



Time-optimal and privacy preserving route planning for carpool policy

Congcong Zhu¹ · Dayong Ye¹ · Tianqing Zhu¹ · Wanlei Zhou²

Received: 1 June 2021 / Revised: 14 October 2021 / Accepted: 10 February 2022 /
Published online: 4 March 2022
© The Author(s) 2022

Abstract

To alleviate the traffic congestion caused by the sharp increase in the number of private cars and save commuting costs, taxi carpooling service has become the choice of many people. Current research on taxi carpooling services has focused on shortening the detour distances. While with the development of intelligent cities, efficiently match passengers and vehicles and planning routes become urgent. And the privacy between passengers in the taxi carpooling service also needs to be considered. In this paper, we propose a time-optimal and privacy-preserving carpool route planning system via deep reinforcement learning. This system uses the traffic information around the carpooling vehicle to optimize passengers' travel time, not only to efficiently match passengers and vehicles but also to generate detailed route planning for carpooling vehicles. We conducted experiments on an Internet of Vehicles simulator CARLA, and the results demonstrate that our method is better than other advanced methods and has better performance in complex environments.

Keywords Taxi carpooling problem · Deep reinforcement learning · Route planning · Time estimation

This article belongs to the Topical Collection: *Special Issue on Web Intelligence = Artificial Intelligence in the Connected World*

Guest Editors: Yuefeng Li, Amit Sheth, Athena Vakali, and Xiaohui Tao

✉ Tianqing Zhu
Tianqing.Zhu@uts.edu.au

Congcong Zhu
Congcong.ZHU@student.uts.edu.au

Dayong Ye
Dayong.Ye@uts.edu.au

Wanlei Zhou
wlzhou@cityu.edu.mo

¹ School of Computer Science, University of Technology Sydney, Ultimo, Australia

² City University of Macau, Macau, Macao

1 Introduction

In recent years, traffic congestion caused by the sharp increase in the number of vehicles has become a significant problem that must be solved in the development of cities. In order to ease traffic congestion and save commuting time, public transportation has become the choice of many people, in which taxis (which also include ride-hailing apps like Uber, Lyft, and DiDi) play an important role. According to a New York City reports [5], there were more than 20 million classic taxi trips daily in New York City in 2019, and 76 million ride-hailing trips were taken daily. Faced with such a massive demand for taxis, to further alleviate traffic congestion [7], a new taxi travel mode based on the current traffic road network, carpooling, is proposed.

In a taxi carpooling service, the starting point and destination of each passenger may be different, but there is overlap in the itinerary. This method can not only alleviate traffic congestion but also can reduce users commuting costs. For example, in the report in 2019, carpooling was accounted for 23% of the total taxi trips, and each carpooling trip had at least two passengers, so nearly half of the users chose carpooling service.

Carpooling service problem can be divided into two scenarios. One is private car-sharing, and another is taxi carpooling. In private car sharing, users share their private car with other passengers. In this scenario, private car drivers have their own destination, and the routes of other passengers generally need to be highly overlapping with the driver, and the travel distance is usually quite long. In the scenario of taxi carpooling, taxi drivers have no personal destination, and the routes are completely based on the needs of passengers. The routes between passengers do not need to be highly overlapped, and the distance is short. Compared with private car sharing, taxi carpooling is more complex and more commonly used. Due to the different starting points and destinations of passengers in taxi carpooling, how to dispatch vehicles and assign passengers to taxis to achieve more effective carpooling services has become the main issue of current research.

In a classic taxi carpooling environment, multiple individual passengers in the same area use the ride-hailing apps to upload their destination and current location to a central decision-making system. The system searches for available taxis and, based on location information, assigns passengers in a similar direction to a taxi, which picks them up and drops them off at the destination one by one. In this process, the central decision-making system needs to match passengers and taxis according to the location, destination, the maximum number of passengers taxis can carry. Efficiently matching taxis and passengers is an essential part of decision-making.

At present, the research on carpooling service problem mainly focuses on shortening the travel detours and minimizing vehicle mileage and passenger travel cost [27]. However, with the rapid development of electric vehicles and transportation networks in smart cities, vehicle mileage and detours are no longer the primary concerns, and commuting time is more of a problem. Another critical issue in carpooling is privacy [25]. Many passengers have concerns about their privacy when choosing a carpooling service [18], which exposes the location of their home or business to other passengers. To address these problems, we propose a Time-Optimal and Privacy-Preserving carpool (TO-PP) route planning system. The main contributions of our work are summarized as follows:

- 1) We preserve passenger-to-passenger privacy in the carpooling problem.
- 2) We use the real-time traffic information to optimize the commuting time of passengers, not only to efficiently match passengers but also to generate detailed route planning for carpooling vehicles.

The rest of the paper is organized as follows: Section 2 described related work on carpooling service problems and deep reinforcement learning. In Section 3, privacy, traffic information, and the model are defined. Section 4 proposes our time-optimal and privacy-preserving carpooling planning system. Section 5 includes the experiments. Section 6 provides the conclusion and future work.

2 Related work

2.1 Carpooling

Huang et al. [8] presented an intelligent carpool system based on the service-oriented architecture and a fuzzy-controlled genetic-based carpool algorithm by using the combined approach of the genetic algorithm and the fuzzy control system, with which to optimize the route and match the vehicles and the passengers in the intelligent carpool system. In the same year, Huang et al. [8] also proposed an advanced carpool system called the intelligent carpool system (ICS), which provides passengers the use of the carpool services via a mobile phone anywhere any time. The carpool service agency in the ICS is integrated with the abundant geographical, traffic, and societal information and used to manage requests.

Plezer et al. [19] proposed a method that aims to best utilize carpooling potential while keeping detours below a specific limit. The method specifically targets carpooling systems on a very large scale and with a high degree of dynamics, and the road network is divided into distinct partitions which define the search space for ride matches. This allows optimizing the partitioning with regard to sharing potential utilization and inconvenience minimization.

Duan et al. [5] focused on removing the static capacity constraint and proposed a greedy approach based on multi-round matching. It allows a vehicle to carry more passengers than the vehicle's capacity, which is possible if some people are dropped off, and new passengers take their places during the journey.

Ma et al. [16] presented path optimization models and algorithms of taxi carpooling based on the single improved objective and multiple-objective genetic algorithm, and it could quickly get the taxi carpooling path and increase the income of taxi drivers while reducing the cost for passengers.

