2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Turn-by-turn Intelligent Manoeuvring of Driverless Taxis: A Recursive Value Model Enhanced by Reinforcement Learning

Bo Yang[1] and Qianxiao Li[2]

Abstract— **We develop a protocol for constructing intelligent manoeuvring of driverless taxis - especially for empty taxis - with the aim of optimising the efficiency of the taxi system (e.g. minimising the average waiting time of the commuters). The manoeuvring is formulated as a stochastic policy matrix giving optimal choices on which way to go for taxis at any locations, based on the historical patterns of commuter origin and destination distribution. A recursive value (RV) model is formulated for turn-by-turn navigation. The policy matrix is solved by a combination of the RV model and reinforcement learning, together with a comprehensive agent-based microscopic simulation platform we have developed. We show that even for very large road networks, very efficient manoeuvring algorithms can be readily found with reinforcement learning, especially with our RV model as the initial state. Such algorithms are indispensable for the fleet of driverless taxis, and can also give helpful recommendations to human drivers operating taxis. The general framework of our protocol allows the construction of optimal manoeuvring algorithms for a wide range of road networks in a systematic way.**

I. INTRODUCTION

Bridging services such as taxis and shuttles play an important role for balancing the need for comfort, flexibility, and affordability of the transport service in modern cities, enhancing the overall efficiency of urban traffic system[1], [2] that also includes the public transportation and private vehicles. In most cities, taxi is the dominant mode of bridging services with the ability to retain most, if not all, of the benefits of owning a private vehicle [3], [4]. On the other hand, it is also important to note that at the system level, increasing the market share of taxi services can have adverse effects in large and crowded cities. For commuters to choose taxis over public transportation, it generally results in more trips on the road, more energy consumption and green house gas emissions at similar levels as private cars[5]. One should also note that this is true when (potential) car-owners choose taxis over private vehicles, especially if taxi ride-sharing is not a common option. If we replace one trip with a personal vehicle from home to office by a taxi ride, it can lead to additional pick-up trips and effectively more vehicles on the road (i.e. taxis that ply on the road without commuters consume road space, and thus contributing to congestions)[6].

There is thus much room for improving the efficiency of the taxi services, either via the introduction of ride-sharing, or via the development of better routing strategies for the taxis. In this paper, we focus on the latter. The routing of occupied taxis is relatively well studied, with efficient algorithms picking the fastest route from the origin to the destination, depending on the traffic conditions and other constraints. The movement of empty taxis, on the other hand, is a more complicated issue, even with the recent popularity of telephone or App booking. When an empty, un-booked taxi roams the street looking for potential commuters, the efficiency highly depends on the experience of the taxi driver. This is also the part of the journey that is "wasted", and should be minimised as much as possible with minimal impact on the overall service quality of the entire taxi fleet.

There are a number of works in literature on modelling taxi systems[7], [8], [9], [10], understanding the dynamics between taxis and commuters[11], [12], as well as customer searching and demand predictions[13], [14], [15], [16], [17], [18], [19]. In this paper, we employ systematic and novel approaches to develop optimal turn-by-turn navigation algorithm for empty taxis searching for potential commuters. Thus in principle, existing algorithms and approaches form a subset of a much larger action space we search over for the optimal navigation. The methodology consists of microscopic agent-based modelling, crafted rule-based algorithms and reinforcement learning[20], [21]. Based on the historical trip distribution of the commuters throughout the entire road network[6], the output of our algorithm will inform an empty taxi at every time step which action to take among all possible choices (e.g. go straight or turn left), depending on the time and location of this empty taxi. Each empty taxi will make its own decision based on the same algorithm, though given the same time and location, the output choice is stochastic following some optimal probabilistic distribution. While in principle the navigation algorithm can be developed for dynamic road network with evolving traffic conditions, in this paper we focus on a static road network, in which the traffic condition can be fully accounted for, though is assumed to be constant in time.

