User Association for Load Balancing in Vehicular Networks: An Online Reinforcement Learning Approach

Zhong Li, Cheng Wang, and Chang-Jun Jiang

***Abstract*— Recently, a number of technologies have been developed to promote vehicular networks. When vehicles are associated with the heterogeneous base stations (e.g., macrocells, picocells, and femtocells), one of the most important problems is to make load balancing among these base stations. Different from common mobile networks, data traffic in vehicular networks can be observed having regularities in the spatial–temporal dimension due to the periodicity of urban traffic flow. By taking advantage of this feature, we propose an online reinforcement learning approach, called ORLA. It is a distributed user association algorithm for network load balancing in vehicular networks. Based on the historical association experiences, ORLA can obtain a good association solution through learning from the dynamic vehicular environment continually. In the long run, the real-time feedback and the regular traffic association patterns both help ORLA cope with the dynamics of network well. In experiments, we use QiangSheng taxi movement to evaluate the performance of ORLA. Our experiments verify that ORLA has higher quality load balancing compared with other popular association methods.**

***Index Terms*— User association, online reinforcement learning, load balancing, vehicular networks.**

I. Introduction

AS THE development of vehicular networks, more and more vehicles need to associate with the heterogeneous base stations (different ties of transmit powers, physical sizes and costs) in vehicular networks [1]. In a city, there are great differences among these requirements. In the dense traffic area, the association requirements are more than that in the sparse traffic area. Under the traditional max-SINR scheme, a powerful/strong base station may attract more vehicles to associate with it, as illustrated in Fig.1. Even with a targeted deployment where the weak base stations are placed in the dense traffic areas, most vehicles still receive the powerful downlink signal from the strong base stations. This will result in the strong ones having heavy loads while the weak ones having many idle resources. For vehicles, even they associate with the strong base stations, the service rates are still very bad, since the strong ones serve too many vehicles. So, a more balanced association scheme is needed for vehicular networks.

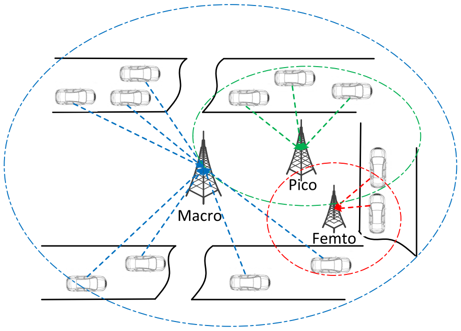


Fig. 1. Illustration of vehicular association with heterogeneous base stations under max-SINR scheme.

Unfortunately, most popular optimization technologies, like gradient descent method, Lagrange multiplier method, only apply to the scenario where the traffic flow generated by mobile users is approximately static. They assume the channel quality is stable. However, in real world, the change of the traffic is not stable. The assumption results in the invalidation of association solution. Even if we apply them in the unknown dynamic environment, the lack of feedback signals from environment causes the gradient descent losing its direction. Besides, once the network scenario changes, the traditional association algorithms must rerun in the whole network with high costs. Fortunately, we find that in vehicular networks, there exists potential regularities of spatial-temporal distribution for traffic flows every day. The goal of our study is to learn and utilize the spatial-temporal association experiences so as to directly obtain the association solution in the dynamic vehicular environment.

To this end, we introduce the reinforcement learning method into this work. *Reinforcement learning* is learning what to do or how to map situations to actions so as to maximize a

Manuscript received January 12, 2017; revised May 21, 2017; accepted May 24, 2017. Date of publication June 16, 2017; date of current version July 31, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61602110, in part by the Sailing Program under Grant 16YF1400200, in part by the Shanghai Science and Technology Innovation Action Plan Project under Grant 16511100900, in part by the National Science Foundation for Postdoctoral Scientists of China under Grant 2016M591575, in part by the National Natural Science Foundation of China under Grant 61571331, in part by the Fok Ying-Tong Education Foundation for Young Teachers in the Higher Education Institutions of China under Grant 151066, in part by the Shuguang Program from Shanghai Education Development Foundation under Grant 14SG20, and in part by Fundamental Research Funds for the Central Universities under Grant 2232016D-30. The Associate Editor for this paper was D.-H. Lee. *(Corresponding author: Cheng Wang.)*

Z. Li is with the College of Information Science and Technology, Donghua University, Shanghai 201620, China (e-mail: 007lizhong@gmail.com).

C. Wang and C.-J. Jiang are with the Department of Computer Science, Tongji University, Shanghai 201804, China, also with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 10014176, China, and also with the Shanghai Electronic Transactions and Information Service Collaborative Innovation Center, Shanghai 200000, China (e-mail: cwang@tongji.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2017.2709462

numerical reward signal [2]. Different from the supervised and unsupervised learning, trial-error search and delayed reward are the two most important features of reinforcement learning. Through interacting with the unknown environment continuously, an agent should know in what states what actions should be taken so as to make a right decision. In our problem, we face an unknown and dynamic vehicular environment. Because there are no labels that can be obtained in advance, our problem can not be formalized to an artificial neural network which is usually used to solve the classification and regression problems. The trial-error search and delayed labels are the features of our problem. Thus, we develop ORLA, an online reinforcement learning approach for user association in vehicular networks. The main idea is that through feeding back from the dynamic vehicular environment iteratively, ORLA obtains a near-optimal association solution based on historical association experiences. In the future, enlightened by literature [3], we can further consider the deep reinforcement learning to investigate our problem.

In the paper, there are two challenging problems when designing ORLA.

- How to use the reinforcement learning model to define an association problem in the dynamic environment?
- How to utilize the spatial-temporal regularities to design ORLA in vehicular networks?

For the first challenge, we transform the association problem into an 'N-armed bandit problem' [2]. We take advantage of a price-based idea to propose an *initial reinforcement learning method*. In the method, through feeding back from the current environment, we design a reward function that directs the change of price. The reward is defined as a deviation of all users' average service rates, which reflects the network load balance to some extent. Through learning, we can obtain the best association decision from the maximum long-term cumulative reward.

For the second challenge, we design a *historical-based reinforcement learning method*. After the initial reinforcement learning, each base station obtains its own historical association patterns, i.e., which users are associated with it. Since the traffic flow has the spatial-temporal regularities, there may exist similarity between the historical associations and the current case. Thus the historical association patterns can be utilized as references for the forthcoming traffic flows. The detailed definition about *association pattern*[1] is given in Section V-B. When the network keeps up changing, the base station uses a ε greedy method to learn the association actions based on its historical association patterns (decisions). In the historical-based reinforcement learning, ORLA uses the Pearson distance and Kullback-Leibler distance to calculate the similarity between the current case and the historical recorded pattern. The similarity helps the base station to choose the appropriate action and accelerate learning. After that, based on the difference of association allocations between the current requirement and the historical decision, ORLA proposes a *binary approaching method* and a *multi-spot diffusion method* to obtain the association decision for the current network. Finally, when the historical-based reinforcement learning ends, each base station records the current association solution again for accumulation.

In this work, ORLA is executed on each base station in a distributed way. Each station masters its current situation. The learning and decision are put on the base station side. The user side does not need to do any sophisticated computation. After the initial reinforcement learning ends, we let the historical-based reinforcement learning always stay to cope with the dynamic changes. Each base station uses its historical experiences to give an association decision. Meanwhile, through a reward from the current environment, ORLA adjusts this decision to ensure it is a good solution. Through learning from the historical patterns, ORLA avoids users trying blindly one by one in vehicular networks. In the long run, ORLA can finish the association tasks well and obtains good service rates for vehicles.

