This document is downloaded from DR-NTU (https://dr.ntu.edu.sg)
Nanyang Technological University, Singapore.

A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time

Cao, Zhiguang; Guo, Hongliang; Zhang, Jie

2018

Cao, Z., Guo, H., & Zhang, J. (2018). A multiagent-based approach for vehicle routing by considering both arriving on time and total travel time. ACM Transactions on Intelligent Systems and Technology, 9(3), 25-. doi:10.1145/3078847

https://hdl.handle.net/10356/84440

https://doi.org/10.1145/3078847

© 2017 ACM. All rights reserved. This paper was published by ACM in ACM Transactions on Intelligent Systems and Technology and is made available with permission of ACM.

Downloaded on 27 Aug 2022 03:40:22 SGT

A Multiagent-based Approach for Vehicle Routing by Considering both Arriving on Time and Total Travel Time

ZHIGUANG CAO[1], HONGLIANG GUO[2], JIE ZHANG[3],
[1]Energy Research Institute @NTU, Nanyang Technological University, Singapore
[2]School of Automation Engineering, University of Electronic Science and Technology of China, China
[3]School of Computer Science and Engineering, Nanyang Technological University, Singapore
ULRICH FASTENRATH[4],
[4]Department of Traffic Information Management and Routing Optimization, BMW Group, Germany

Arriving on time and total travel time are two important properties in multiagent-based route guidance. Existing route guidance approaches always consider them independently because they may conflict with each other. In this paper, we develop a decentralized multiagent-based vehicle routing approach to integrate the two properties by expressing them as two objective terms of a route assignment problem. Regarding arriving on time, it is formulated based on the probability tail model, which aims to maximize the probability of reaching destination before deadline. Regarding total travel time, it is formulated as a quadratic term, which aims to minimize the expected travel time from current location to the destination based on the potential assignment. To better reduce the additional travel time caused by loose deadlines, we design a weight for the term of total travel time, the value of which is comparatively large if the deadline is loose. This multiagent-based approach is characterized by two types of agents, in which, vehicle agents follow the local route guidance by infrastructure agents at each intersection, and infrastructure agents perform the route guidance by solving the corresponding route assignment problem. Additionally, we improve the proposed approach of route guidance in several aspects, including travel time prediction, computational efficiency and infrastructure agent communication. Experimental results on real road networks justify its ability to increase the average probability of arriving on time, reduce total travel time and enhance the overall routing performance.

Additional Key Words and Phrases: Intelligent Transportation Systems; Multiagent-based Route Guidance; Arriving on Time; Probability Tail Model; Total Travel Time.

1. INTRODUCTION

Vehicle route guidance has been a challenging problem in transportation and mobility, which is crucial for the sustainable development of any city [Zheng et al. 2014; Zhang et al. 2015; Cao et al. 2016a]. It attracts broad and deep attention from the government, industry and research communities due to its high relevance to people's daily life [Chester 2015; Guo et al. 2014; Zheng and Xie 2011; Yuan et al. 2010; Rubinstein et al. 2012]. Multiagent-based approaches are often applied in route guidance [Gan et al. 2015; France et al. 2003; Wilt and Botea 2014; Chen et al. 2014; Zolfpour-Arokhlo et al. 2014; Lujak et al. 2015], because the agent metaphor for modeling a participant or decision-maker can capture complex constraints connecting all problem-solving phases [Smith et al. 2006; Oh and Smith 2008], especially in cooperative vehicle route guidance [Bazzan and Klügl 2014]. A transportation system can be modeled as a large, distributed and dynamic multiagent system where vehicles represented as agents move on the road network following their own routes, which are determined by themselves or roadside infrastructure agents [Jiang et al. 2014; Cao et al. 2016c].

In multiagent-based route guidance, LET (least expected travel time) paths are first proposed to provide route guidance to the vehicle agents by the policy-making agents

This work is supported by the BMW Group, Munich, Germany.

