{random}$. From Figure 12, we can see that no matter which priority assignment is used, the collection rate of SWDCP-SCmules scheme increases with the increase of the number of data centers, which conforms to intuition. Besides that, we can also see that $\wp_{500}$ outperforms $\wp_{random}$ since the red line is above the black line, which means means the collection rate of $\wp_{500}$ is larger than the collection rate of $\wp_{random}$ no matter how many data deployed. This proves the effectiveness of SA-PA algorithm.

2) PERFORMANCE WITH RESPECT TO $\mathcal{J}_{\overline{\mathcal{R}}}$

As illustrated in section 6.2, three concrete optimization target functions are used to do experiments: $\mathcal{J}_{\mathcal{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathcal{S}}$. In this subsection, $\mathcal{J}_{\overline{\mathcal{R}}}$ is used as optimization target function to illustrate the performance of SWDCP-Scmules scheme and SA-PA algorithm.

We first try to prove the effectiveness of SA-PA algorithm experimentally. To prove its effectiveness, we fix elements $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{Q}$ of configuration $\mathbb{C}$, and observe if the value of $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_2,t_3)}$ increases with the increase of $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_1,t_2)}$ where $[t_1, t_2)$ is the time interval of training set and $[t_2, t_3)$ is the time interval of test set. If $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_1,t_2)}$ and $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_2,t_3)}$ have a positive correlation, it proves the social-aware patterns in the past are similar to those in the future and these similarities lead to similar performance, which proves the basis of SA-PA algorithm.

In Figure 15, the blue line roughly tilts up with the increase of horizontal axis, which means that the priority assignment $\wp$ that can optimize $\mathcal{J}_{\overline{\mathcal{R}}}(\mathcal{P}, [t_1, t_2))$ will also roughly optimize $\mathcal{J}_{\overline{\mathcal{R}}}(\mathcal{P}, [t_2, t_3))$, i.e. $\mathcal{J}_{\overline{\mathcal{R}}}(\mathcal{P}, [t_1, t_2))$ and $\mathcal{J}_{\overline{\mathcal{R}}}(\mathcal{P}, [t_2, t_3))$ have a positive correlation.

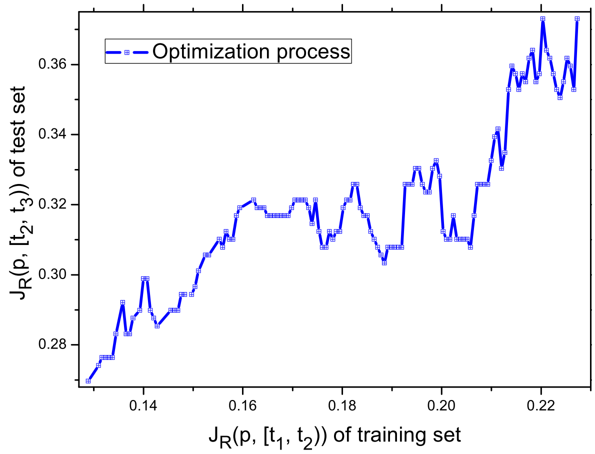


FIGURE 15. The optimization process of $\mathcal{J}_{\overline{\mathcal{R}}}$.

Next we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_2,t_3)}$ with respect to the increase of storage limit in test set.

In Figure 16, the black line illustrates the change of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the change of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{1000}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{1000}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 1000$ based on $\wp_{random}$.

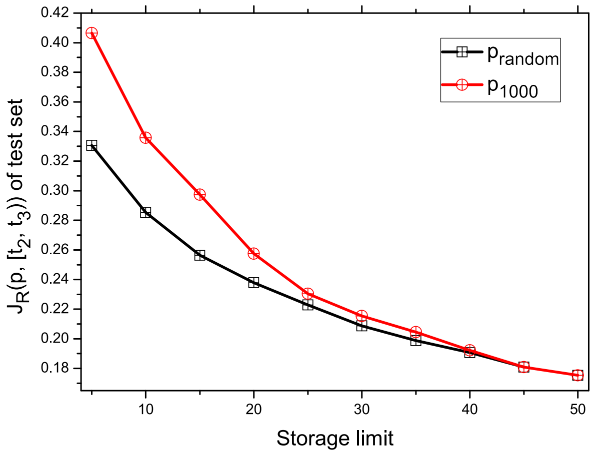


FIGURE 16. The change of $J_{\overline{\mathcal{R}}}$ with respect to storage limit.