We test ORLA over the real world vehicular movement. We compare ORLA with the traditional max-SINR scheme and the popular 3D (Distributed Dual Decomposition Optimization) scheme. We use the variance and the overall service rates as the metrics of load balancing. From the experiments, first in ORLA, the number of associations with the femtocells is more, while that with the macrocells is less. It shows the effectiveness of ORLA in load balancing, intuitively. Second, we count the CDFs of the overall service rates. Compared with max-SINR and 3D scheme, ORLA has large overall service rates with higher proportion. Meanwhile, ORLA performs better than those comparison algorithms, with smaller variance of the service rates. Third, we valid the convergence time of ORLA. The results show that the worst time is below 1500ms. Besides, the convergence time of the historical phase decreases 76.9% largely compared with the initial phase of ORLA. It helps ORLA to deal with the dynamic environment better.

The rest of this paper is organized as follows. We review the related work in Section II and give the system model in Section III. In Section IV, we briefly introduce the preliminary knowledge about the reinforcement learning used in our design. We then highlight the design architecture and describe the ORLA scheme in Section V. We report the results from our extensive experimental evaluation in Section VI. Finally, we conclude the paper in Section VII.

II. Literature Review

Recently, the load balancing problem in cellular networks has been studied by utilizing various kinds of optimization techniques, such as dynamic programming, Markov decision processes and game theory. Some outstanding studies are done by Shen *et al.* [4], Boccardi *et al.* [5], Ye *et al.* [6], Andrews *et al.* [7], and Jo *et al.* [8]. These studies explore the network load balancing and resource allocation systematically from domains of cellular networks, OFDM systems and massive MIMO systems. Some eminent studies of wifi offloading are done by Cheng *et al.* [9], [10]. The research focuses on offloading the services from base stations to wifi access points. There are also some famous studies done by Yue *et al.* [11] and

[1] In the following, we often use the term 'pattern' and the term 'association pattern' alternatively, without confusion.

Han *et al.* [12] . The research considers the social properties and energy consumptions to obtain the network load balancing. The latest review can refer to literature [7], [9], [13].

In order to achieve load balancing, researchers usually transform the association problem to a convex optimization problem [14]. Then, the heuristic method [15], gradient projection and dual decomposition [6] can be utilized to solve the optimization problem. After obtaining the association solution, the studies enable the cell breathing to realize associations. By adjusting the transmit power, the cell breathing technique [16], [17] can dynamically change the coverage area depending on the load situation of the cells. In this course, studies often adopt the Poisson point process (PPP) to model the locations of users and base stations. The PPP model indeed simplifies the optimization analysis. However, in many scenarios, the homogeneous PPP model is not realistic, especially for the traffic flows in vehicular networks.

Some studies use the dynamic programming to solve the balancing problem. Like literature [18], it is novel to solve the energy balancing in vehicular networks by taking advantage of multihops between vehicles. The study specially considers the services under the delay tolerant environment.

Some studies [19], [20] exploit the Markov decision processes to study the association problem in discrete systems and stochastic systems. However, it is difficult to define reasonable state transition models and appropriate states since the complex environment is usually unknown to users.

Some studies [10], [21]–[23] adopt the game theory to solve the network selection problem. It requires the selection game to converge to Nash equilibrium. In literature [22], a group of players form a population. And then, players from one population choose strategies against users from other populations. This method neglects the interactions among users in a population group. However, in the real system, an individual user has a major impact on the performance of others.

In recent years, reinforcement learning is used to address the load balancing problem. In literature [24], a fuzzy rule-based Q-learning method is proposed to solve the enterprise LTE femtocells load balancing. In literature [13], [25], a monotone hysteretic policy and an improved Q-learning method are explored respectively to solve the energy efficiency balancing problem in heterogeneous cellular networks. These studies do not pay attention to the spatial-temporal distribution of traffic flows, either in homogeneous or heterogeneous cellular networks. A simple reinforcement learning method is not enough to cope with the complicated environment in vehicular networks of our study.

III. System Model and Assumptions

A. System Model

Recently a number of economical base stations (BS) have been deployed in the cellular network to meet the surging traffic demands. We consider a heterogeneous case with macrocells, picocells and femtocells coexisting in vehicular networks. The transmit powers of the three decrease in sequence with deploying picocells and femtocells denser than macrocells.

In this paper, we focus on downlink (DL) in the association scheme. We assume that all base stations have full buffers. In a real system, it is much more difficult to implement multi-BS association than single-BS association. Therefore, we consider the single-BS association, i.e., one user is exactly associated with one base station.

Let B and V denote the set of base stations and vehicles, respectively. During the connection period, we define the *achievable rate* as c_{ij}. Typically,

$$c_{ij} = \log_2(1 + SINR_{ij}) = \log_2(1 + \frac{P_j g_{ij}}{\sum_{k \in B, k \neq j} P_k g_{ik} + \sigma^2}), \tag{1}$$

where P_j denotes the transmit power of base station j, σ^2 denotes the noise power level, and g_{ij} denotes the channel gain between vehicle i and base station j, which includes antenna gain, path loss and shadowing. The sum $\sum_{k \in B, k \neq j} P_k g_{ik}$ represents the interferences coming from the heterogenous base stations.

B. The Measurement of Load Balancing

In this work, load balancing is a concept for describing the status of the whole network system, not for a single base station. Since each base station generally serves more than one vehicle, vehicles associated with the same base station need to share resources. Therefore, the key metric for performance is service rate, not SINR simply [6], [7]. The service rate experienced by a vehicle depends on the load of a base station, i.e., how the base station allocates its resources among its associated vehicles. We set the start time as t_0 and the current time as t. We use τ $(t_0 \leq \tau \leq t)$ as a variable about time in the following Eq. (2), without confusing with the current time t. If vehicle i is associated with base station j, we define the long term *service rate* as $R_{ij}(t)$, having

$$R_{ij}(t) = f_{ij}(t) \int_{t_0}^{t} x_{ij}(\tau) c_{ij}(\tau) d(\tau), \tag{2}$$

where $f_{ij}(t)$ denotes the fraction of resources that the base station j serves vehicle i, having $f_{ij}(t) = \frac{\sum_{\tau=t_0}^{\tau=t} x_{ij}(\tau)}{t-t_0}$. Let $x_{ij}(\tau)$ denote a scheduling indicator, $x_{ij}(\tau) \in \{0, 1\}$. If the base station j schedules the vehicle i at time τ, $t_0 \leq \tau \leq t$, we have $x_{ij}(\tau) = 1$, and vice versa.

When we measure the load balancing of a network, a good load balancing should satisfy the following two characteristics:

- *The overall service rates* $\sum_{j \in B} \sum_{i \in V} R_{ij}(t)$ *is large.*
- *The variance of users' service rates* $R_{ij}(t)$ *is small.*

It is intuitive in reality. First, a network with good balancing means that the network congestion is not severe. So the overall service rates should be large. Second, the aim of vehicular association is to make most of the vehicles can be associated with the base stations, rather than only several ones. Meanwhile, each associated vehicle should have a relative good service rate, with the rate fluctuating around the average level at least. Otherwise, a bad service rate below the average level greatly is meaningless for vehicles.

Here, we need to point out that there are some other metrics of load balancing. For example, in literature [9], [10], Sherman et al. use the utility function of delay to measure the load balancing in wifi offloading problem. This metric is usually used to study the performance about offloading some services from base stations to wifi access points. Yue *et al.* [11] and Han *et al.* [12] usually use the Jain's index of throughput to measure the load balancing. This utility is similar to above two characteristics of service rate stated by us. Based on the application background and convenience for comparing with the similar schemes, we use the *overall service rates* and the *variance of service rates* as the metrics of load balancing in our following experiment, which is as same as the metrics in literature [6].

C. Mobility and Resource Allocation

In literature [6], the system model is applied to the low mobility environment. The achievable rate c_{ij} is assumed to be independent of channel qualities. Correspondingly, the optimal resource allocation is equal allocation. In our paper, we take into account time-varying channels and vehicle mobility. Thus, the *proportional fair scheduling* is adopted in the paper. In stochastic settings, the proportional fair scheduling considers the system performance and user fairness. It can satisfy the service requirements of those terminals with good channel qualities, and meanwhile pay attention to the terminals with bad channel qualities.