Qiang et al. [20] used the data field energy function to calculate the field energy of each data point in the passenger taxi off-point dataset. They proposed a clustering algorithm for urban taxi carpooling based on data field energy and point spacing.

Jindal et al. [10] proposed a reinforcement learning based system to learn an effective policy for carpooling that maximizes transportation efficiency so that fewer cars are required to fulfill the given amount of trip demand. AN et al. [1] modeled the carpooling problem as a Markov Decision Process problem and introduced Deep Deterministic Policy Gradient, a state-of-the-art reinforcement learning framework. In this model, a new reward method called picking, and parking bonus is proposed to solve an imbalance problem of shared cars in spatial distribution. In the process of driving, the environmental state space of surrounding vehicles is also complex, so DQN can be used to generate specific travel routes.

2.2 Reinforcement learning in routing planning

Zolfpour et al. [33] proposed a path planning system model based on the multi-agent reinforcement learning algorithm to solve the problem of vehicle delay. In this model, a

combination of Q-value based dynamic programming and Boltzmann distribution is adopted to create a priority route plan for vehicles by studying the weights of weather, traffic, road safety, fuel capacity, and other factors in the road network environment.

Due to the problem that the actual delivery routes of most orders in online food delivery are not consistent with the recommended routes, Liu et al. [15] proposed a deep inverse reinforcement learning algorithm to capture the preferences of food deliverymen from historical GPS tracks and recommend their preferred delivery routes. Considering the features of food delivery routes, the model uses Dijkstra's algorithm instead of value iteration to determine the current policy.

Dynamic taxi route recommendations are crucial to ease passenger wait times and increase taxi drivers' income. Therefore, Ji et al. [9] studied the dynamic taxi route recommendation problem as a sequential decision problem and presented an adaptive deep reinforcement learning method. The deep policy network in this method can better fuse the extracted spatial-temporal features to realize effective route recommendations.

Koh et al. [11] presented a deep reinforcement learning method to enable real-time interaction between vehicles and complex urban environments. By defining tasks as a sequence of decisions, a real-time intelligent vehicle routing and navigation system is constructed.

Wang et al. [26] modeled the carpooling vehicle planning problem as a Markov decision process and proposed a learning solution based on deep Q-network and action search to optimize the scheduling policy for drivers on the carpooling platform. To improve the adaptability and efficiency of learning, a new transfer method, correlation feature progressive transfer, is used in this method which realizes the transfer of knowledge in spatial and temporal space.

2.3 Privacy in carpooling and routing planning

Current online route planning services require the precise location of users [32]. However, due to privacy concerns, some users may be reluctant to reveal their current locations and destination locations. Providing a false location to an existing service or not providing a location can result in poor service quality or failure to provide a service. Vicente et al. [24] proposed a solution that can return accurate path planning results when using source and destination areas to achieve privacy-preserving.

Collaborative route planning optimizes vehicle routing by collecting data on planned routes from connected vehicles at the cost of increasing privacy risks for participating users. The current location, destination, and route of passengers and drivers are all highly sensitive information, so Florian et al. [6] demonstrated how collaborative routing planning could be implemented primarily by the anonymous fashion of publication intentions with strong privacy guarantees, without significantly reducing utility or cost.

To reduce the cruising time of ride-hailing vehicles and improve the efficiency and benefit, Shi et al. [22] proposed a route planning method for ride-hailing based on deep reinforcement learning. Considering the location of ride-hailing vehicles, different time periods of the day, competition between ride-hailing vehicles, and other factors, the method enables the online ride-hailing service center to understand the dynamic service environment and plan the route of idle vehicles.

In the process of carpooling, fog computing raises new privacy issues while providing low-latency local data processing, in which users' personal information (such as identity and location) may be compromised. Therefore, Li et al. [14] proposed an efficient and privacy-preserving carpooling scheme that uses blockchain-assisted vehicle fog computing to support

conditional privacy, one-to-many matching, destination matching, and data auditability. The method uses a private proximity test to realize one-to-many proximity matching and extends it to effectively establish the secret communication key between passenger and driver.

3 Background

3.1 Route planning

Route planning is the calculation of the best possible path in a road network from a given source to a given target location. It is often used in daily life, such as people using it to plan car trips [21]. Many applications, such as logistics planning or traffic simulation, need to address many path queries and planning. With the rapid development of urban traffic networks, path planning service providers have to provide strong computing capacity to achieve accurate services to customers, which will also bring high costs to service providers [30]. For these reasons, researchers have considerable interest in developing more efficient and accurate route planning techniques. A road network in route planning can easily be represented as a graph, i.e., as a collection of nodes (junctions) and edges (road segments) where each edge connects two nodes. Each edge is assigned a weight (distance).

At present, most path planning algorithms aim at the shortest path, which can be divided into the following categories [2]:

- Basic technique. The standard solution to the one-to-all shortest path problem is Dijkstra's algorithm [3]. This algorithm preserves a transient distance array for each node, and accesses the nodes of the network in order of the distance from the nodes to the source node, and keeps the transient distance equal to the correct distance of the nodes. The algorithm stops when the target node is accessed. Dijkstra's algorithm is generally used as a baseline to evaluate the quality of other route planning algorithms.
- Goal-Directed technique. Dijkstra's algorithm scans all nodes whose distance is less than that of the source node to the target node. In contrast, the purpose of goal-directed techniques is to "direct" search to a target by avoiding scanning nodes that are not in the direction of the target. They make use of either the (geometric) embedding of the network or properties of the graph itself to accelerate query, such as the structure of the shortest path tree leading to the (compact) region of the graph.
- Separator-based technique. Planar graphs have small and computable separators, and although road networks are not planar (tunnels or overpasses), they have also been observed to have small separations. The separator-based technique uses this feature of the Planar graphs to divide the graph into smaller parts, which can then be used with smaller overlay graphs to accelerate (partial) query algorithms.
- Hierarchical technique. The hierarchical Technique takes advantage of the inherent layers of the road network so that the shortest paths long enough eventually converge into a small trunk network of essential roads, such as highways. Intuitively, when the query algorithm is far away from the source, and target nodes, only the nodes of the subnet can be scanned, which can also accelerate the query.
- Bounded-Hop technique. The bounded-Hop technique calculates the distance between nodes in advance, adding this hidden information to the graph first. Queries can then return the length of the virtual path with only a few hops, and this process can accelerate the query algorithms.