The algorithm we developed can be easily implemented on-board as a recommendation system for taxi driver, and we will study the effect of compliance rate for human drivers (as sometimes they prefer to follow their own experience/intuition) in a separate work. More straightforward applications can be found in driverless taxis[22], which could go mainstream in the near future given the rapid development in autonomous driving technologies. One would expect shuttles and taxis to be the first in adopting driverless technologies en masse; this is because while driverless

[1]Bo Yang is with Computing Science Department, Institute of High Performance Computing, A*STAR, Singapore, 138632 yangbo@ihpc.a-star.edu.sg

[2]Qianxiao Li is with Computing Science Department, Institute of High Performance Computing, A*STAR, Singapore, 138632 liqix@ihpc.a-star.edu.sg

technologies could be just a luxury for private car owners, for taxi companies there are huge benefits in cost-saving. Our work could be especially useful in the long term, since for driverless taxis, a turn-by-turn navigation algorithm for empty taxis is a necessity.

The paper is organized as follows: In Sec. II we give a well-defined formulation of turn-by-turn manoeuvring as a mathematical problem; in Sec. III we introduced the microscopic agent-based simulation platform we have developed; in Sec. IV we show in details our recursive value model for turn-by-turn manoeuvring, and the reinforcement learning algorithm in Sec. V; in Sec. VI we report a number of evaluations from our numerical simulation platform, and in Sec. VII we summarise our results and some of the future works.

II. Problem Statement

Finding optimal turn-by-turn navigation can be formulated as a well-defined mathematical problem. The road network can be fully described by a graph denoted as $G(N, E)$. Here, N is the collection of nodes of the graph, physically corresponding to all the locations of a region (i.e. a city) where passengers can either board or alight a taxi; E is the collection of directed, weighted edges that can be represented by an adjacency matrix $\mathcal{A}$. For nodes i, j, there is an edge directed from i to j if and only if the taxi can drive from i to j without passing through any other nodes (so the j^{th} node is directly connected from i^{th} node). The weight is given by the time needed to drive from i to j, so the traffic condition is fully taken into consideration. Even for the same physical distance between the two nodes, the edge weight can be different due to speed limit or the average travel speed that depends on whether or not the road is congested. In our definition of the adjacency matrix, $\mathcal{A}_{ij} = 0$ if j is not directly connected from i; otherwise $\mathcal{A}_{ij}$ is the *inverse* of the travel time from i to j, for $i \neq j$. For $i = j$ we define $\mathcal{A}_{ii} = 1$.

Two additional arrays are used as inputs extracted from the synthetic data or historical empirical data: the vector g_i is the probability of a commuter appearing and waiting at the i^{th} node for each time step; the matrix M_{ij} is the probability that a commuter with origin at the i^{th} node is going to the destination at the j^{th} node. We thus have $\sum_i g_i = n$ and $\sum_j M_{ij} = 1$, where n is the total number of commuters generated in the entire road network per time step. In principle, $\mathcal{A}_{ij}, g_i, M_{ij}$ can all be time dependent; but in this paper we focus on the case where all these three quantities are stationary.

The algorithm we develop will be entirely encapsulated in the policy matrix $\mathcal{P}_{ij}$ for empty, unbooked taxis; physically it means if there is an empty taxi located at i, the probability of this taxi going in the direction of j (which is a neighbouring node) is given by $\mathcal{P}_{ij}$. Thus $\sum_j \mathcal{P}_{ij} = 1$, and if $\mathcal{A}_{ij} = 0, \mathcal{P}_{ij} = 0$. Note that $\mathcal{P}_{ii}$ can be non-zero, which gives the probability of the taxi staying put for one time step. The policy matrix from the solution should minimise some utility function, depending on the purpose of the problem at hand.

Inputs	Output
Network adjacency matrix $\mathcal{A}_{ij}$ Origin probability distribution g_i Destination probability distribution M_{ij}	Policy matrix $\mathcal{P}_{ij}$

TABLE I

Summary of inputs and outputs with notations.

In this paper, we choose the average waiting time of the commuters to be the utility function, which $\mathcal{P}_{ij}$ intends to minimise. The notations of the problem is summarised in Table. I.