In proportional fair scheduling scheme, we have the *allocation priority* of vehicle i as,

$$AP_{ij}(t) = \frac{c_{ij}(t)}{R_{ij}(t-1)}. \tag{3}$$

According to Eq. (3), each vehicle calculates its priority at each time slot. The base station will schedule vehicles based on the priority. We can see if the base station j continually schedules a vehicle with good channel quality, the value of denominator increases. Then the priority decreases and the fraction of resources of the vehicle to obtain also decreases.

Besides, in vehicular networks, the speed of a vehicle is much lower than that of the high-speed railway. So, for convenience, we set a threshold ϱ to update the achievable rate $c_{ij}(t)$. It means that if $|c_{ij}(t) - c_{ij}(t-1)| < \varrho$ (ϱ is a small positive number), we have $c_{ij}(t) = c_{ij}(t-1)$, otherwise, we update the achievable rate $c_{ij}(t)$.

IV. Preliminary

In this section, we briefly review the basic idea of reinforcement learning adopted in our design. In the introduction, we give the definition of reinforcement learning and introduce its characteristics briefly. The model of reinforcement learning system is shown in Fig.2.

There are four elements in the reinforcement learning system: a policy, a reward function, a value function and an optional model of the environment.

A policy defines the learning agent's way of behaving at a given time. Generally speaking, a policy is a mapping from perceived states of the environment to actions to be taken when in those states [2]. Assuming that S denotes the state space and A denotes the action space, we define a policy $\pi(s, a)$ as the probability of choosing action a in state s, $S \times A \rightarrow [0, 1]$.

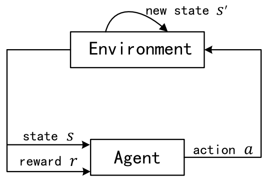


Fig. 2. The model of reinforcement learning.

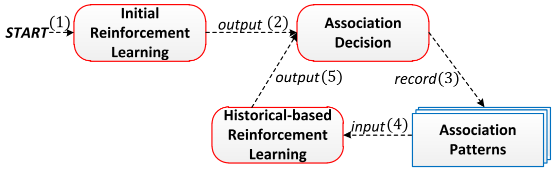


Fig. 3. The architecture of ORLA.

A reward function maps each perceived state (or state-action pair) of the environment to a scalar number. A reward r is an immediate sense with indicating the intrinsic desirability of that state.

A value function specifies what is good in the long run accumulation. Rewards are given directly by the environment, but values must be estimated and re-estimated from the sequences of observations. The common value functions include T-steps cumulative reward $\mathbb{E}[\frac{1}{T}\sum_{t=1}^{T} r_t]$ and discounting cumulative reward $\mathbb{E}[\sum_{t=0}^{+\infty} \gamma^t r_{t+1}]$, in which r_t denotes the reward in the t-th step and $\mathbb{E}$ denotes the expectation.

A model of environment is usually an optional component in some reinforcement learning systems. It mimics the behavior of the environment. Given a state and an action, the model might predict the resultant next state and next reward. In most reinforcement learning problems, the model of the environment is unknown.

From Fig.2, an agent receives an input s from the environment in the reinforcement learning system. Then according to the inner inference scheme, the agent outputs the corresponding action a. Under the action a, the environment transits to a new state s' and generates an immediate reward single r feeding back to the agent. Based on the current reward and state, the agent chooses the next action. The choosing principle is to make the probability of the positive reward increase. The object of a reinforcement learning agent is to maximize the total rewards it receives in the long run.

The model mentioned above is just a basic architecture of reinforcement learning. How to use it smartly in our ORLA design is provided in Section V.

V. Online Reinforcement Learning Association Scheme

In this section, we discuss the design principle and give the detailed description of ORLA. The architecture of ORLA is shown in Fig.3. In ORLA, we first design the *initial reinforcement learning method* to obtain the association results for vehicles in the dynamic environment (Steps (1) and (2) in Fig.3). These association results are cumulated in each base station (Step (3)). After a period of learning, when the base station meets network changes again, the base station can use the historical association patterns to solve the new association results directly and adaptively (Steps (4) and (5)), i.e., the *historical-based reinforcement learning method* for ORLA. Then the new obtained association results will be recorded again in each base station (Step (3)). Thus, taking advantage of historical association patterns, each base station does not need to do the initial reinforcement learning without any referential experiences. We let the historical-based reinforcement learning operate continually in ORLA. In Fig.3, we can see the association patterns, the historical-based reinforcement learning, and the association decision form a circle (Steps (3), (4) and (5)) to adaptively handle the network changes.

A. Initial Reinforcement Learning

1) Initialization: Each base station knows its service supplies/resources K_j and its service demands D_j. The initial value of D_j is defined as the number of vehicles that are in the communication range of base station j. Each user measures the SINR by using pilot signals and broadcasts its achievable rate c_{ij} to all base stations at each time slot.

Then, we define a *price value* for each base station as $\mu_j = D_j - K_j$. The price value can be either positive or negative. We also define a *decision value* between base station j and vehicle i as $d_{ij} = c_{ij} - \mu_j$.[2] We can see if the base station j is over-loaded, its price μ_j is high. Then the decision value may be small.

Besides, through communicating with other base stations periodically, each base station can maintain an *SINR matrix* with the element c_{ij} and an *association matrix*[3] with element $\{0, 1\}$. The value 1 means that there is an association between vehicle i and base station j, and vice versa. If the achievable rates c_{ij} of some vehicles are not received by the base stations, we set the corresponding values as zero in the SINR matrix. The dimensions of the two matrices are both $|V| \times |B|$.

2) Learning Method: Based on the above information, we design the initial reinforcement learning method for ORLA. It can be seen as a single-step reinforcement learning task. The theoretical model is *an N-armed bandit problem* [2]. In ORLA, each base station acts as an independent learning agent.

- The *environment* is the current vehicular network.
- The *action* is defined as the base station trying to build associations with some vehicles.
- The *reward* is defined as a reciprocal of the deviation of average service rate for all users (see Eq. (4)). For a base station j, assume that we obtain the association results through learning, i.e., knowing which vehicles associated with base station j. The reward r_j defined for the association of base station j can be calculated as:

$$r_j = \frac{1}{\sum_{i=1}^{|S_j|} \frac{1}{|S_j|} \cdot (R_{ij} - \frac{\sum_{k=1}^{|B|} \sum_{i=1}^{|V|} R_{ik}}{|V|})^2}, \tag{4}$$

where S_j denotes the set of vehicles associated with base station j. The values of S_j and R_{ij} can be obtained through the SINR matrix and the association matrix.

Before describing the learning method, we define a mathematical operator $Z \wr z$. It means that under the condition z we calculate the value of function Z.

At t-th iteration of initial reinforcement learning:

Step 1: Each base station calculates the decision value $d_{ij}(t) = c_{ij}(t) - \mu_j(t)$.

Step 2: Each base station sends the decision values to all vehicles.

Step 3: Each vehicle chooses the best decision value, i.e., $\arg\max_j d_{ij}(t)$, and tries to associate with the corresponding base station j. Then the set of actions $S_j(t)$ can be obtained. Note that in iterations, there may exist two or more set of actions that are totally equal, e.g., $S_j(t) = S_j(t+1)$. In the following parts, we set an index l to differ the different set of actions, denoted as S_j^l $(l = 1, 2, \ldots)$.

Step 4: Based on Step 3, each base station can calculate its current reward $r_j(t)$ according to Eq. (4), i.e., the value of $r_j(t) \wr S_j^l$.

Step 5: Calculate the *long term average cumulative reward* $Q_j(t)$ for the action set S_j^l, having

$$Q_j(t) \wr S_j^l = \frac{(Q_j(t-1) \wr S_j^l) \times count(S_j^l) + (r_j(t) \wr S_j^l)}{count(S_j^l) + 1}, \tag{5}$$

where $count(S_j^l)$ represents a counter to calculate the cumulative number of choosing the action set S_j^l.