3.2 Deep reinforcement learning

Reinforcement learning [31] is an AI-based method in which an agent acts based on the feedback of the environment, through continuous interaction with the environment in a trial and error manner, and finally accomplishes a specific purpose or maximizes the revenue.

It is a process in which an agent takes action to change its state and get rewards to interact with the environment. At the beginning of this process, the agent takes random actions first in the initial state because in the absence of any additional information and strategies in the initial state, the agent can only randomly select an action first to explore the environment. With the exploration of the environment, the agent will take the actions with the highest estimated value and maximum reward to approach the final goal [23].

Reinforcement learning can be divided into model-based learning methods and model-free learning methods. In model-based reinforcement learning, the transition of the state of the environment must be known, that is, model-based. However, in model-free reinforcement learning [29], agents are used to constantly explore the environment, constantly try, make mistakes, and learn. Therefore, the data efficiency of the model-free method is not high. Model-based approaches, by comparison, can make full use of existing models and make efficient use of data. In model-free reinforcement learning, the most widely used learning approach is Q-learning. The main idea of this algorithm is to define the Q function (performance function), substitute the observed online data into the following update formula for iterative learning of the Q function, and get the exact solution and save the record of the value function in the form of a table.

$$Q(s', a) \leftarrow (1 - \alpha) \cdot Q(s, a) + \alpha \cdot (r + \gamma \cdot \max_{a'} Q(s', a')) \quad (1)$$

Where α is the learning rate and γ is the discount factor. The greater the α , the less effective the retention of previous training is. $Q(s', a)$ is Q value of action a on state S' , and r is reward. Q learning uses a reasonable strategy to generate the action, according to the action interaction with the environment to get the next state and reward to learn to acquire optimal Q function.

Controlling agents directly by learning higher-dimensional perceptual inputs (images, speech, etc.) is a long-term challenge for reinforcement learning [28]. The quality of reinforcement learning results seriously depends on the quality of feature selection [12]. The development of deep learning makes it possible to extract high-level features from original data directly. Deep learning has strong perception ability but lacks specific decision-making ability. Reinforcement learning has the ability to make decisions but can do nothing about the perception problem.

Therefore, the combination of the two complementary advantages provides a solution for the problem of perception decisions in complex systems. Deep Reinforcement Learning (DRL) is an algorithm that combines deep learning with reinforcement learning and realizes end-to-end learning from perception to action. In the early stage of deep reinforcement learning research, the research mainly focused on dimension reduction of high-dimensional input data by the deep neural network. Lange et al. [13] proposed a Deep Auto-Encoder (DAE) model by combining the deep learning model and the reinforcement learning method. However, DAE is only applicable to control problems with visual perception as input signal and small dimension of state space. Mnih et al. [17] combined the convolution neural network with the Q-learning algorithm and proposed the Deep Q-Network (DQN) model. This model is used to deal with control tasks with huge state space and is a pioneering work in the field of DRL.

4 Problem definition

4.1 Privacy problem and definition

Different from the classic taxi, passengers in taxi carpooling are strangers to each other. It has great privacy concerns. As is shown in Figure 1, if passenger A gets on the vehicle first, then passenger B gets on and gets off the vehicle before passenger A gets off, then the starting point and destination of passenger B are exposed to passenger A, which is a serious privacy problem. Therefore, we define a passenger’s starting point and destination together as a passenger’s privacy, and a passenger’s starting point and destination are both exposed to another passenger as privacy leakage. This includes passengers who have the same starting point or destination. This seems to be an unavoidable problem in carpooling, and it’s one of the reasons why so many people don’t choose to carpool. However, suppose passenger A gets on first. In that case, passenger B gets on, then passenger A gets off, then passenger B gets off, in which case passenger A only knows the starting point of passenger B. Passenger B only knows the destination of passenger A, then privacy issues can be minimized, which is shown in Figure 2. At the same time, we can see the distance of the vehicle in Figure 2 is longer than in Figure 1, leading to a longer time.

However, taking actual traffic conditions into account, as shown in Figure 3, an intersection in Figure 1 route is a red light, and another intersection in Figure 2 route is a green light. Although the distance of route 2 is longer, it takes less time to complete route 2, and it also can protect the privacy of two passengers. In more complex traffic conditions with more passengers and vehicles, more routes look like a detour but actually get to the destination faster. This is how we protect passenger privacy by not allowing one passenger’s whole itinerary to be included in another.

4.2 Traffic information

Traffic information refers to the number of vehicles in the surrounding environment of the agent, which can be transmitted to each other by other agents or directly to the central decision-making system. As shown in Figure 4, red vehicles are our target vehicles,