(e.g., city manager or transportation authority) [Yamashita et al. 2005; Li et al. 2009; Claes et al. 2011; Pan et al. 2013; Liang and Wakahara 2014; Jiang et al. 2014; Wang et al. 2014a],. They consider LET paths as the optimal option mainly for two reasons: (1) Drivers generally prefer shorter travel time; (2) LET paths are likely to result in less total travel time for all vehicles in the whole transportation system, which is environment-friendly in view of that fuel consumption and air pollution are directly relevant with the total travel time [Liang and Wakahara 2014; Jiang et al. 2014]. Although the LET path in multiagent-based route guidance is popular and has achieved big success considering the environmental impact, a critical issue still remains to be addressed: in real traffic, different drivers may have different deadlines, and even the same driver may have different deadlines in various scenarios. For instance, if they want to catch up important appointments, their deadlines might be tight; if they go shopping, deadlines might be loose. Simply seeking LET paths for all drivers may cause some drivers with tight deadlines to miss their deadlines due to the influence from other drivers, especially those with loose deadlines. This will increase the drivers' frustration and impatience, and, as a consequence, the accident occurrence rate.

The *probability tail model* is proposed as a criterion of the optimal route, which aims to maximize the probability of reaching destination before deadline (i.e., arriving on time) [Fan et al. 2005; Nikolova et al. 2006; Nie and Wu 2009; Lim et al. 2009; Lim et al. 2013]. This criterion has two attractive properties: 1) it takes the specific demand of deadline into account, giving drivers an extra dimension of settings; 2) it uses a probabilistic instead of deterministic metric to evaluate stochastic traffic situations (i.e., the variance in the probability always relates to the *risk*), which is more realistic especially when facing real-world uncertainties. The probability tail model is much consistent with real-world travel behavior, and has been applied in many crucial scenarios, e.g., flight catching, fire rescue and organ delivery. However, the probability tail model is originally designed for a single vehicle, which independently pre-computes a path before each vehicle's departure. Traffic is known to be dynamic, so the optimality of a pre-computed path may not hold when all vehicles are en-route. Therefore, a multiagent-based cooperative route guidance approach had been developed in [Cao et al. 2016a], which extends the probability tail model to the multiple vehicle settings by solving a route assignment problem at each intersection. This approach is promising in that it is decentralized, and can increase the chances of arriving on time for all relevant vehicles by providing local route guidance. However, a critical issue is unsolved in this approach: it only considers arriving on time, and some vehicle agents may be assigned to detoured routes as long as they can guarantee reaching destination before deadline. Those routes are likely to cause additional total travel time, especially for the vehicle agents with loose deadlines, which are deviated from the normal driver's preference of shorter travel time. Moreover, unnecessary total travel time always causes additional fuel consumption and air pollution, which is not environment-friendly.

Therefore, it is desirable to leverage the advantages of considering both arriving on time and total travel time at the same time in the muitiagent-based route guidance. Consequently, in this paper, we extend the work in [Cao et al. 2016a], by integrating the properties of arriving on time and total travel time into the same route assignment problem. More specifically, we incorporate a quadratic term, which represents the expected travel time from current location to the destination for the potential road link assignment, into the objective function of the original arriving on time problem. We also design a weight for the expected travel time term, the value of which is comparatively large if the deadline is loose. Besides, three improvements are proposed for our new approach: the predicted travel time on each assigned road link is refined by iteratively linearizing a non-linear function; the computational efficiency is enhanced by reformulating the route assignment problem as a mixed integer linear programming (MILP) problem; and the overall performance is improved by allowing communication between neighboring infrastructure agents. Experimental results on real road networks show that our approach outperforms traditional methods, which can increase the chances of arriving on time and reduce the total travel time.