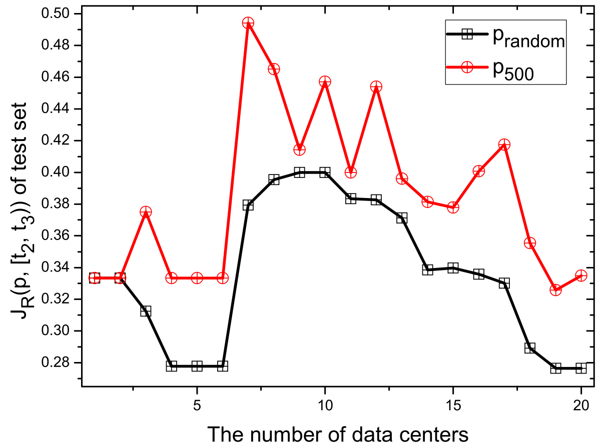


FIGURE 18. The change of $\mathcal{J}_{\overline{\mathcal{R}}}$ with respect to the number of data centers.

From Figure 16, we can see that no matter priority assignment is used, the effective collection rate of SWDCP-SCmules scheme decreases with the increase of storage limit, which is because the more storage SCmules have, the larger probability they will carry redundant data/messages. Besides that, we can also see that $\wp_{1000}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the effective collection rate of red line is higher than the effective collection rate of black line, especially when the storage limit is not large. This proves the effectiveness of SA-PA algorithm. However, when storage limit is large, the effectiveness of optimization is not significant, this is because the storage is large enough to get rid of the limitation of storage.

Then we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}^{\mathbb{C}}_{\overline{\mathcal{R}}[t_2,t_3)}$ with respect to the increase of time in test set.

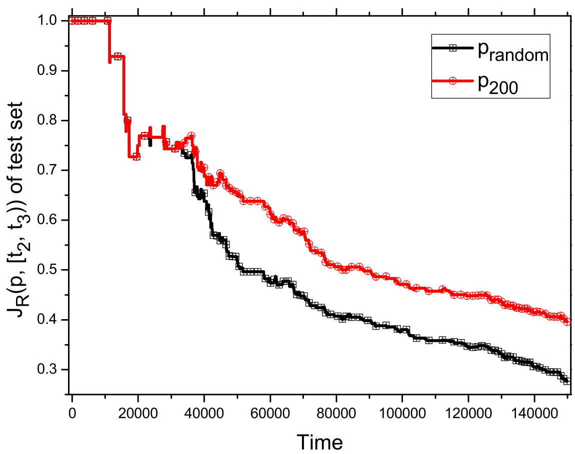


FIGURE 17. The change of $\mathcal{J}_{\overline{\mathcal{R}}}$ with respect to time.

In Figure 17, the black line illustrates the change of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of time where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the change of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{200}, [t_2, t_3))$ with respect to the increase of time where $\wp_{200}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 200$ based on $\wp_{random}$. From Figure 17, we can see that no matter which priority assignment is used, the effective collection rate of SWDCP-SCmules scheme roughly decreases with the increase of time. Besides that, we can also see that $\wp_{200}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the effective collection rate of $\wp_{200}$ is larger than the effective collection rate of $\wp_{random}$ in every time. This proves the effectiveness of SA-PA algorithm.

Finally, we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}^{\mathbb{C}}_{\overline{\mathcal{R}}[t_2,t_3)}$ with respect to the increase of data centers deployed in Smart City in test set.

In Figure 18, the black line illustrates the increase of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of the number of data centers where $\wp_{random}$ is a randomly priority assignment. The red line illustrates the increase of $\mathcal{J}_{\overline{\mathcal{R}}}(\wp_{500}, [t_2, t_3))$ with respect to the increase of the number of data centers where $\wp_{500}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 500$ based on $\wp_{random}$. From Figure 18, we can see that the effective collection rate fluctuates with the increase of the number of data centers, this is because the effective collection rate really depends on the selection of positions to deploy data centers. Besides that, we can also see that $\wp_{500}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the effective collection rate of $\wp_{500}$ is larger than the effective collection rate of $\wp_{random}$ no matter how many number of data centers are deployed. This proves the effectiveness of SA-PA algorithm.