Step 6: Then we adjust the price value according to the three following points.

- If the current reward $r_j(t) \wr S_j(t) \geq \frac{\sum_{k \in B, k \neq j} r_k(t) \wr S_k(t)}{|B|-1}$, we maintain the price $\mu_j(t+1) = \mu_j(t)$.
- Else if $|\frac{\sum_{i=1}^{|S_j(t)|} R_{ij}(t)}{|S_j(t)|} > \frac{\sum_{k=1}^{|B|} \sum_{i=1}^{|V|} R_{ik}(t)}{|V|}|$, we decrease the price value with $\mu_j(t+1) = (1 - \delta(t)) \cdot \mu_j(t)$.
- Else, we increase the price value with $\mu_j(t+1) = (1 + \delta(t)) \cdot \mu_j(t)$, where $\delta(t) \in [0, 1)$ is a stepsize chosen in the experiment.

Step 7: If for all S_j^l $(l = 1, 2, \ldots)$, satisfying $|Q_j(t) \wr S_j^l - Q_j(t-1) \wr S_j^l| < \epsilon$ (ϵ is a small positive number), the iteration ends. The base station obtains the final association results S_j^l based on $\arg\max_{S_j^l}(Q_j(t) \wr S_j^l)$.[4] Else, we turn to Step 1 and iterate continually.

Step 8: According to the final association results S_j^l in Step 7, the base station j knows which vehicles can be associated with it. Then base station j informs the corresponding vehicles to start associations, and meanwhile, base station j records the association results in itself.

[2] In the experiment, through scaling up/down, we set c_{ij} and μ_j having the same order of magnitude.

[3] Note that we consider the time-varying channels and vehicle mobility, the above SINR matrix and association matrix is related with time t. To simplify notations, we omit the index t in some parts, especially in Algorithm 1-3.

[4] If a vehicle receives more than one association signal, it will choose the best one with the maximal service rate to associate.

3) Analysis: In above initial reinforcement learning, there are two factors that motivate the iterations. One is the dynamic achievable rate $c_{ij}(t)$, the other is the price $\mu_j(t)$ resulting from the unbalanced associations. When the base station tries to build associations with some vehicles, it will receive a reward corresponding to these actions. Although each base station makes decisions itself, we have the reward r_j containing the global average service rate to guarantee the effect of learning. Then if the reward is bad, the base station adjusts its price, i.e., we use reward r_j as a basis to adjust the price of a base station. Through multiple iterations, we learn the best cumulative reward of these actions according to Eq. (5). The optimal actions can be chosen based on the cumulative reward. The overhead analysis of initial reinforcement learning is provided in Section V-D.

Due to mobility, the initial reinforcement learning with time-varying $c_{ij}(t)$ is a little bit complicated than the other optimization methods with constant rate c_{ij}. But it avoids the invalidation of association solution with bad overall service rates. In experiments, we obtain that the worst iteration time of initial reinforcement learning is below 1500ms. It is tolerable in vehicular associations.

B. Association Pattern

ORLA uses the initial reinforcement learning as a cold start. Due to network changes, each base station can cumulate a series of association patterns (association results) of its own after a period of time. The *association pattern* is defined as follows.

An association pattern of base station k is defined as under what kind of SINR condition and price value, which vehicles are associated with the base station k. Specially, the association pattern can be described/recorded by using the three following elements.

- **An SINR matrix**. It is denoted by C_p^k with the element c_{ij}, of which k denotes the index of the base station, p denotes the sequence number of the pattern.
- **An association matrix**. It is denoted by A_p^k with the element $\{0, 1\}$, of which the value 0 or 1 means whether the vehicle is associated with base station k or not;
- **A price value**. It has been defined in Section V-A.

Those recorded patterns are the valuable experiences for our following historical-based reinforcement learning in Section V-C. When doing the historical-based reinforcement learning, new obtained association patterns are continually recorded in each base station.

C. Historical-Based Reinforcement Learning

Based on above pattern records, when the environment changes, base stations will face new association demands coming from vehicles. Since the spatial-temporal regularities exist in vehicular networks, we can use the historical association patterns to deal with the network changes. Then, ORLA designs a historical-based reinforcement learning method for load balancing. Here, each base station is also regarded as a reinforcement learning agent. We give the pseudo-codes of historical-based reinforcement learning for base station k in Algorithm 1.

- The *environment* is the current vehicular network.
- The *action* is defined as the base station choosing one of the historical association patterns.
- The *reward* is as same as the definition in Eq. (4).

Algorithm 1 Historical-Based Reinforcement Learning

```
1: $r = 0$; $P$ denotes the set of the historical association patterns;
2: for ($p = 1$; $p \leq |P|$; $p++$) do
3:     $Q(p) = 0$, $count(p) = 0$;
4: end for
5: for ($t = 1$; $t \leq T$; $t++$) do
6:     if ($\max($function $sim(p', p)) < \lambda$) then
7:         Turn to the phase of the initial reinforcement learning;
8:     else
9:         if ($rand() < \varepsilon$) then
10:            if (elements in set $P$ are not totally selected ) then
11:                Choosing pattern $p$ with the condition
                   $\arg\max_p$ (function $sim(p', p)$);
12:                $P = P \setminus \{p\}$;
13:            else
14:                Choosing pattern $p$ from the set $P$ uniformly;
15:            end if
16:        else
17:            Choosing pattern $p$ with the condition
               $\arg\max_p Q(p)$;
18:        end if
19:    end if
20:    Do function allocation$(p', p)$;
21:    Calculating the reward $r$ according to Eq. (4);
22:    $Q(p) = \frac{Q(p) \times count(p) + r}{count(p) + 1}$;
23:    $count(p) = count(p) + 1$;
24: end for
25: Making a final action decision, i.e., choosing the final pattern $p$ with maximal $Q(p)$;
26: Do function allocation$(p', p)$;
27: Output the association matrix $A_{p'}^k$;
```

In the historical-based reinforcement learning, we assume that on base station k, it has a set of pattern records P. So there are $|P|$ different kinds of actions. We initialize $count(p)$ as the chosen times of the historical pattern p and $Q(p)$ as the cumulative reward of pattern record p (Lines 2-3). After that, for a current case p' with current SINR matrix $C_{p'}^k$ and current price $\mu_{p'}^k$ of the base station k, we calculate the similarity between the current case p' and each historical pattern p. If the maximal similarity is below a threshold λ (defined in experiments), we will turn to the phase of initial reinforcement learning (Lines 6-7). It means that the historical experiences have low utilities to solve the current association problem. Otherwise, we focus on choosing actions among the $|P|$ association patterns in the number of T iterations, $T > |P|$. The T iterations end until for all $p \in P$, we have $Q(p)$ converge, i.e., the value of $|Q_t(p) - Q_{t-1}(p)|$ is below a small positive number. Note that, since we consider the time varying channels, the new pattern p' may change after once iteration. This is the reason why we need to learn. The environment changes, the reward values in Lines 21-22 also change.

In both initial reinforcement learning and the historical-based reinforcement learning, the convergence solutions are usually the vehicles whose channel changing is not greatly severe. For those vehicles that are in the border of two base stations' coverage areas, the solutions often converge in vehicles with the achievable rates calculated in the area where the head of the vehicle towards to.