The rest of this paper is organized as follows. In Section 2, we lay down related work on route guidance approaches considering total travel time and arriving on time, respectively. In Section 3, we illustrate the framework of our multiagent-based vehicle route guidance. In Section 4, firstly, we elaborate the proposed approach that considers arriving on time for all vehicle agents; then we describe how to integrate the total travel time, both of which are expressed as route assignment problems, respectively; and finally we provide the bound analysis and algorithm summary. In Section 5, we propose three improvements to our route guidance approach. In Section 6, we present experimental settings and performance results regarding all the approaches and improvements. The paper ends with conclusions and possible future works in Section 7.

2. RELATED WORK

Most of the multiagent-based vehicle routing approaches rely on the LET paths for route guidance, to reduce the total travel time of vehicle agents in the whole transportation system. In one of the early multiagent-based approaches for route guidance [Yamashita et al. 2005], a global server agent constantly collects intentions of routes from all vehicle agents. It then computes a predicted LET path for each individual vehicle agent by cooperatively exploring the collected intentions, and all vehicle agents update their routes at each intersection accordingly. Based on the work in [Yamashita et al. 2005], Li et al. [2009] predict the potential traffic on each road link at different time slots according to the collected intentions, then a corresponding LET path is constantly computed for each individual vehicle. Similarly, a modified A* algorithm [Pan et al. 2013] incorporates a repulsion scheme into the expression of weights on all road links. Then each vehicle agent recursively computes a LET path in a centralized manner, to avoid the situation where too many vehicle agents rush into a same route. Liang and Wakahara [2014] propose a personalized rerouting strategy by first ranking vehicles and then calculating a predicted LET path for a vehicle according to the decision of those who ranked higher. In another centralized approach [Amarante and Bazzan 2012], each vehicle agent is assumed to know real-time traffic condition on all road links, and dynamically travels along the latest LET path. Centralized approaches often suffer from low computational efficiency. Jiang et al. [2014] propose a decentralized pheromone-based vehicle rerouting approach, in which whenever congestion is predicted by a local infrastructure agent, the concerned vehicle agents will update their routes by choosing one of the best k LET paths. In another decentralized approach, Weerdt et al. [2013] aim to minimize the delays at charging stations for electric vehicles, which incorporates the intentions into the probabilistic arrival time of those vehicles to each road link. To find the best location to charge, they maximize the expected utility function for each vehicle. Nevertheless, the results are only verified by the measurement of total expected travel time rather than whether vehicles arrive on time. Decentralized multiagent approaches have the capability of adaptively updating routes according to dynamic traffic, and they are environment-friendly because they are likely to reduce the total travel time. However, they do not consider specific demands of vehicle agents, i.e., preferred deadlines.

Several optimization approaches has been proposed to take drivers' preferred deadlines into account. Particularly, the probability tail model is widely adopted for vehicle route guidance [Fan et al. 2005; Lim et al. 2009; Lim et al. 2013; Cao et al. 2016b], which aims to maximize the probability of arriving at destination before deadline and is formally expressed as [Lim et al. 2013; Cao et al. 2016b]:

$$\max_{\vec{\mathbf{x}}} Prob(\vec{\mathbf{w}}^\top \vec{\mathbf{x}} \leq T) \ \big| \mathbf{M}\vec{\mathbf{x}} = \vec{\mathbf{b}};\ \vec{\mathbf{x}} \in \{0,1\}^{|A_r|}, \tag{1}$$