3) PERFORMANCE WITH RESPECT TO $\mathcal{J}_{\mathbb{S}}$

As illustrated in section 6.2, three concrete optimization target functions are used to do experiments: $\mathcal{J}_{\mathbb{C}}$, $\mathcal{J}_{\overline{\mathcal{R}}}$ and $\mathcal{J}_{\mathbb{S}}$. In this subsection, $\mathcal{J}_{\mathbb{S}}$ is used as optimization target function to illustrate the performance of SWDCP-SCmules scheme

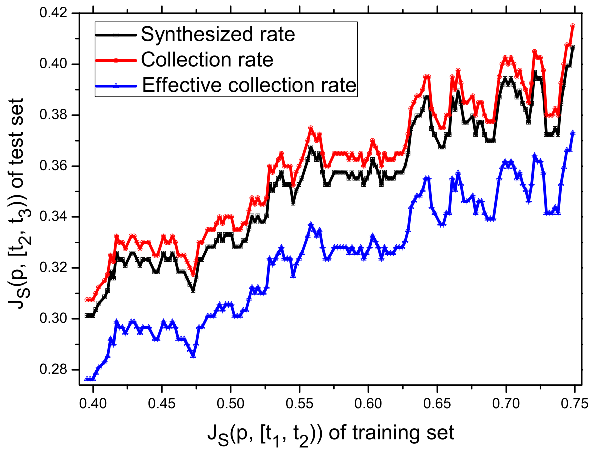


FIGURE 19. **The optimization process of $\mathcal{J}_{\mathcal{S}}$.**

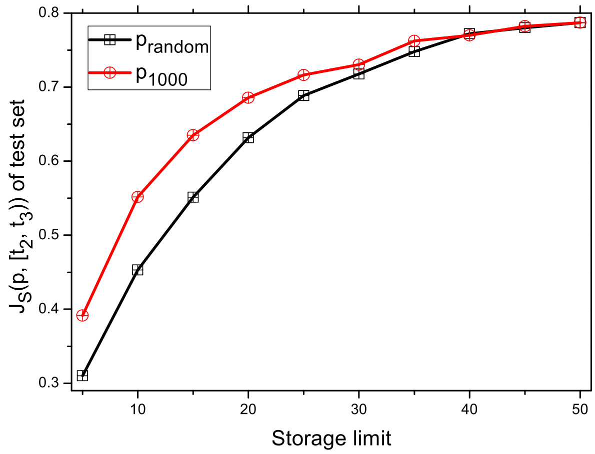


FIGURE 20. **The change of $\mathcal{J}_{\mathcal{S}}$ with respect to storage limit.**

and SA-PA algorithm. The concrete $\mathcal{J}_{\mathcal{S}}$ we use in this subsection is

$$\mathcal{J}_{\mathcal{S}_{\mathcal{T}}}^{\mathbb{C}} = 0.8\mathcal{C}_{\mathcal{T}}^{\mathbb{C}} + 0.2\left(1 - \mathcal{R}_{\mathcal{T}}^{\mathbb{C}}\right)$$

where $\lambda_1 = 0.8$, $\lambda_2 = 0.2$ and T is a time interval.

We first try to prove the effectiveness of SA-PA algorithm experimentally. To prove its effectiveness, we fix elements $\mathcal{S}$, $\mathcal{H}$, $\mathcal{V}$ and $\mathcal{Q}$ of configuration $\mathbb{C}$, and observe if the value of $\mathcal{J}_{\mathcal{S}_{[t_2,t_3)}}^{\mathbb{C}}$ increases with the increase of $\mathcal{J}_{\mathcal{S}_{[t_1,t_2)}}^{\mathbb{C}}$ where $[t_1, t_2)$ is the time interval of training set and $[t_2, t_3)$ is the time interval of test set. If $\mathcal{J}_{\mathcal{S}_{[t_1,t_2)}}^{\mathbb{C}}$ and $\mathcal{J}_{\mathcal{S}_{[t_2,t_3)}}^{\mathbb{C}}$ have a positive correlation, it proves the social-aware patterns in the past are similar to those in the future and these similarities lead to similar performance, which proves the basis of SA-PA algorithm.