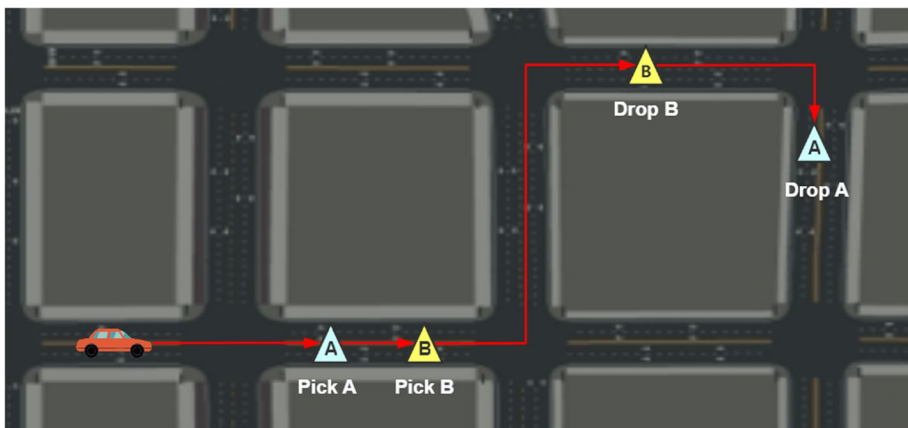


Fig. 1 Taxi picking up route without privacy concern

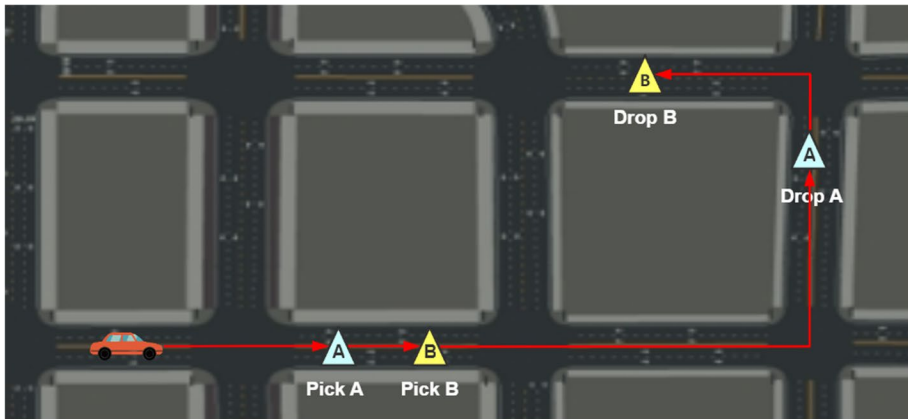


Fig. 2 Taxi picking up route with privacy and detour concern

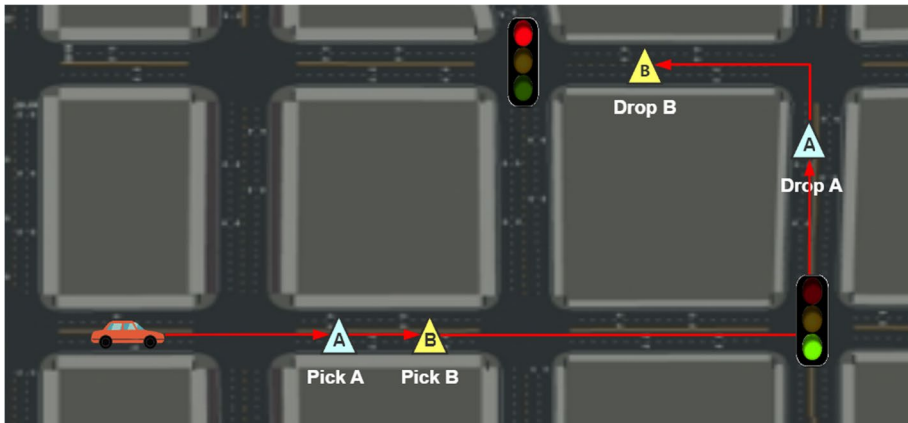


Fig. 3 Taxi picking up route with privacy concern, detour and traffic signal concern

and the surrounding environmental vehicles can be divided into several different parts as the state space of the target vehicles. For example, the intersection in the red box in Figure 4 as the center can be used as the first part of the state space of the target vehicle, in which there are two cars.

5 Time-optimal and privacy-preserving carpool route planning system

Our Time-Optimal and Privacy-Preserving carpool route planning system is divided into four steps: Distribution of passengers, Task planning, Generate the route, Estimate the time and choose the optimal route. Figure 5 shows the sequences of those steps. Before we present the details of those steps, we formally define the route planning model.

Fig. 4 Traffic information around the agent

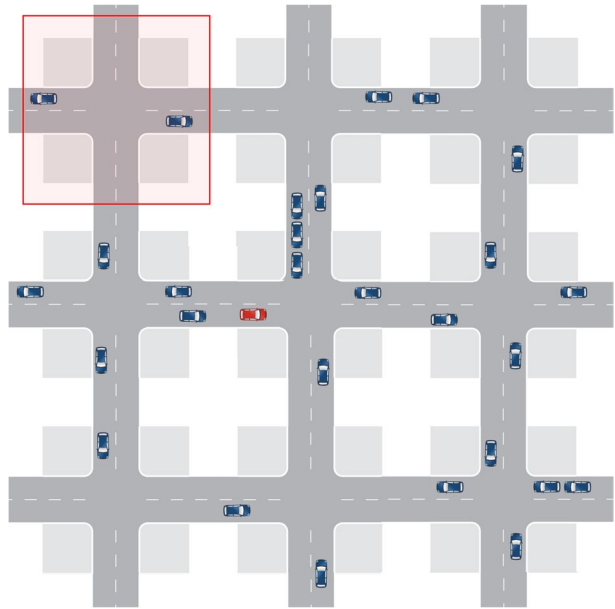
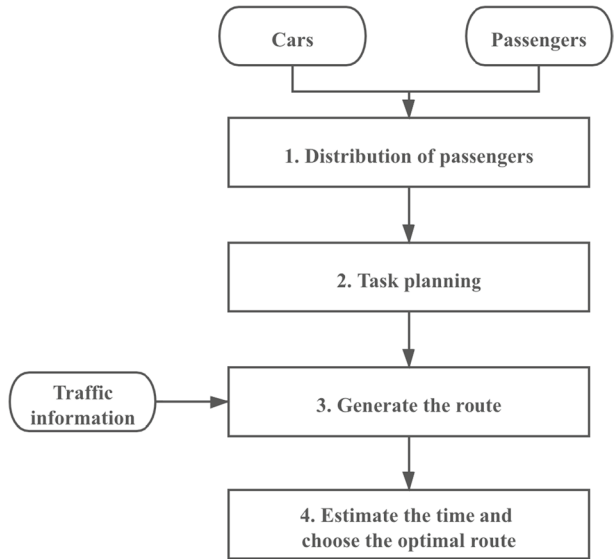


Fig. 5 Overview of Time-Optimal and Privacy-Preserving carpool route planning system



5.1 Model definition

In our model, passengers are defined as a set of people $P = \{ p_1, p_2, p_3, \dots, p_n \}$, each passenger i has a starting point S_i , and a destination point D_i . Both the starting point and the destination point contain horizontal and vertical coordinates (latitude and longitude coordinates).

Taxis are defined as a set of vehicles $V = \{v_1, v_2, v_3, \dots, v_m\}$, each vehicle i has the current location l_i , maximum seating capacity M_i and passenger matching list $m_i = (p_1, p_2, \dots, p_o)$. This means vehicle i is responsible for picking up these passengers and o is equal or less than M_i . Each vehicle also has an action task list $T_i = (Pick_{p_1}, Pick_{p_2}, \dots, Pick_{p_o}, Drop_{p_1}, Drop_{p_2}, \dots, Drop_{p_o})$, such as $T_1 = (Pick_{p_1}, Pick_{p_2}, Drop_{p_1}, Drop_{p_2})$, it means the action task for vehicle 1 is to pick up passenger 1 first, then pick up passenger 2, after that drop off passenger 1, finally, drop off passenger 2. Each item in action task list is also the stop point for the vehicle.