Algorithm 2 Similarity of Two Patterns: *sim*(p', p)

```
Input:
    The current and historical SINR matrices $C^k_{p'}$ and $C^k_p$;
    The current and historical prices $\mu^k_{p'}$ and $\mu^k_p$;
Output:
    The similarity value;
1: Set $vec(k, p')$= sort the $k$-th column of matrix $C^k_{p'}$;
   Set $vec(k, p)$= sort the $k$-th column of matrix $C^k_p$;
2: Calculate the Pearson distance $PD(vec(k, p'), vec(k, p))$ between vector $vec(k, p')$ and $vec(k, p)$;
3: Set $w(k, p') = \sum_{j=k} c_{ij}, c_{ij} \in C^k_{p'}$;
   Set $w(k, p) = \sum_{j=k} c_{ij}, c_{ij} \in C^k_p$;
4: Set $W(p', p) = w(k, p')/w(k, p)$;
5: Set $U(p', p) = \mu^k_{p'}/\mu^k_p$;
6: Calculate the Kullback-Leibler distance $KL(W(p', p)\|U(p', p))$;
7: Output the similarity value
   $\alpha \cdot PD(vec(k, p'), vec(k, p)) - \beta \cdot KL(W(p', p)\|U(p', p))$;
```

In Algorithm 1, there exists a trade-off problem between exploration and exploitation in the reinforcement learning. Exploration-only gives the chance to each action uniformly while exploitation-only gives the chance to the best rewarded action at present. Obviously, exploration-only can estimate the corresponding reward for each action with losing many chance to choose the optimal action. While, exploitation-only cannot estimate the expectation of the reward well for each action. If we want to maximize the final cumulative reward, we need to find a middle way between exploration and exploitation.

Here we use a ε greedy method [2] to choose the actions. It can balance the exploration and exploitation. When the random value is below the threshold ε, we first use the maximal similarity to choose the action (Lines 9-12). Then if all the patterns are traversed, we will choose the action uniformly (Lines 13-15). When the random value is beyond the threshold ε, we choose the action with maximal cumulative reward (Lines 16-18). Through multiple iterations, the average reward $Q(p)$ of each action can be approached (Lines 19-23). In this method, the maximal similarity can guarantee the algorithm to find the possible optimal value quickly. Finally, we choose the association action with the best cumulative reward (Lines 25-26).

In Algorithm 1, there are two important components, Algorithm *sim*(p', p) and Algorithm *allocation*(p', p), shown in Algorithm 2 and Algorithm 3. Algorithm *sim*(p', p) is used to calculate the similarity between the historical pattern p and the current case p'. Algorithm *allocation*(p', p) is used to allocate the possible association actions for vehicles in current case p' based on the historical association pattern p. We describe the two algorithms as follows.

Algorithm 3 Association Allocation: *allocation*(p', p)

```
Input:
    The current and historical SINR matrices $C^k_{p'}$ and $C^k_p$;
    The current and historical prices $\mu^k_{p'}$ and $\mu^k_p$;
    The historical association matrix $A^k_p$;
Output:
    An association matrix $A^k_{p'}$ for SINR matrix $C^k_{p'}$;
1: Sort the elements $c_{ij}$ in $C^k_p$ with $j = k, c_{ij} \neq 0$ and put them into vector $X_{p,k}$;
2: Sort the elements $c_{ij}$ in $C^k_{p'}$ with $j = k, c_{ij} \neq 0$ and put them into vector $X_{p',k}$;
3: Sort the elements $c_{ij}$ in $X_{p,k}$ with the corresponding element in $A^k_p = 1$ and put them into vector $Y_{p,k}$;
4: Define a set $Y = \emptyset$ that is used to record the chosen association elements for the current case $p'$;
5: Calculate the number of vehicles that requires to be associated as:
   $NUM = \lceil \frac{\mu^k_{p'}}{\mu^k_p} \times \frac{dim(X_{p',k})}{dim(X_{p,k})} \times dim(Y_{p,k}) \rceil$;
   $/*$ Let $dim(\cdot)$ denote the dimension/length of a vector. $*/$
6: if $(NUM < dim(Y_{p,k}))$ then
7:     Use binary approaching method to obtain association matrix $A^k_{p'}$, which is described in Algorithm 4;
8: else
9:     Use multi-spot diffusion method to obtain association matrix $A^k_{p'}$, which is described in Algorithm 5;
10: end if
11: Output the association matrix $A^k_{p'}$;
```

1) Similarity of Two Patterns: In this work, the *similarity* of two patterns is defined as the proximity degree of the distribution of service requirements c_{ij} under a certain distribution of prices. Therefore, in Algorithm *sim*(p', p) (pseudo-codes in Algorithm 2), we first use the Pearson distance to calculate the distribution similarity of the service requirement c_{ij} between the historical pattern p and the current case p' for base station k (Lines 1-2). The Pearson distance is used to measure the correlation between two samples. It can be used in the situation with different orders of magnitudes or evaluation criteria. Since in our study, the current requirement c_{ij} in $C^k_{p'}$ and the historical c_{ij} in C^k_p may have different scales, the Pearson distance is suited to characterize the similarity between them. The range of value is $[-1, 1]$. The value 1 means the maximal positive correlation.

Then, we use the Kullback-Leibler distance to calculate the distribution similarity between the requirement ratio (Line 4) and the price ratio (Line 5) for the historical pattern p and the current case p' (Lines 3-6). The Kullback-Leibler distance is used to measure the similarity between two distributions. It can obtain the uncertainty degree of distribution $W(p', p)$¡¡

by using distribution $U(p', p)$ approximately estimating the distribution $W(p', p)$. The range of value is [0, 1]. The value 0 means that the two distributions are the same.

Finally, we have a synthetical similarity with putting different weights α and β on the Pearson distance and Kullback-Leibler distance, respectively. Usually we have $\alpha = \beta = 0.5$.

2) Association Allocation: In Algorithm 1, when the base station chooses a historical pattern as its current action, ORLA uses Algorithm $allocation(C^k_{p'}, C^k_p)$ to make an association allocation for the current vehicles based on the historical experience. Our *principle* is to make the current association allocation having the similar allocation distribution to the historical association. The pseudo-codes are described in Algorithm 3.

First, for a base station k, ORLA needs to solve how many vehicles can be associated with it in the current case. Here, ORLA uses the proportional allocation method (Lines 1-5 in Algorithm 3). Let $dim(\cdot)$ denote the dimension of a vector. Specially, if the base station k allocates the number of $dim(Y_{p,k})$ vehicles from $dim(X_{p,k})$ demands with price μ^k_p in the historical pattern p, the base station k will allocate the proportional number of NUM vehicles from $dim(X_{p',k})$ demands with price $\mu^k_{p'}$ for the current case p'.

Second, ORLA needs to solve which vehicles can be associated with the base station k in the current case. It is classified into two situations (Lines 6-10 in Algorithm 3). Note that in Algorithm 4 and 5, some initialization/prior information has been provided in Algorithm 3 (Lines 1-4).

- **Situation 1:** $NUM < dim(Y_{p,k})$. It means that the number of vehicles required to be associated in the current case p' is below the number of allocated vehicles $dim(Y_{p,k})$ in the historical pattern p. Here, we use a *binary approaching method*, described in Algorithm 4.
- **Situation 2:** $NUM \geq dim(Y_{p,k})$. It means that the number of vehicles required to be associated in current case p' is beyond the number of allocated vehicles $dim(Y_{p,k})$ in historical pattern p. Here, we use a *multi-spot diffusion method*, described in Algorithm 5.

3) Analysis and Brief Summary: In Algorithm 4 and 5, the principle is to choose the vehicles having the similar features to the historical association pattern.

When $NUM < dim(Y_{p,k})$, we use the binary approaching method (Algorithm 4) to scale down and find the appropriate elements. We utilize the binary method to partition the historical association vector continually and find the feature distribution of the associated vehicles. By using the same feature distribution, we can finally find the appropriate vehicles for forthcoming associations. The detailed explanations of each step in Algorithm 4 can be found in APPENDIX A [26].

When $NUM \geq dim(Y_{p,k})$, we mainly utilize the multi-spot diffusion method (Algorithm 5) to scale up and find the appropriate elements. First, we choose the equal number of elements from the current pattern p' with the same rank location in the historical pattern p (Lines 1-7). We call these elements as *spots*. Then, around these spot elements, we enlarge the search range and choose integral multiple elements (Lines 8-13). If having a remainder, we turn to the binary approaching method to complete the association allocation (Lines 14-16). Finally, we obtain the association matrix.