In Figure 19, the black line roughly tilts up with the increase of horizontal axis, which means that the priority assignment $\wp$ that can optimize $\mathcal{J}_{\mathcal{S}}(\mathcal{P}, [t_1, t_2))$ will also roughly optimize $\mathcal{J}_{\mathcal{S}}(\mathcal{P}, [t_2, t_3))$, i.e. $\mathcal{J}_{\mathcal{S}}(\mathcal{P}, [t_1, t_2))$ and $\mathcal{J}_{\mathcal{S}}(\mathcal{P}, [t_2, t_3))$ have a positive correlation. Besides the black line, red line and blue line illustrates the changes of collection rate and effective collection rate in the process of optimization, respectively.

Next we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{S}_{[t_2,t_3)}}^{\mathbb{C}}$ with respect to the increase of storage limit in test set.

In Figure 20, the black line illustrates the increase of $\mathcal{J}_{\mathcal{S}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{random}$ is a randomly generated priority assignment. assignment. The red line illustrates the increase of $\mathcal{J}_{\mathcal{S}}(\wp_{1000}, [t_2, t_3))$ with respect to the increase of storage limit where $\wp_{1000}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 1000$ based on $\wp_{random}$. From Figure 20, we can see that no matter which priority assignment is used, the synthesized rate of SWDCP-SCmules scheme roughly increases with the increase of storage limit, which is because the weight we set to collection rate is much larger than the weight we set to redundancy rate, i.e. $\lambda_1 > \lambda_2$, in synthesized rate. Besides that, we can also see that $\wp_{1000}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the synthesized rate of red line is rate of black line, especially when the storage limit is not large. This proves the effectiveness of SA-PA algorithm. However, when storage limit is large, the effectiveness of optimization is not significant, this is because the storage is large enough to get rid of the limitation of storage.

Then we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{S}_{[t_2,t_3)}}^{\mathbb{C}}$ with respect to the increase of time in test set.

In Figure 21, the black line illustrates the increase of $\mathcal{J}_{\mathcal{S}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of time where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the increase of $\mathcal{J}_{\mathcal{S}}(\wp_{200}, [t_2, t_3))$ with respect to the the increase of time where $\wp_{200}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 200$ based on $\wp_{random}$. From Figure 21, we can see that no matter which priority assignment is used, the rate of SWDCP-SCmules roughly increase with the increase of time, which is because the weight we set to collection rate is much larger than the weight we set to redundancy rate, i.e. $\lambda_1 > \lambda_2$, in synthesized rate. Besides synthesized rate. Besides that, we can also see that $\wp_{200}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the synthesized rate of $\wp_{200}$ is larger than the synthesized rate of $\wp_{random}$ in every time. This proves the effectiveness of SA-PA algorithm.

Finally we closely compare the degree of improvement provided by SA-PA algorithm from the change of $\mathcal{J}_{\mathcal{S}_{[t_2,t_3)}}^{\mathbb{C}}$ with respect to the increase of the number of data centers in test set.

In Figure 22, the black line illustrates the increase of $\mathcal{J}_{\mathcal{S}}(\wp_{random}, [t_2, t_3))$ with respect to the increase of the number of data centers where $\wp_{random}$ is a randomly generated priority assignment. The red line illustrates the increase of

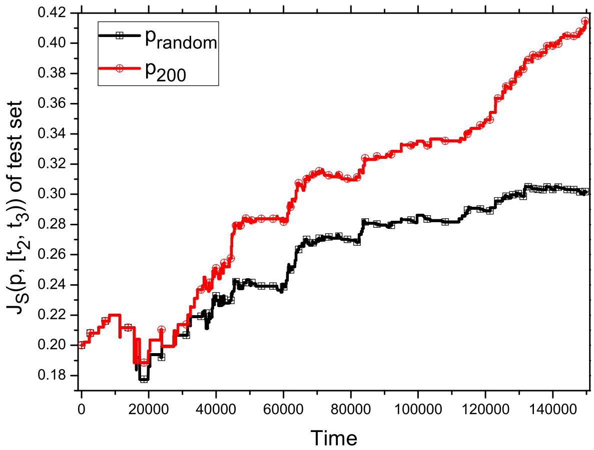


FIGURE 21. The change of $\mathcal{J}_{\mathcal{S}}$ with respect to time.