III. The Simulation Platform

An agent-based, microscopic simulation platform plays the essential role of evaluating the performance of $\mathcal{P}_{ij}$, given different $\mathcal{A}_{ij}, g_i, M_{ij}$. It also generates a collection of instances containing relevant information for reinforcement learning. We have developed a highly versatile simulation platform[6] consisting of two types of agents: the taxis and the commuters. With this simulation platform, the performance of $\mathcal{P}_{ij}$ can be benchmarked by various indicators from different perspectives. These include the average commuter waiting time, average earning of the drivers, and the average fuel cost. In this paper we focus on the average commuter waiting time.

In the microscopic simulation, the commuters and the associated origin-destination (OD) pairs are generated stochastically according to g_i and M_{ij}, and the positions of the taxis are initiated at random nodes at the beginning of the simulation. There are three states for the taxis: empty, booked, and occupied. Two modes of commuter pickups can be implemented: real-time booking and road-side hailing. For the former, when a commuter is generated in the network, the empty taxi (if available) that is nearest to the commuter will be assigned to this commuter, and for this taxi the status will change from "empty" to "booked". The booked taxi will follow the fastest route from its current location to the origin of the assigned commuter. Once it reaches the origin, the commuter will be picked up and the taxi status will change from "booked" to "occupied". Following the fastest route from the origin to the destination of the commuter, the taxi status will change from "occupied" to "empty" once the commuter alights at the destination. The road-side hailing mode is more straightforward. Empty taxis will only pick up a commuter when the location of the taxi and the origin of the commuter coincides.

In both cases, the movement of "booked" or "occupied" taxis follows the corresponding fastest route, which could be computed efficiently using Dijkstra's algorithm. The movement of the empty taxis will be completely governed by $\mathcal{P}_{ij}$. The simplest policy is the random policy, given by $\mathcal{P}_{ij} = 1/n_i$ if $\mathcal{P}_{ij} \neq 0$. Here, n_i is the number of neighbours or $\mathcal{A}_{ij}$ that is not zero, including $j = i$.

The simulation is completely well-defined once $\mathcal{A}_{ij}, g_i, M_{ij}, \mathcal{P}_{ij}$ are given, and various statistical quantities can be calculated after running the simulation for a certain

period of time T, measured in terms of the number of time steps. Each time step can correspond to different physical time unit. Without loss of generality, we take each time step to be one second.

IV. Recursive Value Model

Crafting a good policy matrix by hand turns out to be highly non-trivial. For example, just letting the empty taxi to go to the neighbouring node j with the highest g_j will actually have worse performance as compared to a completely random policy in most road networks. We present here an effective approach for obtaining efficient policy matrix $\mathcal{P}_{ij}$, which will be validated in the following section. Given the well-defined formulation of the mathematical problem in Sec. II, with a specific input of a pair of g_i, M_{ij} on a specific road network fully described by $\mathcal{A}_{ij}$, there should exist globally optimal solution(s) that minimise the average waiting time of the commuters. In practice, such globally optimal solutions are difficult to find even with simplified systems. The approximate solution we present here is based on the assumption that empty taxis do not interact with each other. In particular, each empty taxi is not aware of the positions of all other empty taxis. It is also not aware of the number of commuters already waiting at each node.

For an empty taxi located at the i^{th} node, its action space for this particular time step consists of either staying put, or moving towards one of the accessible neighbouring node j ($\mathcal{A}_{ij} \neq 0$ and note $\mathcal{A}_{ii} = 1$). Ignoring the potentially existing queue of commuters at each node, the probability of picking up a commuter at the next node is given by $1-(1-g_j)^{\mathcal{A}_{ij}^{-1}}$. We also need to normalise this probability by the number of time steps it takes to reach j. This is because even though g_j may be large, it is still not worthwhile to go to j if it is too far away. We can thus assign an immediate value of the neighbouring node j as follows:

$$\mathcal{M}_{ij} = \frac{1-(1-g_j)^{\mathcal{A}_{ij}^{-1}}}{\mathcal{A}_{ij}^{-1}} \tag{1}$$