When the tasks of each vehicle are combined without conflict (A passenger can only be picked up and dropped off by one vehicle) and can be responsible for the completion of all passenger pickup is called a possible solution for this group of passengers and a group of vehicles. Each action task of vehicle i has the corresponding sum of the Manhattan distance $ManD_i$, it is the Manhattan distance between the latitude and longitude coordinates of each adjacent stop point in the task. The goal of our model is to find a solution that minimizes the total estimated travel time.

5.2 Distribution of passengers

According to the given P , V and M_i for each vehicle, all combinations of vehicles and passengers are enumerated. Meanwhile, to satisfy the privacy-preserving requirement in Section 4.1, the combination of the same destination or starting point (two points are considered identical if the distance between them is less than d_s) of passengers in the same vehicle is filtered out. One of these combinations is:

$$\begin{aligned} v_1 &: (p_1, \dots, p_q) \\ v_2 &: (p_q + 1, \dots, p_w) \\ &\dots \\ v_m &: (p_w + 1, \dots, p_n) \end{aligned}$$

For example, one of the combinations in 3 vehicles and 6 passengers task:

$$\begin{aligned} v_1 &: (p_1, p_2) \\ v_2 &: (p_3, p_4) \\ v_3 &: (p_5, p_6) \end{aligned}$$

It means vehicle 1 is responsible for picking up passengers 1 and 2, vehicle 2 is responsible for picking up passengers 3 and 4, and vehicle 3 is responsible for picking up passengers 5 and 6.

5.3 Task planning

Specific action tasks are planned for each combination of vehicle and passenger in each possible solution, such as $v_1 : (p_1, p_2)$. Specific tasks for vehicle 1 could be: $t_1 = [Pick_{p_1}, Pick_{p_2}, Drop_{p_2}, Drop_{p_1}]$ However, in this specific task, starting point and destination passenger 2 are both exposed to passenger 1. According to Section 4.1 privacy definition, this task does not satisfy the privacy-preserving requirement. Cross all the passengers itineraries in the task, so only one of a passenger's destination and starting point is exposed to other passengers. In this way, privacy between

passengers is preserved to a certain extent. Therefore, specific tasks for vehicle 1 can be: $t_2 = [Pick_{p1}, Pick_{p2}, Drop_{p1}, Drop_{p2}]$ or $t_3 = [Pick_{p2}, Pick_{p1}, Drop_{p2}, Drop_{p1}]$

For each solution, every vehicle could enumerate specific tasks, the number of all possible combinations is overmuch, and the subsequent calculation cost is too large. Many of these possible tasks actually took a long detour, so we calculated the Manhattan distance $ManD_i$ of the specific action task of i in each solution.

The total distance of each solution is sorted from small to large, and the first k solutions are screened out according to the sorting results, and then the subsequent calculation for these screened solutions was carried out.

5.4 Generate the route

In this step, we use the trained DQN to generate the route that takes the least time for the specific task of each vehicle in the first k solutions. In these DQNs, the state is the current position of the vehicle and the traffic condition information in the current area, the reward is the time, and the action is the direction the vehicle chooses at the intersection. The traffic condition information here refers to the number of vehicles in different directions at each intersection. Well-trained DQNs can learn the change of traffic flow and traffic lights through the traffic information and travel time to take action at the intersection.

For example, in the specific task of vehicle 1 $[Pick_{p1}, Pick_{p2}, Drop_{p1}, Drop_{p2}]$, the first part of the task is from the starting point of the vehicle to the position of Passenger 1. The traffic data that also passes through these two points is searched in the data set, and a DQN is trained with these data. The surrounding traffic information of vehicle 1 is used as input, trained DQN will generate specific actions, which will be repeated many times until the stop point is reached, and then a series of actions will be taken as the specific route of this journey. Repeat this step to get the specific route of all the journeys of vehicle 1.

5.5 Estimate the time and choose the optimal route

We use our surrounding traffic sub-path (ST-SP) time estimation to estimate the required travel time for the specific routes of the k solutions in the last step. The time estimator estimates the required travel time based on the starting point, destination, route, and traffic information of the vehicle. It is based on a large number of vehicle traffic data.

The route of each vehicle is divided into small parts according to the stopping point. The vehicle location and vehicle information of the first part are firstly used to look for similar vehicle traffic data in the data set, and then the estimated traffic time of the first part is estimated according to the mean of the traffic time of the similar data. Then use the traffic information from the similar data after arriving at the destination as the starting point traffic information for the next part of the route, and repeat these steps until the travel time for all parts of the route has been estimated. Finally, all route travel time for each solution can be estimated

According to the estimated time of each part of the solution, calculate the average travel time for passengers in each solution. The shortest average travel time is chosen as the optimal one, and the output is the specific task and the route for each vehicle.

5.6 Privacy analysis

The passenger's privacy is defined as his both starting point and destination together in the itinerary. Privacy is guaranteed if the passenger cannot acquire other passengers' both starting point and destination in the same vehicle. In our proposed Time-Optimal and Privacy-Preserving carpool route planning system, the following two cases of privacy leak are filtered out.

Two passengers in a vehicle have the same starting point or destination. In this case, one passenger can acquire the other's both starting point and destination because their starting point or destination is the same, and they overlap in the carpooling service. We filter out the combination of the passengers who have the same destination or starting point (two points are considered identical if the distance between them is less than d_s) in step 1 distribution of passengers. One passenger's starting point and destination are in another passenger's itinerary. In this case, the passenger with the longer trip can acquire both the starting point and destination of another passenger with the shorter trip. We cross all the passengers' itineraries in step 2 task planning to avoid this case. Therefore, our method can completely preserve passengers' privacy in carpooling service.

6 Experiments

6.1 Experiment setup

We applied our ST-SP time estimation and TOPP carpool route planning system to CARLA [4], which is one of the commonly used simulators for the research of the internet of vehicles (IOV).

The simulation platform provides 3D models of various static objects, such as buildings, grass plants, traffic signs, and infrastructure, as well as dynamic objects, such as vehicles and pedestrians. And these objects can be created under control, which makes the experimental environment more like the real world. Here we use the map Town05 in the CARLA platform (as shown in Figure 6) as the experimental environment because the layout of the map is relatively complex, with multiple traffic lights at crossroads, and the map is similar to the layout of many real city CBDs, which can better simulate the real world.