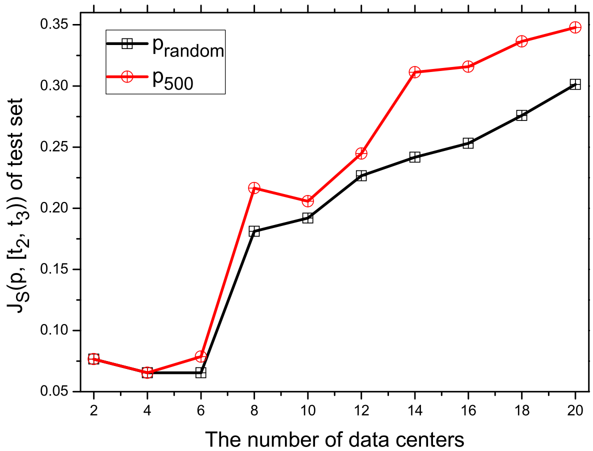


FIGURE 22. The change of $\mathcal{J}_{\mathcal{S}}$ with respect to the number of data centers.

$\mathcal{J}_{\mathcal{S}}(\wp_{500}, [t_2, t_3])$ with respect to the increase of the number of data centers where $\wp_{500}$ is the priority assignment generated by a running of SA-PA algorithm with $nbMV = 500$ based on $\wp_{random}$. From Figure 22, we can see that no matter which priority assignment is used, the synthesized rate of SWDCP-SCmules roughly increase with the increase of the number of data centers, which is because the weight we set to collection rate is much larger than the weight we set to redundancy rate, i.e. $\lambda_1 > \lambda_2$, in synthesized rate. Besides that, we can also see that $\wp_{500}$ outperforms $\wp_{random}$ since the red line is above the black line, which means the synthesized rate of $\wp_{500}$ is larger than the synthesized rate of $\wp_{random}$ no matter how many data centers are deployed. This proves the effectiveness of SA-PA algorithm.

4) SCmules VERSUS MULES

In this subsection, we analyze the influence of storage-constraint, in other words, we compare the performance of SCmules introduced in this paper with the performance of mules (i.e. oblivious data mules) proposed by Bonola *et al.* [17].

TABLE 2. Main notations.

	SCmules	Mules
$\mathcal{J}_{\mathcal{S}}$	82.0353	81.0222
$\mathcal{J}_{\mathcal{C}}$	98.2500	98.2500
$\mathcal{J}_{\overline{\mathcal{R}}}$	17.1766	12.1109

We use SCmules with a priority assignment optimized by SA-PA algorithm with respect to the optimization target function $\mathcal{J}_{\mathcal{S}}$, where the size of storage of SCmules is 20, to compare with mules, which has no storage-constraint. Table 2 gives the value of $\mathcal{J}_{\mathcal{S}}{}^{\mathbb{C}}_{[t_2,t_3)}$, $\mathcal{J}_{\mathcal{C}}{}^{\mathbb{C}}_{[t_2,t_3)}$ and $\mathcal{J}_{\overline{\mathcal{R}}}{}^{\mathbb{C}}_{[t_2,t_3)}$ using SCmules and mules. From this table, we can draw the conclusion that, although the performance, especially the collection rate, can be influenced by whether the oblivious data mules are storage-constrained or not, the bad influence can be significantly mitigated by using a proper priority assignment found by SA-PA algorithm. In addition, the proper priority assignment found by SA-PA algorithm can reduce the redundancy rate and therefore reducing energetic waste and increasing network lifetime.