As a zeroth order approximation, we can assign $\mathcal{P}_{ij} \sim \mathcal{M}_{ij}$, so the policy matrix is determined by comparing the values of the neighbouring nodes. This is a crude approximation, because there is also a finite probability of not picking up a commuter at j, given by $(1-g_j)^{\mathcal{A}_{ij}^{-1}}$. Thus even in the case that $\mathcal{M}_{ij}$ is large, it is still not preferable to go to j, if all neighbours of j have very low probabilities of generating commuters. The same reasoning can be extended to even further neighbours, so that even the decision on which immediate neighbour of i to pick depends on the property of the entire road network. To account for such complications systematically, we define the following two high-dimensional arrays:

$$\mathcal{M}_{i_1\cdots i_n} = \frac{1-(1-g_{i_n})^{A^{-1}_{i_1i_2}+A^{-1}_{i_2i_3}\cdots+A^{-1}_{i_{n-1}i_n}}}{A^{-1}_{i_1i_2}+A^{-1}_{i_2i_3}\cdots+A^{-1}_{i_{n-1}i_n}} \tag{2}$$

$$\mathcal{N}_{i_1\cdots i_n} = (1-g_{i_n})^{A^{-1}_{i_1i_2}+A^{-1}_{i_2i_3}\cdots+A^{-1}_{i_{n-1}i_n}} \tag{3}$$

where Eq.(2) gives the normalised probability of finding a commuter at node i_n, after going through i_1 to $i_2\cdots$ to i_{n-1}. On the other hand, Eq.(3) is the probability of not finding a commuter at node i_n, after going through the same path. We can now define the n^{th} order contribution as follows:

$$\mathcal{T}^n_{ij} = \sum_{k_1,k_2,\cdots,k_n} \mathcal{N}_{ij}\mathcal{N}_{ijk_1}\mathcal{N}_{ijk_1k_2}\cdots N_{ijk_1\cdots k_{n-1}}\mathcal{P}_{jk_1}\mathcal{P}_{k_1k_2}\cdots\mathcal{P}_{k_{n-1}k_n}\mathcal{M}_{ijk_1\cdots k_n} \tag{4}$$

which is the normalised probability of the empty taxi moving from node i to j, and then to k_1, followed by $k_2\cdots$ until picking up a commuter at k_n. The summation i_n is over all the neighbours of the node i_{n-1}. Each choice of the next node is based on the policy matrix $\mathcal{P}$. At each intermediate node, there are no waiting commuters, until reaching the final node k_n and picking up a commuter. One should also note that $\mathcal{T}^0_{ij} = \mathcal{M}_{ij}$.

We can thus give the full formulation of the policy matrix as follows:

$$\mathcal{P}_{ij} \simeq \sum_{n=0}^{\infty} \mathcal{T}^n_{ij} \tag{5}$$

where $\simeq$ means equal up to the normalization condition that $\sum_j \mathcal{P}_{ij} = 1$. Hence the value of each neighbouring node is renormalised recursively by the policy matrix and the values of its own neighbours. We thus name this particular model the recursive value (RV) model.

In general, Eq.(5) has to be solved numerically in an approximate way with proper iterative processes. Physically, the higher orders are needed if in practice, an empty taxi has to go through a large number of nodes before finally picking up a commuter, which is generally the case for large road networks and when the demand for taxi is low. One should also note that the array elements of Eq.(2) and Eq.(3) are both smaller than unity. While that does not guarantee that Eq.(4) is smaller for large n because of the summations, that is empirically the case for all the simulations we have carried out in the following sections. We can now define the n^{th} order approximation of $\mathcal{P}_{ij}$ as follows:

$$\mathcal{P}_{ij}^{(n)} \simeq \sum_{k=0}^{n} \tilde{\mathcal{T}}_{ij}^{k}, \qquad \tilde{\mathcal{T}}_{ij}^{k} = \sum_{k_1,k_2,\cdots,k_n} \mathcal{N}_{ij}\mathcal{N}_{ijk_1}\mathcal{N}_{ijk_1k_2}\cdots N_{ijk_1\cdots k_{n-1}}\mathcal{P}_{jk_1}^{(n-k)}\mathcal{P}_{k_1k_2}^{(n-k)}\cdots\mathcal{P}_{k_{n-1}k_n}^{(n-k)}\mathcal{M}_{ijk_1\cdots k_n} \quad (6)$$

One can thus easily compute recursively the policy matrix at a specific order, which serves as good approximations to Eq.(5). The evaluation of the performance of the policy matrix will be carried out in the following sections.