6.2 Time estimation

We select a starting point, a destination, and a specific route on this map, and then control a vehicle to drive at a constant speed from the starting point, arrive at the destination, calculate the real travel time, and compare it with the time estimated by our method. In the experiment, the starting point and destination, as well as the specific route connecting the two points, were randomly selected. The total distance was about 400-600m, and each route contained 4-6 intersections. We adjusted the complexity of the experimental environment by adjusting the total number of vehicles on the map. The more the total number of vehicles, the longer the waiting time at the traffic light intersection will be, and the more serious the traffic jam will be.



Fig. 6 Top View of Map Town05 in CARLA

1) *Evaluation method*: Here, we list the travel time estimation methods compared to ST-SP:

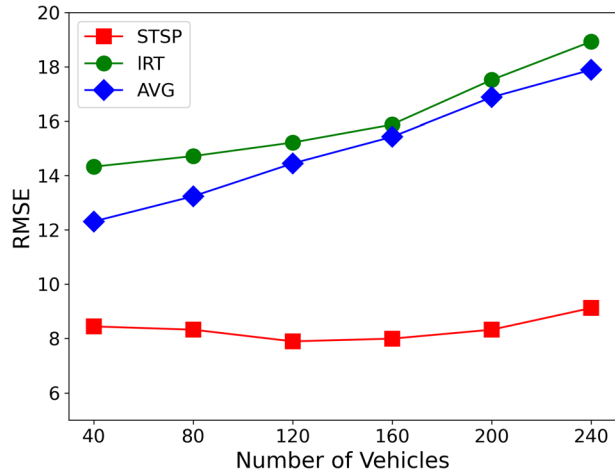
- a. MLR, We performed a multiple linear regression based on the route distance and the total number of surrounding vehicles to predict the time.
- b. AVG. This method finds the same route as the target route in the data set to calculate the average time.
- c. ST-SP. This is our method proposed in Section 5.5.

2) *Evaluation measures*: We use *Root Mean Squared Error (RMSE)* to evaluate the performance of these three methods. *RMSE* is the square root of the mean of the square of all of the difference between estimated travel time f_i and real-time y_i .

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n \left(\frac{y_i - f_i}{\sigma_i} \right)^2} \quad (2)$$

Figure 7 compares the performance of the proposed method for time estimation. From this figure, we observe that our proposed method is far better than the other two methods. Although the method of multiple linear regression considers the two most critical factors, traffic condition, and distance, compared with our method, there is a big gap, and the error increases with the increase of the complexity of the environment. Another approach AVG is to find historically identical routes in the data set to estimate the time. This approach does not consider traffic conditions, so as traffic conditions become more complex as the total number of vehicles on the map increases, RMSE becomes large. Our method makes full use of similar routes and traffic conditions to estimate the time, so RMSE does not fluctuate greatly when the total number of vehicles on the map increases and the traffic conditions become complicated.

Fig. 7 RMSE performance comparison of ST-SP with the other approaches for time estimation



6.3 Carpooling

Firstly, we need to consider the value of parameter k in the proposed method. Parameter k is the task planning in Section 4.2, and first k solutions are screened out according to the sorting results by Manhattan distance. The k is very important for the whole method. The method used in the previous generation of the solution is enumerated. If all solutions are not further filtered, the computation cost for the subsequent generation of routes will be huge. If the value of k is too large, it will increase the computation cost; if it is too small, it will affect the final result (the optimal solution is screened out). Therefore, we need to determine the value of k in different environments by experiment.

The complexity of the task varies with the relative value of k . Therefore, we conducted experiments on our proposed method without privacy-preserving mechanism (TO) in the task of 2 vehicles matching 5 passengers and the task of 3 vehicles matching 7 passengers, respectively. The number of other vehicles in the surrounding environment in both tasks was 120.

From the task results of 2 vehicles matching 5 passengers in Figure 8(a), we can see that the time converges rapidly with the increase of k value, the possibility of the optimal solution among the first k solutions is also increasing. At $k = 8$, the time converges almost completely. When k is greater than 8, there is a slight decrease in time, but the change is very small. Therefore, in the case of considering the computation cost and the optimal result of time, the value of k is 8 in the task of 2 vehicles matching 5 passengers.

As can be seen from the task of 3 vehicles matching 7 passengers in another Figure 8(b), with the increase of k value, the time converges quickly and completely when $k = 13$. Therefore, the value of k is 13 in the task of 3 vehicles matching 7 passengers, also taking into account the computation cost and the optimal result of time. With the privacy-preserving mechanism, the number of all enumerated solutions will also be reduced. Therefore, TO-PP converges faster than TO with the increase of k value in the same task. However, for the purpose of fairer comparison in subsequent experiments, we still adopted the same k value for the two methods (TO and TO-PP).

After determining the value of k , we then conduct carpooling experiments to compare with other methods. Here we list the methods compared to TO-PP.

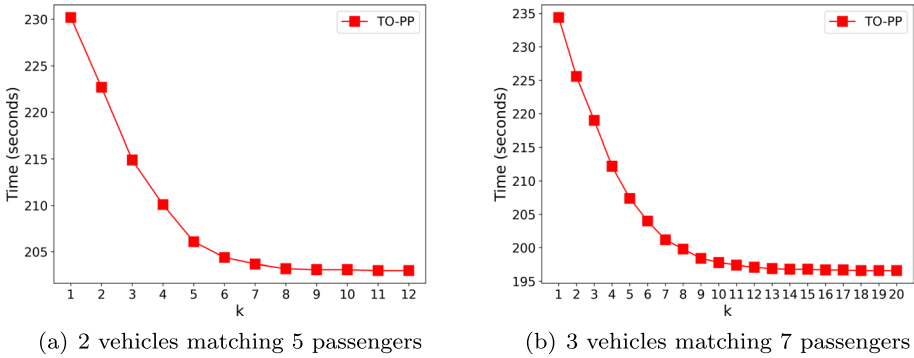


Fig. 8 Time performance of TO with different k in the task of 2 vehicles matching 5 passengers (a) and task of 3 vehicles matching 7 passengers (b)

- a. TO, This is the method without privacy preserving proposed in this paper.
- b. TO-PP. This is the method with privacy preserving proposed in this paper.
- c. LNM. Location nearest match, this method is to match vehicles and passengers according to their location, and the specific route planning is generated by path planning system in CARLA itself.
- d. IPMA. This is a greedy approach based on multi-round matching [5], and it is modified for implement on CARLA, and the specific route planning is generated by path planning system in CARLA itself.
- e. D-DQN. This is a distributed deep Q-network-based route scheduling algorithm [22], and it is also modified for implement on CARLA.