where $\vec{\mathrm{w}}$ denotes travel time for each road link; M is the node-arc incidence matrix of the road network; T is the preferred deadline[1]; $\vec{\mathrm{b}}$ is an O-D vector, all elements of which are zeros except those for origin ("1") and destination ("-1"); $\vec{\mathrm{x}}$ refers to the set of road links where an element is "1" if the referred road link is on the concerned path. The equality constraint in Eq. (1) guarantees that $\vec{\mathrm{x}}$ is a connected path from origin to destination. This model is more consistent with real-world travel queries. To improve the computational efficiency of solving the probability tail model for stochastic vehicle routing, several quasi-convex optimization based solutions have been developed [Nikolova et al. 2006; Nie and Wu 2009; Lim et al. 2009]. But these solutions are limited by strong assumptions, such as Gaussian distribution of travel time, independence among travel time on different road links, and sufficiently large deadlines, which prevent its wide application in real transportation systems. To overcome these limitations, a data-driven solution is proposed [Cao et al. 2016b], which formulates the original stochastic routing problem as a cardinality minimization problem by directly exploring the real trajectory on road links [Wang et al. 2014b]. Although success is achieved from the computational perspective, most of existing approaches incorporating the probability tail model independently pre-calculate a path for each individual vehicle before it departs, without considering the intentions of others [Cao et al. 2016b]. Since traffic is always dynamic, optimality of a pre-computed path may not hold any more once all vehicles are en-route due to the influences from each other. To overcome this limitation, a decentralized multiagent-based route guidance approach is proposed, which aims to cooperatively increase the chances of arriving on time for all vehicle agents. Particularly, this approach is characterized by two types of agents, i.e., vehicle agents and infrastructure agents. Vehicle agents follow the route guidance by infrastructure agents at each intersection, and infrastructure agents locally collect intentions of concerned vehicle agents and formulate the route guidance as a route assignment problem, to guarantee their arrival on time. However, this approach does not consider total travel time, and some vehicle agents (i.e., especially those with loose deadlines) may be allocated to detoured routes to give way although they can still guarantee arrival on time. In this case, those detoured routes are likely to cause unnecessary additional travel time, which are not preferred by the normal drivers, or not environment-friendly either.

It is thus desirable to leverage both the advantages of the path based on probability tail model (that considers arriving on time regarding different levels of deadlines) and the LET path (that aims to reduce the total travel time) in the multiagent-based route guidance approach, which is what we propose in this paper.

3. MULTIAGENT-BASED ROUTE GUIDANCE

In this section, we briefly introduce the scheme of our decentralized multiagent-based approach for vehicle route guidance, which involves two types of agents, i.e., vehicle agents and infrastructure agents. Vehicle agents representing drivers, travel on a road network by following route guidance from infrastructure agents. Infrastructure agents, located at road intersections, collect intentions (i.e., deadlines and destinations) from vehicle agents, and conduct route guidance to the local vehicle agents.

In real traffic, vehicle agents may influence each other due to limited road capacity and rush hour effect. To consider such influence, intention collection is necessary [Ya-

[1] T is actually the remaining time to deadline, and we use "deadline" for simplification in the whole paper.

mashita et al. 2005; Li et al. 2009; Claes et al. 2011]. In our approach, each vehicle agent determines a destination and a preferred deadline before departure, and travels along an initial route given by the probability tail model in Eq. (1). Each infrastructure agent is associated with all traffic lights at a road intersection. It collects intentions from vehicle agents, which are (1) located on road links directly connected to that infrastructure agent; and (2) facing red lights. The motivation for the latter is to avoid unnecessary and frequent route change. Facing green lights may imply that the current route is sufficiently satisfactory.

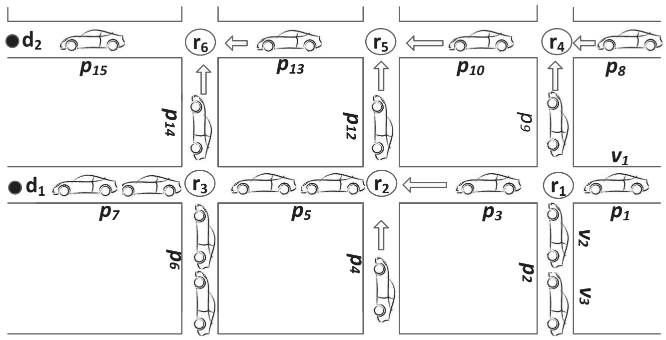


Fig. 1: Two Types of Agents and Intention Collection

Take a one-way road network in Fig. 1 as an example, where r_g is an infrastructure agent, v_i is a vehicle agent, and p_k is a road link. Assume that at this moment, the traffic light associated with r_1 shows red color to p_1 and p_2. Then r_1 collects intentions of v_1, v_2 and v_3. Since this red color will last for a while, r_1 will also collect intentions of other vehicle agents if they later enter p_1 or p_2 during the same red color period. Other infrastructure agents also work in the same manner.