Algorithm 4 Binary Approaching Method

```
1: Split the vector $X_{p,k}$ into two equal vectors $X^{up}_{p,k}$ and $X^{down}_{p,k}$;
   Split the vector $X_{p',k}$ into two equal vectors $X^{up}_{p',k}$ and $X^{down}_{p',k}$;
2: Count the common elements both in vectors $Y_{p,k}$ and $X^{up}_{p,k}$, denoted as $N_{up}$; Count the common elements both in vectors $Y_{p,k}$ and $X^{down}_{p,k}$, denoted as $N_{down}$;
3: if $(NUM = 1)$ then
4:     Choose a non-zero value $y$ randomly from $X_{p',k}$ except the elements in set $Y$;
5:     Set $Y = Y \cup \{y\}$;
6: else if $(N_{up} > N_{down}$ and $|\frac{N_{up}-N_{down}}{dim(X_{p,k})}| > \theta)$ then
7:     Choose a non-zero value $y$ uniformly from vector $X^{up}_{p',k}$ except the elements in set $Y$;
8:     Set $Y = Y \cup \{y\}$; $NUM = NUM - 1$;
9:     Set $X_{p,k} = X^{up}_{p,k}$ and $X_{p',k} = X^{up}_{p',k}$;
10:    Do binary recursion continually;
11: else if $(N_{up} < N_{down}$ and $|\frac{N_{up}-N_{down}}{dim(X_{p,k})}| > \theta)$ then
12:    Do the similar operations as Steps 7-10 in the opposite vector;
13: else if $(|\frac{N_{up}-N_{down}}{dim(X_{p,k})}| \leq \theta)$ then
14:    Choose two non-zero values $y$ and $y'$ uniformly from vector $X^{up}_{p',k}$ and vector $X^{down}_{p',k}$ respectively except the elements in set $Y$;
15:    Set $Y = Y \cup \{y, y'\}$; $NUM = NUM - 2$;
16:    Do the similar operations as Steps 9-10;
17: end if
18: Set the corresponding values of set $Y$ in matrix $A^k_{p'}$ as 1;
```

After the base station learns the association decision $A^k_{p'}$ by using the historical-based reinforcement learning, it informs vehicles to associate with it.

Note that in Algorithm 3, since there are spatial-temporal regularities in vehicular networks, some patterns have high similarity. ORLA only reserves one of them in order to avoid the curse of dimensionality. It also can speed up the learning process of ORLA.

D. Complexity Analysis

We analyze the complexity of ORLA from the following three aspects.

First, in initial reinforcement learning, we let each base station act as an independent agent. Although the association decision is made in a distributed manner, we still need some cooperative information to guide the iterations without losing the global aim of load balancing. Besides, it is normal that coping with the dynamic environment may bring about more overheads than the static one. The amount of cooperative information is order of $I \cdot (|B||B-1| + |B||V|)$, where I is the total number of iterations. The information exchange includes the interactions among base stations (SINR matrix

Algorithm 5 Multi-Spot Diffusion Method

```
Define a new set $SPOT$;
Initialize an object $rank$ that is used to label the rank location of an element in a vector;
for (each element $i$ in vector $Y_{p,k}$) do
    $rank.i$ =the location of element $i$ in vector $X_{p,k}$;
    Choose the corresponding element $j$, with rank location $rank.j = \frac{dim(X_{p',k})}{dim(X_{p,k})} \times rank.i$, from $X_{p',k}$;
    Put the element $j$ into the set $SPOT$;
end for
Set $q = \lfloor \frac{NUM}{dim(Y_{p,k})} \rfloor$;
for (each element $j$ in set $SPOT$) do
    Set the element $j$ as a center spot in $X_{p',k}$ and choose number of $q$ elements from $X_{p',k}$ with the nearest distance from the center spot $j$, including the spot itself.
    Put the number of $q$ elements into the set $Y$;
end for
Set the corresponding values of set $Y$ in matrix $A^k_{p'}$ as 1;
if ($NUM - q \cdot dim(Y_{p,k}) \neq 0$) then
    Turn to use the binary approaching method;
end if
```

and association matrix) and the interactions between the base stations and vehicles (c_{ij} broadcast).

Fortunately, in a big vehicular network, the network area is usually divided into many small areas based on road intersections. It is allowed that the load balancing is achieved only in small local areas. Thus the overhead can be controlled in local areas. In our experiments, we also choose a region to do the association algorithm. The scalability in large scale dynamic networks is our future work.

Second, in historical-based reinforcement learning, the action selection does not need to try blindly in many times since each action can be traversed by using similarity as its guidance. Besides, due to the spatial-temporal features of traffic flows, there are many similar historical patterns. ORLA combines the similar ones and only maintains some representative association patterns. It guarantees the search space will not increase too large. Thus, the historical-based reinforcement learning can also converge well. In the experiment, we also can see the good performance of the convergence for ORLA.

Third, since we push all the calculations on the base station side, not the user side, the computational capability can be guaranteed for some tasks like binary search and element sorting.

VI. Evaluation

We use real-life GPS-based vehicle mobility traces to evaluate the efficiency of ORLA. Our dataset comes from QiangSheng taxi movement. The dataset contains traces about 117 taxis. The traces are collected from April 1st to April 30th in Shanghai, China, 2015. The taxi periodically sends reports back to the data collector via an onboard GPS-enabled device. The information in the dataset includes vehicle ID, latitude, longitude, timestamp, vehicle moving speed, heading direction and onboard information.

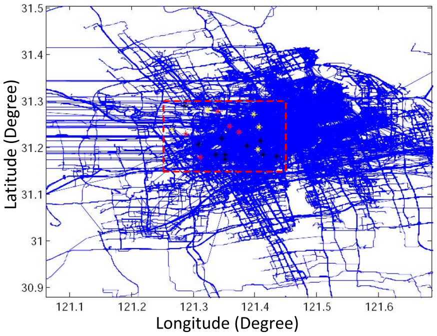
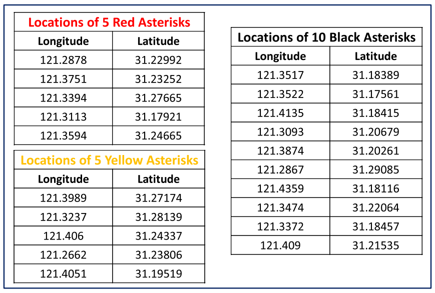


Fig. 4. The traces of QiangSheng taxi movement and the experiment area marked by the red rectangle.



Locations of 5 Red Asterisks	
Longitude	Latitude
121.2878	31.22992
121.3751	31.23252
121.3394	31.27665
121.3113	31.17921
121.3594	31.24665

Locations of 5 Yellow Asterisks	
Longitude	Latitude
121.3989	31.27174
121.3237	31.28139
121.406	31.24337
121.2662	31.23806
121.4051	31.19519

Locations of 10 Black Asterisks	
Longitude	Latitude
121.3517	31.18389
121.3522	31.17561
121.4135	31.18415
121.3093	31.20679
121.3874	31.20261
121.2867	31.29085
121.4359	31.18116
121.3474	31.22064
121.3372	31.18457
121.409	31.21535

Fig. 5. The GPS locations of 20 base stations.