V. Reinforcement Learning

Instead of a top-down approach in crafting rule based algorithms as we did in the previous section, we can also search for optimal policy matrices from bottom up with reinforcement learning [20]. More concretely, we employ the policy gradient method with function approximation [21] as the reinforcement learning algorithm of choice. In essence, we formulate the problem of turn-by-turn navigation as a Markov decision process. At each time step and for each empty vehicle, we give a direction for the subsequent direction of motion based on a parameterized policy matrix $\mathcal{P}_{ij}(\theta)$, where θ are trainable weights that can be adjusted to maximise some long-term performance metric. In our experiments, we use a shallow neural network for this parameterization task. Note that the present setting is a multi-agent control environment, but we simplify matters by using a common parameterization for all vehicles and combine their reward signals. More concretely, at each optimisation iteration, a large number of instances consisting of empty taxis' positions, choices of actions, as well as the long term rewards of such actions, are generated from the simulation. Note that information from different vehicles are combined. These information allows one to compute, using the policy gradient theorem [21], the (stochastic) gradient of the performance metric with respect to θ, and hence can be used to iteratively improve our parameterized policy matrix $\mathcal{P}_{ij}(\theta)$. The process is carried out iteratively until the performance of the numerical simulations based on the predefined metric converges. The overall procedure is illustrated in the flow chart in Fig.(1). We refer to [20] for a more detailed exposition on the policy gradient method.

In our particular implementation, the performance metric to maximise the average reward of all taxi drivers over the simulation time. For each taxi driver, a unit reward is bestowed for every time step when there is a passenger in the taxi. There is no punishment for empty taxis moving around, so in general maximising the average reward is equivalent to minimising the commuter waiting time, especially if there is no correlation between origins and the length of the trip. A typical procedure of the reinforcement learning is shown in Fig.(2).

VI. Numerical Simulations

In this section we carry out extensive microscopic numerical simulations to evaluate and compare the performance of different policy matrices. While we can do this both on artificial road network or the empirical road network

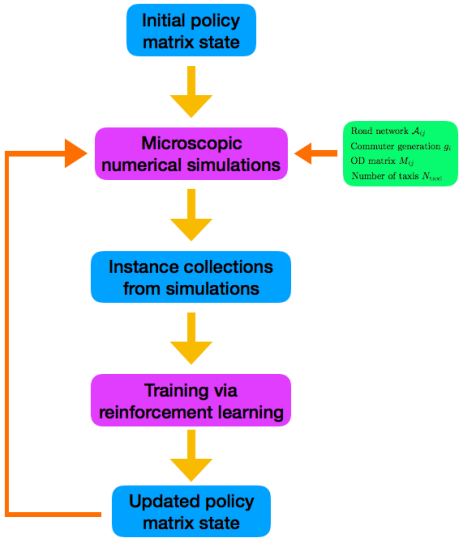

Fig. 1. The iterative process with reinforcement learning.

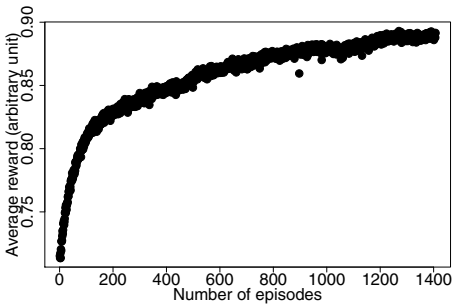

Fig. 2. A typical run of reinforcement learning, starting with a random policy. The average reward increases with the number of episode. Each episode consists of a simulation run of the taxi dynamics for 50000 seconds.

from different cities, the concepts and the behaviours of the simulations are qualitatively the same. For simplicity and without loss of generality, we focus on artificial road network in this paper. The benchmark for evaluating the matrix policies is the average total travel time of the commuters, which is the sum of the averaged waiting time and the averaged trip time (the time spent in the taxi). The artificial road network consists of a square lattice with random bi-directional edge weights and a total of 10000 nodes. Both g_i and M_{ij} are non-uniform and generated randomly. We would like to emphasise here that many different realisations of the commuter generation and OD patterns are tested, the results are qualitatively very similar.