Once the intentions are collected, the infrastructure agent at each road intersection will compute and provide route guidance, to increase the chance of arriving on time and reduce total travel time for all concerned vehicle agents. Then the vehicle agents update routes accordingly. On the other hand, vehicle agents generally have different types of deadlines. Simply seeking LET paths for all vehicle agents may cause some of them with tight deadlines to miss their deadlines due to the influences from other vehicle agents with loose deadlines. Motivated by this concern, a desirable approach is to distribute vehicle agents with loose deadlines to detour crowded paths, to give ways to those with tight deadlines if necessary. At the same time, vehicle agents always face choices at an intersection: go straight, turn left, turn right or turn back to enter the next road link. Thus, in our approach, the infrastructure agent at each road intersection provides route guidance to vehicle agents by formulating it as a route assignment problem [Papageorgiou 1990], which relies on the collected intentions. In the next section, we will detail the computation process of the route assignment problem.

4. ROUTE ASSIGNMENT FOR THE ROUTE GUIDANCE

In this section, we focus on the route assignment problem formulation. We start with the simple case of only considering arriving on time, then proceed with integrating both arriving on time and total travel time. Bound analysis and implementation summary are provided in the later part of this section.

4.1. Route Assignment Formulation

4.1.1. Route Assignment Considering Arriving on Time. Take infrastructure agent r_1 and vehicle agents v_1, v_2 and v_3 in Fig. 1 as an example, and focus on the route assignment for v_1. Assume that: (1) destination of v_1 is d_2, and its preferred deadline is T_1; (2) v_1, v_2 and v_3 are currently facing red light, and will next enter p_3 or p_9. Assignment for an vehicle agent always influences others. Suppose that the predicted travel time of v_1 on p_3 and p_9 are T^p_{13} and T^p_{19}, which linearly depend on the number of assigned vehicle agents to the road links and their lengths. Besides, we need to consider traffic conditions from r_2 and r_4 to d_1 or d_2, where historical expected travel time is used because r_2 and r_4 are comparatively far away from r_1. Assume that the expected travel time from r_2 and r_4 to d_2 are T^e_{22} and T^e_{42}. Hence, if v_1 is assigned to p_3 or p_9, there would be relative deadlines on p_3 and p_9 for v_1, denoted as T^r_{13} and T^r_{19}. Since deadlines should always be non-negative, we have $T^r_{13} = max\{0, T_1 - T^e_{22}\}$, and $T^r_{19} = max\{0, T_1 - T^e_{42}\}$. Thus there would be a potential delay ξ_1 for v_1, satisfying $T^p_{13} - T^r_{13} \leq \xi_1$ and $T^p_{19} - T^r_{19} \leq \xi_1$ (i.e., ξ_1 is non-negative, and there is no delay only if $\xi_i = 0$). To guarantee arriving on time for the three vehicle agents, r_1 should minimize the cardinality[2] of $\vec{\xi}$ whose components are ξ_i ($i = 1, 2, 3$).