VII. CONCLUSIONS

In this paper, we propose a Social Welfare Data Collection paradigm based on Storage-Constrained Oblivious Data Mules (SWDCP-SCmules scheme). In this scheme, intelligent devices are embedded to infrastructures of Smart City to detect and report their status, data centers are deployed in hotspot areas of Smart City to collect data from intelligent devices and SCmules are mobile IoT nodes moving in Smart City to obliviously pick up data reported by intelligent devices and store-carry-forward to data centers. The SWDCP-SCmules scheme enables the intelligent management of cities and boost the overall social welfare.

However, the storage size of SCmules are limited, to cope with the storage limitations, the concept of priority is introduced to model the selection strategy used in the situation that SCmules lack of spare storage and converts selection strategy to priority assignment. The Simulated Annealing for Priority Assignment Algorithm (SA-PA algorithm) is proposed to guide the priority assignment of intelligent devices. The SA-PA algorithm is a universal machine learning algorithm which can automatically find well-performed priority assignment with respect to various optimization targets by learning social-aware patterns from the past GPS trajectory data of SCmules and significantly improve the performance of SWDCP-SCmules scheme.

In general, the SWDCP-SCmules combined with SA-PA algorithm can propel the construction of Smart City and thereby lead to significant improvement of overall social welfare.

REFERENCES

[1] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, "Cluster content caching: An energy-efficient approach to improve quality of service in cloud radio access networks," *IEEE J. Sel. Areas Commun.*, vol. 34, no. 5, pp. 1207–1221, May 2016.

[2] S. He, S. He, D.-H. Shin, J. Zhang, J. Chen, and Y. Sun, "Full-view area coverage in camera sensor networks: Dimension reduction and near-optimal solutions," *IEEE Trans. Veh. Technol.*, vol. 65, no. 9, pp. 7448–7461, Sep. 2016, doi: 10.1109/TVT.2015.2498281.

[3] M. Peng, H. Xiang, Y. Cheng, S. Yan, and H. V. Poor, "Inter-tier interference suppression in heterogeneous cloud radio access networks," *IEEE Access*, vol. 3, pp. 2441–2455, 2015.

[4] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 8, pp. 2053–2064, Aug. 2014.

[5] S. He, J. Chen, X. Li, X. Shen, and Y. Sun, "Mobility and intruder prior information improving the barrier coverage of sparse sensor networks," *IEEE Trans. Mobile Comput.*, vol. 13, no. 6, pp. 1268–1282, Jun. 2015.

[6] X. Liu, "A deployment strategy for multiple types of requirements in wireless sensor networks," *IEEE Trans. Cybern.*, vol. 45, no. 10, pp. 2364–2376, Oct. 2015.

[7] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, "Energy-efficient probabilistic area coverage in wireless sensor," *IEEE Trans. Veh. Technol.*, vol. 61, no. 1, pp. 367–377, Jan. 2015.

[8] Q. Hu, M. Peng, Z. Mao, X. Xie, and H. V. Poor, "Training design for channel estimation in uplink cloud radio access networks," *IEEE Trans. Signal Process.*, vol. 64, no. 13, pp. 3324–3337, Jul. 2016.

[9] H. Li, Y. Yang, T. H. Luan, X. Liang, L. Zhou, and X. S. Shen, "Enabling fine-grained multi-keyword search supporting classified sub-dictionaries over encrypted cloud data," *IEEE Trans. Dependable Secure Comput.*, vol. 13, no. 3, pp. 312–325, May/Jun. 2016.

[10] S. Sarkar, S. Chatterjee, and S. Misra, "Assessment of the suitability of fog computing in the context of Internet of things," *IEEE Trans. Cloud Comput.*, to be published, doi: 10.1109/TCC.2015.2485206.2015.

[11] M. Aazam and E.-N. Huh, "Fog computing micro datacenter based dynamic resource estimation and pricing model for IoT," in *Proc. IEEE 29th Int. Conf. Adv. Inf. Netw. Appl.*, Mar. 2015, pp. 687–694.

[12] L. Piras. (Mar. 2014). *A Brief History of the Internet of Things [Infographic]*. [Online]. Available: http://www.psfk.com/2014/03/internet-ofthings-infographic.html

[13] *Internet of Things Market Forecast Cisco*. [Online]. Available: http://postscapes.com/internet-of-things-market-size

[14] X. Liu, "A novel transmission range adjustment strategy for energy hole avoiding in wireless sensor networks," *J. Netw. Comput. Appl.*, vol. 67, pp. 43–52, May 2016.