These experimental methods were tested in four different tasks, the first task was 2 vehicles matching 4 passengers, the second task was 2 vehicles matching 5 passengers, the third task was 3 vehicles matching 6 passengers, and the fourth task was 3 vehicles matching 7 passengers. The tasks get more complicated, and the four tasks also cover the most common real-time carpooling situation in real life.

The results of the four methods to complete the first task are shown in Figure 9(a). As can be seen overall, with the increase of the number of surrounding vehicles, the average

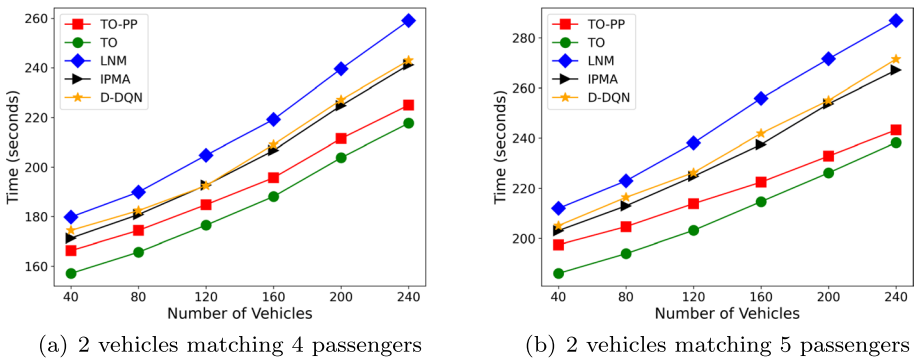


Fig. 9 Time performance comparison of TO-PP with the other methods for carpooling task of 2 vehicles matching 4 passengers (a) and task of 2 vehicles matching 5 passengers (b)

time required to complete the task by the four methods also increases. The increase in the number of surrounding vehicles makes the environment more prone to traffic congestion. Among the four methods, TO has the best performance, followed by TO-PP, IPMA, D-DQN, and LNM. Similarly, as the environment becomes more complex, the gap between TO-PP and TO becomes smaller and smaller, while the gap between TO-PP and TO becomes larger and larger when compared with IPMA, D-DQN, and LNM, indicating that TO-PP and TO have better ability to adapt to the complex environment.

As shown in Figure 9(b), the results of the second task are similar to the results of the first task in terms of the overall trend, except that the time required for all methods increases as the task becomes more difficult. Compared to the first task, the number of carpooling vehicles stayed the same, and there was one more passenger in the second task. When there are only 40 vehicles around in this environment, the difference between the results of our method on the two tasks is 30 seconds, while when there are 240 vehicles around, the difference between the results of our method on the two tasks is only 20 seconds, which also shows that our method performs better in complex environment and tasks.

The third and fourth tasks' results are shown in Figure 10(a) and (b), respectively. The overall results are also similar to the previous two figures. When the number of environment vehicles is larger than 160 for LNM, D-DQN, and IPMA, the required time per passenger increases rapidly, while the time increases slowly in our two methods. When the number of environment vehicles increases, the difference between TO and TO-PP becomes smaller and smaller, when the number of environment vehicles is 240, the difference between these two results is less than 5 seconds.

Overall, TO has the best performance. The gap between TO and TO-PP becomes smaller and smaller as tasks and environments become more complex, and the performance of our method will be better in complex environments.

7 Conclusion

In this paper, we discuss the problems of commuting time optimization and passenger privacy-preserving by the existing carpooling method. We develop a time-optimal and privacy-preserving carpool route planning system via deep reinforcement learning. This system adopts the traffic information around the carpooling vehicle to optimize the commuting

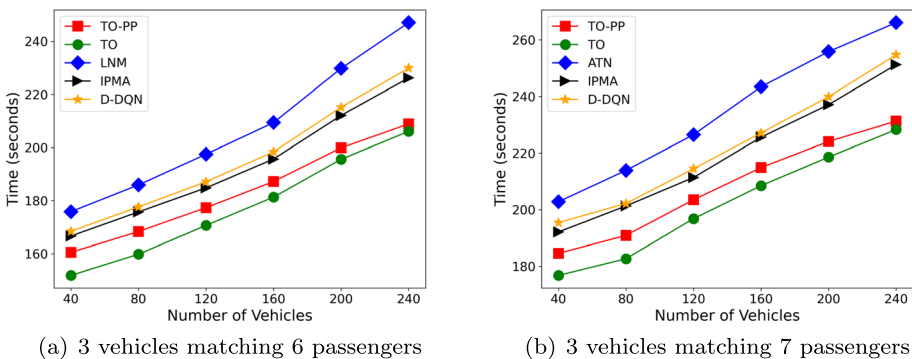


Fig. 10 Time performance comparison of TO-PP with the other methods for carpooling task of 3 vehicles matching 6 passengers (a) and task of 3 vehicles matching 7 passengers (b)

time of passengers. The aim of the system is to efficiently match passengers and vehicles and to generate detailed route planning for carpooling vehicles. To evaluate our methods, we conducted the experiment on IOV simulators CARLA, and the results demonstrate the advantages of our proposed methods over other carpooling algorithms, especially in complex environments and tasks. The idea of passing through multiple stops and using surrounding traffic information to optimize the route can also be used for food delivery and express delivery, which can be studied as future work.

Acknowledgements This work is supported by Australian Research Council Linkage LP180101150.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions.