In view of the above example, we can generalize route assignment performed by each infrastructure agent. First introduce several symbols: (1) l_j, choices of road links to enter next, such as p_3 and p_9 in Fig. 1; (2) $x_{ij} \in \{0, 1\}$, $x_{ij} = 1$ means that v_i is assigned to l_j, otherwise $x_{ij} = 0$; (3) $I = \{1, ..., Q\}$ and $J = \{1, ..., L\}$, indices of vehicle agents and road link choices associated with the infrastructure agent. Then the problem of maximizing the chance of arriving on time for all concerned vehicle agents can be expressed as minimizing the cardinality of the potential delay $\vec{\xi}$, as follows:

$$\min_{\vec{\mathbf{x}}} Card(\vec{\xi}) \left| \begin{array}{l} \sum_{j \in J} (f_j(\vec{\mathbf{x}}) - T^r_{ij}) \cdot x_{ij} \leq \xi_i, \forall i \in I; \\ \sum_{j \in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0, 1\}, \end{array} \right. \quad (2)$$

where $\vec{\xi} = \{\xi_1, ...\xi_Q\}$, and delay occurs for v_i if $\xi_i > 0$; $\vec{\mathbf{x}}_j = (x_{1j}, ..., x_{Qj})$, indicates assignments to l_j; $f_j(\vec{\mathbf{x}})$, a linear function, denotes predicted travel time on l_j; T^r_{ij} is relative deadline for v_i on l_j; $\sum_{j \in J} x_{ij} = 1$ ensures that v_i can only enter one road link, thus only one potential delay takes effect in $\sum_{j \in J} (f_j(\vec{\mathbf{x}}) - T^r_{ij}) \cdot x_{ij}$. Particularly, the linear function $f_j(\vec{\mathbf{x}})$ for l_j is expressed as:

$$f_j(\vec{\mathbf{x}}) = c_j \sum_i x_{ij} + \gamma_j \quad (3)$$

where $\sum_i x_{ij}$ is amount of vehicle agents assigned to l_j, c_j and γ_j are coefficients. The cardinality minimization in Eq. (2) is difficult to be directly solved. We thus use ℓ_1-norm to approximately solve it [Kim et al. 2009], which can be further expressed as:

$$\min_{\vec{\mathbf{x}}} \sum_{i=1}^{Q} \xi_i \left| \begin{array}{l} \sum_{j \in J} (f_j(\vec{\mathbf{x}}) - T^r_{ij}) \cdot x_{ij} \leq \xi_i, \forall i \in I; \\ \sum_{j \in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0, 1\}. \end{array} \right. \quad (4)$$

Eq. (4) is a mixed integer quadratic programming (MIQP) problem in nature, which can be solved by existing solvers.

[2]Cardinality is the number of non-zero components in a vector.

Note that: (1) Eq. (4) only outputs a road link for a vehicle agent to enter next, but the remaining path from assigned road link to destination is also available due to the computation of relative deadline T^r_{ij}. The vehicle agent can then follow this complete route if it does not receive any further guidance after an route assignment, which may happen if afterwards it always faces green light; (2) As time elapses, deadline T_j will decrease, and we always use the latest T_j when route guidance is performed; (3) We previously take a simple one-way road network as an example, and double-way road links can also be easily applied, as long as infrastructure agents dynamically recognize on which road links vehicle agents are facing red light, and which road links are available to be assigned to those vehicle agents.

4.1.2. Route Assignment Considering both Arriving on Time and Total Travel Time. Previously, we formulated the route guidance as an assignment problem addressed by each infrastructure agent. According to Eq. (4), to increase the chance of arriving on time for all vehicle agents in some cases, the infrastructure agent has to assign vehicle agents with loose deadlines to detour crowed routes. This kind of assignment is helpful sometimes since it gives ways to those vehicle agents with tight deadlines. However, it may increase additional total travel time, especially for vehicle agents with loose deadlines.