A comparison of the real-time booking and road-side hailing algorithms are shown in Fig.(3). For the latter, a random policy matrix is used, so that empty taxis just undergo a random walk. In both cases, given the same demand patterns, the

average travel time decreases with an increase of the number of taxis operating in the road network. Interestingly, there is a sharp phase transition with real-time booking, separating the undersaturated region (when there is not enough taxis to meet the commuter demand) and the oversaturated region (when the supply of taxis exceeds demand). In contrast, with road-side hailing and random walking of empty taxis, the averaged travel time follows a smooth exponential-like decrease. The theoretical analysis of such differences will be discussed elsewhere. Intuitively, real-time booking should be more effective, since the taxi drivers have information about where the waiting commuter is (while for ride-side hailing such information is not available to the drivers). However, real-time booking is not as efficient as road-side hailing when the demand for taxis exceeds the supply, as one can see from the averaged travel time in Fig.(3). This is because booked taxis are committed and unable to cater to commuters along their way to pick up the assigned commuter, unlike the case with road-side hailing. With high demand, the impact of missing commuters for booked taxis will exceed the benefit of knowing the locations of the assigned commuters, leading to less efficiency overall. Such interesting dynamical features are important for optimising taxi systems under different situations.

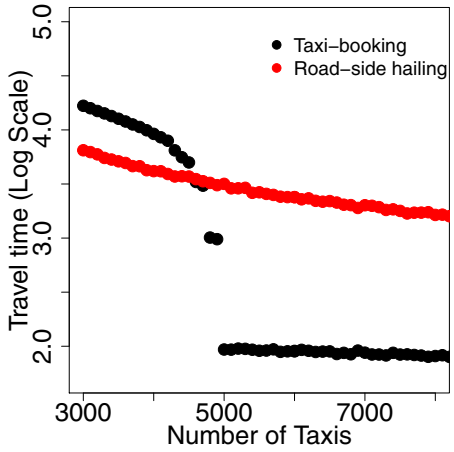

Fig. 3. Comparison of the taxi dynamics between real-time booking and road-side hailing. The former has a clear kink and phase transition from the oversaturated region to the undersaturated region. For the latter, the dependence of average traveil time on the taxi number is a smooth curve that persists to very large number of taxis (not plotted here). In the simulation, the road-side hailing is implemented with random roaming of empty taxis.

We will now focus on the case with road-side hailing, with the empty taxis seeking commuters with different algorithms. If the empty taxis just randomly roam around the road network, we denote it as the random walk (RW) model; if the algorithm from Sec. IV is used, we denote it as the recursive value (RV) model with different orders as defined by Eq.(6); if the policy matrix from reinforcement learning is used, we denote it as the reinforcement learning (RL) model. The performance of different algorithms is illustrated in Fig.(4) when the taxi system is oversaturated (insufficient supply of taxis).

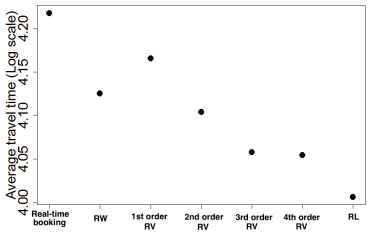

Fig. 4. Comparisons of different models via microscopic simulation, with a square lattice of 10000 nodes and random edge weights, and a total of 100 taxis. One commuter is generated per second stochastically with a random non-uniform distribution over the entire road network

It is quite clear that the RV mode can significantly outperform the RW model (note the Log scale of the y-axis in Fig.(4)). The RL model in general is the most efficient, given enough training with more than 2000 episodes. The performance of the converged RL model also is sensitive to the initial policy matrix. This is because given the complexity of this problem, it is almost impossible to find the global optimal solution for the policy matrix. A better initial guess of the policy matrix not only can improve the rate of converge, it in many cases can also lead to a more optimal solution. In Fig.(5), we show that the RV model is very useful in constructing an initial guess leading to a better converged algorithm via RL model, as compared to using RW model as an initial guess. In many cases, especially with a large number of taxis, RV model is the only viable way to construct efficient algorithm with reinforcement learning; using RW model as an initial state does not lead to convergence with reinforcement learning.