Compliance with ethical standards

Conflicts of interest The authors have no conflicts of interest to declare that are relevant to the content of this article.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. An, L., Ren, C., Gu, Z., Wang, Y., Gao, Y.: Rebalancing the car-sharing system: A reinforcement learning method. In: 2019 IEEE Fourth International Conference on Data Science in Cyberspace (DSC), pp. 62–69. IEEE (2019)
2. Bast, H., Delling, D., Goldberg, A., Müller-Hannemann, M., Pajor, T., Sanders, P., Wagner, D., Werneck, R.F.: Route planning in transportation networks. In: Algorithm engineering, pp. 19–80. Springer (2016)
3. Dijkstra, E.W., et al.: A note on two problems in connexion with graphs. *Numerische Mathematik* **1**(1), 269–271 (1959)
4. Dosovitskiy, A., Ros, G., Codevilla, F., Lopez, A., Koltun, V.: CARLA: An open urban driving simulator. In: Proceedings of the 1st Annual Conference on Robot Learning, pp. 1–16 (2017)
5. Duan, Y., Mosharraf, T., Wu, J., Zheng, H.: Optimizing carpool scheduling algorithm through partition merging. In: 2018 IEEE International Conference on Communications (ICC), pp. 1–6. IEEE (2018)
6. Florian, M., Finster, S., Baumgart, I.: Privacy-preserving cooperative route planning. *IEEE Internet of Things Journal* **1**(6), 590–599 (2014)
7. Gärling, T.: Effectiveness, public acceptance, and political feasibility of coercive measures for reducing car traffic. In: Threats From Car Traffic to the Quality of Urban Life. Emerald Group Publishing Limited (2007)
8. Huang, S.C., Jiau, M.K., Lin, C.H.: A genetic-algorithm-based approach to solve carpool service problems in cloud computing. *IEEE Transactions on Intelligent Transportation Systems* **16**(1), 352–364 (2014)
9. Ji, S., Wang, Z., Li, T., Zheng, Y.: Spatio-temporal feature fusion for dynamic taxi route recommendation via deep reinforcement learning. *Knowledge-Based Systems* **205**, 106302 (2020)
10. Jindal, I., Qin, Z.T., Chen, X., Nokleby, M., Ye, J.: Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining. In: 2018 IEEE International Conference on Big Data (Big Data), pp. 1417–1426. IEEE (2018)
11. Koh, S., Zhou, B., Fang, H., Yang, P., Yang, Z., Yang, Q., Guan, L., Ji, Z.: Real-time deep reinforcement learning based vehicle navigation. *Applied Soft Computing* **96**, 106694 (2020)

12. Kong, Y., Zhang, M., Ye, D.: An auction-based approach for group task allocation in an open network environment. *The Computer Journal* **59**(3), 403–422 (2016)
13. Lange, S., Riedmiller, M.: Deep auto-encoder neural networks in reinforcement learning. In: *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE (2010)
14. Li, M., Zhu, L., Lin, X.: Efficient and privacy-preserving carpooling using blockchain-assisted vehicular fog computing. *IEEE Internet of Things Journal* **6**(3), 4573–4584 (2018)
15. Liu, S., Jiang, H., Chen, S., Ye, J., He, R., Sun, Z.: Integrating dijkstra's algorithm into deep inverse reinforcement learning for food delivery route planning. *Transportation Research Part E: Logistics and Transportation Review* **142**, 102070 (2020)
16. Ma, C., He, R., Zhang, W.: Path optimization of taxi carpooling. *PLoS One* **13**(8), e0203221 (2018)
17. Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M.: Playing atari with deep reinforcement learning. *arXiv:1312.5602* (2013)
18. Olsson, L.E., Maier, R., Friman, M.: Why do they ride with others? meta-analysis of factors influencing travelers to carpool. *Sustainability* **11**(8), 2414 (2019)
19. Pelzer, D., Xiao, J., Zehe, D., Lees, M.H., Knoll, A.C., Aydt, H.: A partition-based match making algorithm for dynamic ridesharing. *IEEE Transactions on Intelligent Transportation Systems* **16**(5), 2587–2598 (2015)
20. Qiang, X., Shuang-Shuang, Y.: Clustering algorithm for urban taxi carpooling vehicle based on data field energy. *Journal of Advanced Transportation* **2018** (2018)
21. Schultes, D.: Route planning in road networks. In: *Ausgezeichnete Informatikdissertationen*, pp. 271–280 (2008)
22. Shi, D., Ding, J., Errapotu, S.M., Yue, H., Xu, W., Zhou, X., Pan, M.: Deep q -network-based route scheduling for tnc vehicles with passengers' location differential privacy. *IEEE Internet of Things Journal* **6**(5), 7681–7692 (2019)
23. Su, X., Zhang, M., Bai, Q., Ye, D.: A dynamic coordination approach for task allocation in disaster environments under spatial and communicational constraints. In: *Workshops at the Twenty-Eighth AAAI Conference on Artificial Intelligence* (2014)
24. Vicente, C.R., Assent, I., Jensen, C.S.: Effective privacy-preserving online route planning. In: *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, pp. 119–128. IEEE (2011)
25. Wang, M., Zhu, T., Zhang, T., Zhang, J., Yu, S., Zhou, W.: Security and privacy in 6g networks: New areas and new challenges. *Digital Communications and Networks* **6**(3), 281–291 (2020)
26. Wang, Z., Qin, Z., Tang, X., Ye, J., Zhu, H.: Deep reinforcement learning with knowledge transfer for online rides order dispatching. In: *2018 IEEE International Conference on Data Mining (ICDM)*, pp. 617–626. IEEE (2018)
27. Xiong, P., Li, G., Ren, W., Zhu, T.: Lopo: a location privacy preserving path optimization scheme for spatial crowdsourcing. *Journal of Ambient Intelligence and Humanized Computing*, 1–16 (2021)
28. Ye, D., Zhang, M., Sutanto, D.: Cloning, resource exchange, and relationadaptation: an integrative self-organisation mechanism in a distributed agent network. *IEEE Transactions on Parallel and Distributed Systems* **25**(4), 887–897 (2013)
29. Ye, D., Zhu, T., Cheng, Z., Zhou, W., Philip, S.Y.: Differential advising in multiagent reinforcement learning. *IEEE Transactions on Cybernetics* (2020)
30. Ye, D., Zhu, T., Shen, S., Zhou, W., Yu, P.: Differentially private multi-agent planning for logistic-like problems. *IEEE Transactions on Dependable and Secure Computing* (2020)
31. Ye, D., Zhu, T., Zhou, W., Philip, S.Y.: Differentially private malicious agent avoidance in multiagent advising learning. *IEEE Transactions on Cybernetics* **50**(10), 4214–4227 (2019)
32. Zhu, T., Ye, D., Wang, W., Zhou, W., Yu, P.S.: More than privacy: Applying differential privacy in key areas of artificial intelligence. *arXiv:2008.01916* (2020)
33. Zolfpour-Arokhlo, M., Selamat, A., Hashim, S.Z.M., Afkhami, H.: Modeling of route planning system based on q value-based dynamic programming with multi-agent reinforcement learning algorithms. *Engineering Applications of Artificial Intelligence* **29**, 163–177 (2014)