In Fig.4, the blue lines depict the moving traces of 117 vehicles. In those traces, there are some noisy data with including some weird long straight segments that could not represent real taxi paths in a city. We preprocess the dataset and capture an area to do experiments. The captured area is marked by the red rectangle in Fig.4. The latitude of the captured area is between $[31.15, 31.30]$. The longitude is between $[121.25, 121.45]$. The high-resolution original image of Fig.4 is provided in [27]. In this area, there are totally 76 vehicles and 20 base stations, including 5 macrocells (red asterisks), 5 picocells (yellow asterisks) and 10 femtocells (black asterisks). The GPS locations of 20 base stations depicted in Fig.4 are provided in Fig.5. The transmit powers of the three ties of base stations are 46dBm, 35dBm and 20dBm, respectively. For the macros/picocells, we set the path loss $L(d_{ij}) = 34 + 40\log(d_{ij})$. For the femtocells, we set the path loss $L(d_{ij}) = 37 + 30\log(d_{ij})$, where d_{ij} denotes the distance between the vehicle i and base station j. The noise power σ^2 is -104dBm. The bandwidth is 10MHz. Besides, we set the similarity threshold λ as -0.25 in the experiment.

A. Loads Among Different Base Stations

In the paper, we compare ORLA with two association schemes, max-SINR and 3D (Distributed Dual Decomposition Optimization) [6]. The former is a traditional scheme, in which users choose the association base station with the maximal SINR. The latter is a popular method that transforms the user association to a utility maximization problem. The problem is solved by using gradient descent and dual decomposition. The algorithm of 3D is deployed separately on user side and base station side in heterogeneous networks.

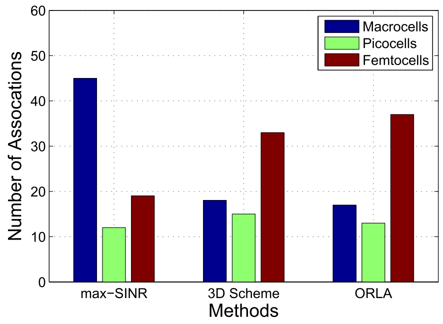


Fig. 6. Number of associations in three ties of base stations.

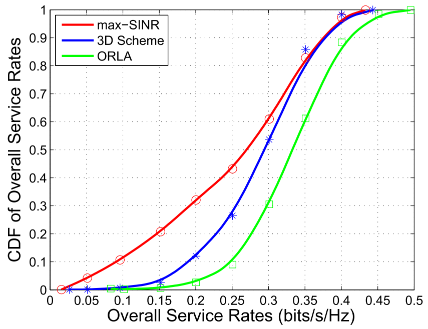


Fig. 7. The CDFs of overall service rates for three comparison methods.

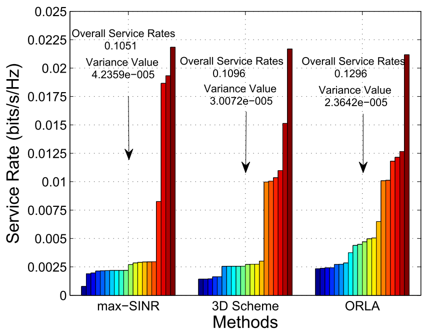


Fig. 8. The distribution of service rates.

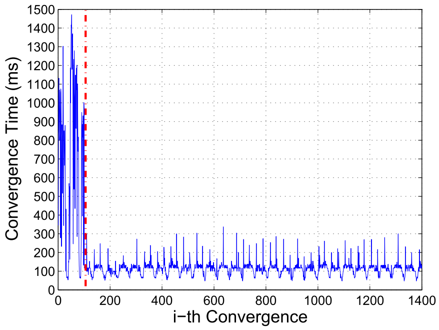


Fig. 9. The convergence time of ORLA.

We compares loads among three different association methods. In Fig.6, we capture the association results when convergence ends. The max-SINR association results in unbalanced loads, in which the macrocells are over-loaded, while the picocells and femtocells only serve fewer users. In ORLA, the load is shifted to the less congested femtocells, which suggests that our scheme ORLA alleviates the asymmetric load problem. It shows the effectiveness of ORLA. The results of 3D and ORLA are similar since the near-optimal results are both obtained by them.

B. Service Rates and Convergence

In Section III-B, we give the metrics of load balancing, i.e., the overall service rates and the variance of users' service rates. The large overall service rates and small variance mean that the base stations are not congested and can provide enough resources to support the network services.

Fig.7 shows the cumulative distribution functions (CDFs) of the overall service rates for three different association schemes. First, we can see that the beginning point of ORLA is bigger than max-SINR and 3D scheme, and the tail of ORLA is also longer than them. Second, the CDF of ORLA improves at a low rate compared with max-SINR and 3D scheme. *All above phenomena show that the bigger overall service rates occupy a high proportion in ORLA*.

Fig.8 shows the distribution of service rates for 20 high-activity vehicles. By using the statistical tools in MATLAB, in Fig.8, the overall service rates are 0.1051bits/s/Hz, 0.1096bits/s/Hz and 0.1296bits/s/Hz for max-SINR, 3D and ORLA, respectively. Meanwhile, the variance values of service rates are 4.2359e-005, 3.0072e-005 and 2.3642e-005 for max-SINR, 3D and ORLA, respectively. We obtain that ORLA has the largest overall service rates with smallest variance value. Therefore, combining Fig.7 with Fig.8, *we can see that ORLA provides a more uniform user experience with higher service rates for vehicles*. It also benefits the overall performance of the whole network system.

Fig.9 shows the convergence time in 1400 successive convergence tests on QiangSheng dataset. Each convergence contains multiple iterations. First, we can see that the worst convergence time is still below 1500ms. Second, there is an obvious boundary in Fig.9, marked by a dash line. The left side of the dash line represents the initial reinforcement learning, and the right side represents the historical-based reinforcement learning. It means that, at the beginning, ORLA pays much more time to cope with the dynamic traffic flows (new arrivals and departures). After that, based on the historical experiences, 76.9 % convergence time is saved. All above phenomena verify the effectiveness of ORLA in vehicular networks.

C. Analysis

Here, we give some analyses of the results in Fig.7, Fig.8 and Fig.9. First, in 3D scheme, the assumption of constant rate c_{ij} and Poisson point process model may result in the invalidation of the association solution in vehicular networks, since the vehicles have run away from the current location. It means that the association solution is not appropriate for the current environment. The performance of dealing with the dynamic case of ORLA is stronger than 3D. Thus, the variance and the overall service rates of ORLA are in better state than that of 3D.

Second, once the network changes, the 3D scheme must rerun in the whole network area. It costs high when doing re-association in the new network environment. In fact, the traffic flows have the spatial-temporal regularity in vehicular networks. It inspires us to directly utilize the historical and regular association patters to allocate the resources to current vehicles. In the historical-based phase of ORLA, we use the historical records to learn the association solutions. The results may not be optimal, but in the dynamic network, the near-optimal is accepted since the most important thing is to make vehicles communicate with base stations timely. In the historical-based phase, taking advantage of the regularity, our association patterns are alleviated into several ones in the experiment. It largely scales down the learning dimension, and meanwhile speeds up our learning process. Besides, we use similarity to help the algorithm traverse each historical association pattern. It avoids the algorithm trying many times blindly. Above all, although it seems that there are more operations in ORLA than in 3D scheme, the spatial-temporal regularities indeed help ORLA save the convergence time a lot so as to cope with the dynamic environment.

D. Discussion

In above experiments, the achievable rate is set to be updated according to threshold ϱ. The threshold is set as 0.0001 bits/s/Hz, a small positive number. This value can be seen as an empirical value which comes from the statistical results of all the historical achievable rates. The threshold decides the sensitivity of the change of achievable rate. The smaller the threshold value is, the higher precision the experimental results have. Specially, a better way is to update this threshold dynamically with respect to the vehicle's mobility change. For example, if the speed of a vehicle is low in reality, the change of achievable rate between the vehicle and the base station will be small. Thus the threshold ϱ can be set small for this vehicle, and vice versa. In the future, we can utilize this way to investigate our study.

Additionally, we know that there are various kinds of network traffic flows generated by private cars, taxies, etc. Due to privacy protection, we cannot collect and obtain the data from the private cars. Thus, in the experiments, we use the taxi traffic to emulate the generic vehicular network traffic. It is an assumption of our study. In the future, we aim to collect more types of traffic flows to enrich our experiments.