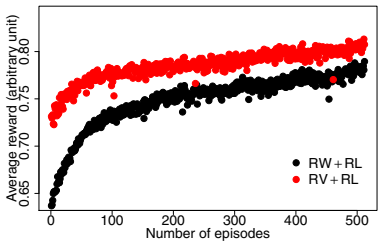

Fig. 5. Comparison of the convergence and performance with reinforcement learning, using the random policy as the initial state, against using the RV policy as the initial state.

VII. CONCLUSIONS AND OUTLOOKS

In this paper, we presented methodologies in finding efficient algorithms for routing of empty taxis so as to maximise the probability of finding commuters, and thus reducing the average waiting time of commuters in a specific taxi system. Using a comprehensive agent based microscopic simulation platform as the tool, we evaluated the performances of the random walk (RW) model, a recursive value (RV) model constructed by us, as well as a reinforcement learning (RL) model based on reinforcement algorithm that can greatly improve the efficiency of the overall taxi system.

The methodologies we develop here can lead to useful applications guiding taxi drivers, as well as intelligent softwares for driverless taxis.

The RV model we have constructed can be easily computed numerically given the pattern of demand for taxis in a road network, and in most cases it can vastly outperform the RW model. While in general the RL model is more effective than the RV model, the latter not only has the advantage from the ease of computation, but also from serving as the initial intelligent guess for reinforcement learning. The combination of the RV and RL algorithm can thus lead to even better algorithms and more efficient taxi systems. All the computations in this paper is done with a single CPU core, and much better performances can be expected with large scale parallelisation and GPU implementation. This is our current work in progress, together with the development of better numerical methods for iterative computations of the RV model.

The methodologies presented in this paper are universally applicable to a wide range of empirical and synthetic systems. Detailed implementations have been carried out with the Singapore road network (see Fig.(6), a total of 27185 nodes are present in the road network) with the empirical origin and destination distributions of the taxi trips, which we did not include in the main text due to the length limit. The reinforcement learning algorithms and the microscopic simulation platform can also be generalised to the cases with taxi ride-sharing, for the optimal routing and route-matching between shared commuters. Substantial improvement of the efficiency of the taxi system would involve optimisation of routing in a hybrid form consisting of real-time booking, turn-by-turn manoeuvring for empty taxis, as well as taxi ride-sharing. We are also working on incorporating traffic conditions (i.e. congestions) in the optimisation, which can be readily implemented by modifying the road network adjacency matrix or making it time-dependent according to the empirical data.

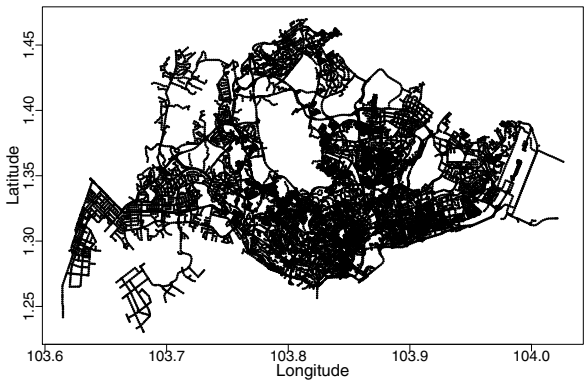

Fig. 6. The road network of the city of Singapore, consisting of a total of 27185 nodes.

ACKNOWLEDGMENT

We thank Dr. Ren Shen and Dr. Li Zengxiang for helpful discussions.

REFERENCES

[1] Moorthy, A., De Kleine, R., Keoleian, G., Good, J. et al., "Shared Autonomous Vehicles as a Sustainable Solution to the Last Mile Problem: A Case Study of Ann Arbor-Detroit Area", *SAE Int. J. Passeng. Cars Electron. Electr. Syst.* **10**(2) (2017)

[2] Pedro M. dOrey, Ricardo Fernandes and Michel Ferreira, "Reducing the Environmental Impact of Taxi Operation: the Taxi-sharing Use Case", *Proc. of 12th Int. Conf. on ITS Telecommunications* (2012)

[3] H. Yang, K.I. Wong and S.C. Wong, "Modeling Urban Taxi Services in Road Networks: Progress, Problem and Prospect", J. Adv. Transport, **35**, 237 (2001).