[15] Y. Hu and A. Liu, "Improving the quality of mobile target detection through portion of node with full duty cycle in WSNs," *Comput. Syst. Sci. Eng.*, vol. 31, no. 1, pp. 5–17, 2016.

[16] Y. Liu *et al.*, "FFSC: An energy efficiency communications approach for delay minimizing in Internet of Things," *IEEE Access*, vol. 4, pp. 3775–3793, 2016.

[17] M. Bonola, L. Bracciale, P. Loreti, P. Amici, A. Rabuffi, and G. Bianchi, "Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers," *Ad Hoc Netw.*, vol. 43, pp. 43–55, Jun. 2016.

[18] N. Luong, D. Hoang, P. Wang, D. Niyato, D. Kim, and Z. Han, "Data collection and wireless communication in Internet of Things (IoT) using economic analysis and pricing models: A survey," *IEEE Commun. Surveys Tut.*, to be published, doi: 10.1109/COMST.2016.2582841,2016.

[19] R. C. Shah, S. Roy, S. Jain, W. Brunette, "Data MULEs: Modeling a three-tier architecture for sparse sensor networks," *Ad Hoc Netw.*, vol. 1, nos. 2–3, pp. 215–233, Sep. 2003.

[20] R. Xie, A. Liu, and J. Gao, "A residual energy aware schedule scheme for wsns employing adjustable awake/sleep duty cycle," *Wireless Pers. Commun.*, vol. 2016, pp. 1–29, Jun. 2016, doi: 10.1007/s11277-016-3428-0.2016.

[21] H. Li, D. Liu, Y. Dai, and T. H. Luan, "Engineering searchable encryption of mobile cloud networks: When QoE meets QoP," *IEEE Wireless Commun.*, vol. 22, no. 4, pp. 74–80, Aug. 2015.

[22] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of Things (IoT): A vision, architectural elements, and future directions," Tech. Rep. CLOUDS-TR-2012-2, Jul. 2012.

[23] C. C. Sobin, V. Raychoudhury, G. Marfia, and A. Singla, "A survey of routing and data dissemination in delay tolerant networks," *J. Netw. Comput. Appl.*, vol. 67, pp. 128–146, May 2016.

[24] M. Raj, N. Li, D. Liu, M. Wright, S. K. Das , "Using data mules to preserve source location privacy in wireless sensor networks," *Pervasive Mobile Comput.*, vol. 11, pp. 244–260, Apr. 2014.

[25] Y.-C. Tseng, F.-J. Wu, W.-T. Lai, "Opportunistic data collection for disconnected wireless sensor networks by mobile mules," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1150–1164, May 2013.

[26] C.-K. Tham and T. Luo, "Fairness and social welfare in service allocation schemes for participatory sensing," *Comput. Netw.*, vol. 73, pp. 58–71, Nov. 2014.

[27] A. Liu, Y. Hu, and Z. Chen, "An energy-efficient mobile target detection scheme with adjustable duty cycles in wireless sensor networks," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 22, no. 4, pp. 203–225, 2016.

[28] Y. Liu, M. Dong, K. Ota, and A. Liu, "ActiveTrust: Secure and trustable routing in wireless sensor networks," *IEEE Trans. Inf. Forensics Secur.*, vol. 11, no. 9, pp. 2013–2027, Sep. 2016.

[29] C.-K. Tham and T. Luo, "Quality of contributed service and market equilibrium for participatory sensing," *IEEE Trans. Mobile Comput.*, vol. 14, no. 4, pp. 829–842, Apr. 2015.

[30] A. Liu, X. Liu, and Y. Liu, "A comprehensive analysis for fair probability marking based traceback approach in WSNs," *Secur. Commun. Netw.*, vol. 9, no. 14, pp. 2448–2475, Sep. 2016.

[31] M. Dong, K. Ota, and A. Liu, "RMER: Reliable and energy-efficient data collection for large-scale wireless sensor networks," *IEEE Internet Things J.*, vol. 3, no. 4, pp. 511–519, Aug. 2016.