VII. Conclusion

In this work we propose a scheme ORLA for load balancing in the vehicular networks with heterogeneous base stations. It can provide good service rates for vehicles. In the paper, ORLA includes the phases of the initial reinforcement learning and the historical-based reinforcement learning. Our design exploits the spatial-temporal characteristics of the traffic flows. Through interacting with the dynamic vehicular networks, we let ORLA learn the association decision intelligently. In the long run, ORLA can cope with the dynamic changes well. Our extensive evaluations demonstrate its effectiveness against the traditional max-SINR scheme and popular 3D scheme.

References

[1] J. Cheng, J. Cheng, M. Zhou, F. Liu, S. Gao, and C. Liu, "Routing in Internet of vehicles: A review," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 5, pp. 2339–2352, Oct. 2015.

[2] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 1998.

[3] L. Li, Y. Lv, and F.-Y. Wang, "Traffic signal timing via deep reinforcement learning," *IEEE/CAA J. Automat. Sinica*, vol. 3, no. 3, pp. 247–254, Apr. 2016.

[4] Z. Shen, J. G. Andrews, and B. L. Evans, "Adaptive resource allocation in multiuser OFDM systems with proportional rate constraints," *IEEE Trans. Wireless Commun.*, vol. 4, no. 6, pp. 2726–2737, Nov. 2005.

[5] F. Boccardi *et al.*, "Why to decouple the uplink and downlink in cellular networks and how to do it," *IEEE Commun. Mag.*, vol. 54, no. 3, pp. 110–117, Mar. 2016.

[6] Q. Ye, B. Rong, Y. Chen, M. Al-Shalash, C. Caramanis, and J. G. Andrews, "User association for load balancing in heterogeneous cellular networks," *IEEE Trans. Wireless Commun.*, vol. 12, no. 6, pp. 2706–2716, Jun. 2013.

[7] J. Andrews, S. Singh, Q. Ye, X. Lin, and H. Dhillon, "An overview of load balancing in HetNets: Old myths and open problems," *IEEE Wireless. Commun.*, vol. 21, no. 2, pp. 18–25, Apr. 2014.

[8] H.-S. Jo, Y. J. Sang, P. Xia, and J. G. Andrews, "Heterogeneous cellular networks with flexible cell association: A comprehensive downlink SINR analysis," *IEEE Trans. Wireless Commun.*, vol. 11, no. 10, pp. 3484–3495, Oct. 2012.

[9] N. Cheng, N. Lu, N. Zhang, X. S. Shen, and J. W. Mark, "Vehicular WiFi offloading: Challenges and solutions," *Veh. Commun.*, vol. 1, no. 1, pp. 13–21, 2014.

[10] N. Cheng, N. Lu, N. Zhang, X. S. Zhang, X. Shen, and J. W. Mark, "Opportunistic WiFi offloading in vehicular environment: A game-theory approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 7, pp. 1944–1955, Jul. 2016.

[11] C. Yue, G. Xue, H. Zhu, J. Yu, and M. Li, "S3: Characterizing sociality for user-friendly steady load balancing in enterprise WLANs," in *Proc. IEEE ICDCS*, Philadelphia, PA, USA, Jul. 2013, pp. 491–499.

[12] H. Han *et al.*, "E3: Energy-efficient engine for frame rate adaptation on smartphones," in *Proc. ACM SenSys*, Rome, Italy, Nov. 2013, pp. 15–28.

[13] X. Chen, J. Wu, Y. Cai, H. Zhang, and T. Chen, "Energy-efficiency oriented traffic offloading in wireless networks: A brief survey and a learning approach for heterogeneous cellular networks," *IEEE J. Sel. Areas Commun.*, vol. 33, no. 4, pp. 627–640, Apr. 2015.

[14] H. Kim, G. De Veciana, X. Yang, and M. Venkatachalam, "Distributed α-optimal user association and cell load balancing in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 177–190, Feb. 2012.

[15] S. Corroy, L. Falconetti, and R. Mathar, "Dynamic cell association for downlink sum rate maximization in multi-cell heterogeneous networks," in *Proc. IEEE ICC*, Ottawa, ON, Canada, Jun. 2012, pp. 2457–2461.

[16] Y. Bejerano and S. J. Han, "Cell breathing techniques for load balancing in wireless LANs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 6, pp. 735–749, Jun. 2009.

[17] S. Das, H. Viswanathan, and G. Rittenhouse, "Dynamic load balancing through coordinated scheduling in packet data systems," in *Proc. IEEE INFOCOM*, San Francisco, CA, USA, Apr. 2003, pp. 786–796.

[18] P. Kolios, K. Papadaki, and V. Friderikos, "Efficient cellular load balancing through mobility-enriched vehicular communications," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2971–2983, Oct. 2015.

[19] E. Stevens-Navarro, Y. Lin, and V. W. S. Wong, "An MDP-based vertical handoff decision algorithm for heterogeneous wireless networks," *IEEE Trans. Veh. Technol.*, vol. 57, no. 2, pp. 1243–1254, Mar. 2008.

[20] S. E. Elayoubi, E. Altman, M. Haddad, and Z. Altman, "A hybrid decision approach for the association problem in heterogeneous networks," in *Proc. IEEE INFOCOM*, San Diego, CA, USA, Mar. 2010, pp. 401–405.

[21] S. Shakkottai, E. Altman, and A. Kumar, "Multihoming of users to access points in WLANs: A population game perspective," *IEEE J. Sel. Areas Commun.*, vol. 25, no. 6, pp. 1207–1215, Aug. 2007.

[22] D. Niyato and E. Hossain, "Dynamics of network selection in heterogeneous wireless networks: An evolutionary game approach," *IEEE Trans. Veh. Technol.*, vol. 58, no. 4, pp. 2008–2017, May 2009.

[23] E. Aryafar, A. Keshavarz-Haddad, M. Wang, and M. Chiang, "Rat selection games in HetNets," in *Proc. IEEE INFOCOM*, Turin, Italy, Apr. 2013, pp. 998–1006.

[24] P. Muñoz, R. Barco, J. M. Ruiz-Avilés, I. de la Bandera, and A. Aguilar, “Fuzzy rule-based reinforcement learning for load balancing techniques in enterprise LTE femtocells,” *IEEE Trans. Veh. Technol.*, vol. 62, no. 5, pp. 1962–1973, May 2013.

[25] Y.-H. Chiang and W. Liao, “Genie: An optimal green policy for energy saving and traffic offloading in heterogeneous cellular networks,” in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2013, pp. 6230–6234.

[26] *Appendix-ITS*, accessed on May 25, 2017. [Online]. Available: https://www.dropbox.com/s/e81srmmh2v3kmq4/appendix-ITS.pdf?dl=0

[27] *Distribute*, accessed on May 19, 2017. [Online]. Available: https://www.dropbox.com/s/st2vxsmtl5ojjef/distribute.pdf?dl=0

Zhong Li received the Ph.D. degree from Tongji University, Shanghai, China, in 2015. She is a Lecturer with Donghua University, Shanghai, China. Her research interests include vehicular networks, wireless networking, and Internet of Things.

Cheng Wang received the Ph.D. degree from the Department of Computer Science, Tongji University, in 2011. He is a Professor of Computer Science with Tongji University. His research interests include wireless networking, mobile social networks, and cloud computing.

Chang-Jun Jiang has been the Leading Scientist of the 973 Program on the project Model and Theory in Internet Information Service. He is currently the President of Donghua University and a Professor with Tongji University, Shanghai. He has authored or co-authored over 200 papers and four books. His research interests include concurrency theory, Petri nets, formal verification, services computing, and massive information service. He is a Council Member of the China Automation Federation and Artificial Intelligence Federation, the Vice-Chair of the Technical Committee of Management Systems of China Automation Federation, and an Information Area Specialist of the Shanghai Municipal Government.