[4] K.I. Wong, S.C. Wong, and H. Yang, "Modeling urban taxi services in congested road networks with elastic demand", *Transportation Research Part B: Methodological* **35**(9) 819–842 (2001)

[5] P.Y. Chen, J.W. Liu, and W.T. Chen, "A Fuel-Saving and Pollution-Reducing Dynamic Taxi-Sharing Protocol in VANETs", *IEEE 72nd Veh. Technol. Conf., Sep.* (2010)

[6] B. Yang, S. Ren, E.F. Legara, Z.X. Li, E.Y.X. Ong, L. Lin and C. Monterola, "Phase Transition in Taxi Dynamics and Impact of Ridesharing", arXiv: 1801.00462

[7] J.M. Salanova, M. Estrada, G. Aifadopoulou and E. Mitsakis, "A review of the modeling of taxi services ", Procedia Social and Behavioral Sciences **20** (2011) 150?161.

[8] X. Hu, S. Gao, Y.C. Chiu and D.Y. Lin, "Modeling Routing Behavior for Vacant Taxicabs in Urban Traffic Networks", Transportation Research Record, **2284**, DOI: 10.3141/2284-10.

[9] A. Poulhes and J. Berrada, "User assignment in a smart vehicles? network: dynamic modelling as an agent-based model", Transportation Research Procedia, **27**, 865 (2017).

[10] M. Ramezani and M. Nourinejad, "Dynamic modeling and control of taxi services in large-scale urban networks: A macroscopic approach", Transportation Research Part C, in press (2017).

[11] H. Yang and T. Yang, "Equilibrium properties of taxi markets with search frictions", Transport. Research Part. B. **45**, 696 (2011).

[12] Y. Shi and Z. Lian, "Optimization and strategic behavior in a passenger?taxi service system", European Journal of Operational Research, **249**, 1024 (2016).

[13] R.C.P. Wong, W.Y. Szeto and S.C. Wong, "Bi-level decisions of vacant taxi drivers traveling towards taxi stands in customer-search: Modeling methodology and policy implications", Transport Policy, **33**, 73 (2014).

[14] L. Moreira-Matias, J. Gama, M. Ferreira and L. Damas, "A predictive model for the passenger demand on a taxi network", International IEEE Conference on Intelligent Transportation Systems (ITSC), DOI: 10.1109/ITSC.2012.6338680.

[15] R.C.P. Wong, W.Y. Szeto, S.C. Wong and H. Yang, "Modelling multi-period customer-searching behaviour of taxi drivers", Transportmetrica B, **2**, 40 (2014).

[16] R. Bai, J. Li, J.A.D. Atkin and G. Kendall, "A novel approach to independent taxi scheduling problem based on stable matching", Journal of the Operational Research Society, **65**, 1501 (2014).

[17] M. Nourinejad and M. Ramezani, "Developing a large-scale taxi dispatching system for urban networks", IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), 2016, DOI: 10.1109/ITSC.2016.7795592

[18] J. Ke, H. Zheng, H. Yang and X. Chen, "Short-term forecasting of passenger demand under on-demand ride services: A spatio-temporal deep learning approach", Transportation Research Part C, **85**, 591 (2017).

[19] J. Long, W.Y. Szeto, J. Du and R.C.P. Wong, "A dynamic taxi traffic assignment model: A two-level continuum transportation system approach", Transportation Research Part B, **100**, 222 (2017).

[20] R.S. Sutton and A.G. Barto, "Reinforcement learning: An introduction", *MIT press Cambridge.* (1998)

[21] R.S. Sutton, D.A. McAllester, S.P. Singh and Y. Mansour, "Policy gradient methods for reinforcement learning with function approximation", *Advances in neural information processing systems.* (2000).

[22] C. Brownell and A. Kornhauser, "A Driverless Alternative: Fleet Size and Cost Requirements for a Statewide Autonomous Taxi Network in New Jersey", Transportation Research Record, **2416**, DOI: 10.3141/2416-09.