[32] J. Long, C. Gao, S. He, X. Liu, and A. Liu, "Bridging the gap among actor–sensor–actor communication through load balancing multi-path routing," *EURASIP J. Wireless Commun. Netw.*, vol. 2015, p. 256, Dec. 2015, doi: 10.1186/s13638-015-0484-1.

[33] Z. Song, C. H. Liu, J. Wu, J. Ma, and W. Wang, "Qoi-aware multitask-oriented dynamic participant selection with budget constraints," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4618–4632, Nov. 2014.

[34] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (VANETs): Challenges and perspectives," in *Proc. 6th Int. Conf. ITS Telecommun.*, Jul. 2006, pp. 761–766, doi: 10.1109/ITST.2006.289012.

[35] Y. Saeed, SA. Lodhi, K. Ahmed, "Obstacle management in VANET using game theory and fuzzy logic control," *Int. J. Commun.*, vol. 4, no. 1, pp. 9–13, Jan. 2013.

[36] M. T. Mitchell, *Machine Learning*. vol. 45. Burr Ridge, IL, USA: McGraw Hill, 1997, p. 37.

[37] C. Bishop, *Pattern Recognition and Machine Learning* (Information Science and Statistics), 2nd ed. 2007.

[38] I. Boussaïd, J. Lepagnot, and P. Siarry, "A survey on optimization metaheuristics," *Inf. Sci.*, vol. 237, pp. 82–117, Jul. 2013.

[39] B. Suman and P. Kumar, "A survey of simulated annealing as a tool for single and multiobjective optimization," *J. Oper. Res. Soc.*, vol. 57, no. 10, pp. 1143–1160, Oct. 2006.

[40] T. Tanaka, T. Suzuki, and K. Kurihara, "Energy harvesting technology for maintenance-free sensors," *Fujitsu Sci. Technol. J.*, vol. 50, pp. 93–100, Jan. 2014.

[41] D. G. Leeper, "A long-term view of short-range wireless," *Computer*, vol. 34, no. 6, pp. 39–44, Jun. 2001.

[42] J. M. Rabaey, M. J. Ammer, J. L. da Silva, D. Patel, and S. Roundy, "Picoradio supports ad hoc ultra-low power wireless networking," *Computer*, vol. 33, no. 7, pp. 42–48, Jul. 2000.

[43] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient MAC protocol for wireless sensor networks," in *Proc. INFOCOM*, vol. 3. Jun. 2002, pp. 1567–1576.

[44] C. Wang, D. Mu, F. Zhao, J. W. Sutherland, "A parallel simulated annealing method for the vehicle routing problem with simultaneous pickup–delivery and time windows," *Comput. Ind. Eng.*, vol. 83, pp. 111–122, May 2015.

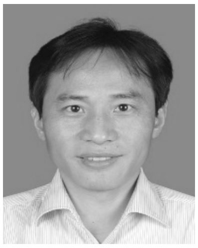
[45] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[46] S. He, J. Chen, F. Jiang, D. K. Y. Yau, G. Xing, and Y. Sun, "Energy provisioning in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 12, no. 10, pp. 1931–1942, Oct. 2013.

[47] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proc. 17th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining KDD*, New York, NY, USA, 2011, pp. 316–324.

ZHIPENG TANG is currently pursuing the degree with the School of Information Science and Engineering, Central South University, China. His research interests include services-based network, crowd sensing networks, and wireless sensor networks.

ANFENG LIU received the M.Sc. and Ph.D. degrees in computer science from Central South University, China, in 2002 and 2005, respectively. He is currently a Professor with the School of Information Science and Engineering, Central South University, China. He is also a member (E200012141M) of China Computer Federation. His major research interest is wireless sensor network.



CHANGQIN HUANG received the Ph.D. degree in computer science and technology from Zhejiang University, Hangzhou, in 2005. He is a currently a Professor with the School of Information Technology in Education, South China Normal University, China. He is also a Guangdong Specially Appointed Professor (Pearl River Scholar), and a Senior Member (E200014100S) of China Computer Federation. He has authored over 80 research papers in international journals and conferences. His research interests include service computing, cloud computing, semantic web, and education informationization.