Before formulating route assignment considering both arriving on time and total travel time, we would like to introduce *deadline coefficient* α for each vehicle agent, which is a positive parameter, denoting the levels of deadlines. Specifically, once the O-D pair of a vehicle agent is determined, an expected travel time T_e between origin and destination can be derived based on historical traffic data. Thus, the deadline can be expressed as:

$$T = \alpha \cdot T_e, \tag{5}$$

where T implies a tight deadline if $\alpha < 1$, and a loose deadline if $\alpha > 1$. Generally, assignment to detour crowed routes may happen to vehicle agents with loose deadlines, especially when vehicle agents with tight deadlines compete with them. Therefore, to reduce additional total travel time while keeping satisfactory probability of arriving on time, we incorporate travel time related measurement into the objective function of route assignment, which is accordingly formulated as:

$$\min_{\vec{\mathbf{x}}} \sum_{i=1}^{Q} (\xi_i + \tau_i \cdot \phi_i(\vec{\mathbf{x}})) \left| \begin{array}{l} \sum_{j \in J} (f_j(\vec{\mathbf{x}}) - T^r_{ij}) \cdot x_{ij} \leq \xi_i, \forall i \in I; \\ \sum_{j \in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0, 1\}, \end{array} \right. \tag{6}$$

where τ_i is the weight for total travel time measurement of v_i; and $\phi_i(\vec{\mathbf{x}})$ is the travel time measurement based on the assignment decision $\vec{\mathbf{x}}$. Particularly, $\phi_i(\vec{\mathbf{x}})$ consists of predicted travel time on assigned road link and historical expected travel time from assigned road link to destination. To be more specific, if v_i is assigned to l_j, $\phi_i(\vec{\mathbf{x}})$ would be expressed as:

$$\phi_i(\vec{\mathbf{x}}) = \sum_{j \in J} (f_j(\vec{\mathbf{x}}) + \psi_j(\vec{\mathbf{x}})) \cdot x_{ij}, \tag{7}$$

where $f_j(\vec{\mathbf{x}})$ is the predicted travel time function on l_j, expressed in Eq. (3), and $\psi_j(\vec{\mathbf{x}})$ refers to the historical expected travel time from l_j to destination of v_i, which can be obtained from a look-up table. To make the two parts in objective function of Eq. (6)

comparable, τ_i is calculated as follows:

$$\tau_i = \frac{\alpha_i(\epsilon_1 + \sum_{j \in J} \frac{max\{T_{ij}^e - T_i, 0\}}{|J|})}{\sum_{j \in J} \frac{T_{ij}^e}{|J|}}, \tag{8}$$

where α_i is the deadline coefficient for v_i; ϵ_1 is a small positive constant; T_{ij}^e is least expected travel time of the path from current location to destination, which has to traverse l_j; T_i is the latest deadline; and $|J|$ is the number of candidate road links to be assigned. The logic behind τ_i in Eq. (8) is to approximate the ratio of average possible delay over average possible travel time. Besides, in many cases, additional travel time is usually caused by the vehicle agent with loose deadline. Therefore, α_i in Eq. (8) will help to assign a high weight for total travel time in Eq. (6) if the deadline is loose. Note that, based on all above statements, Eq. (6) also comes down to an MIQP problem.

4.2. Bound Analysis

The ℓ_1-norm minimization in the route assignment of Eq. (4) is an approximation to the cardinality minimization in Eq. (2), and here we analyze its bounds. We assume that regarding Eq. (4), $\vec{\mathbf{x}}_1^*$ and $\vec{\xi}_1^*$ are the optimal solution to the primal problem, and p_1^* is the optimal value of the objective function. We also assume that the optimal value of objective function in Eq. (2) is p_2^*. Since Eq. (2) and Eq. (4) share the same constraints, $\vec{\mathbf{x}}_1^*$ and $\vec{\xi}_1^*$ are the feasible solution to the cardinality minimization problem in Eq. (2). Therefore p_1^* is an upper bound to p_2^*. The dual function of Eq. (2) can be expressed as $g(\vec{\lambda}, \vec{\nu}) = \min_{\vec{\mathbf{x}}, \vec{\xi}} \mathbf{Card}(\vec{\xi}) + \tau_i \cdot \phi_i(\vec{\mathbf{x}}) + \sum_{i \in I}(\nu_i(\sum_{j \in J} x_{ij} - 1)) + \sum_{i \in I}(\lambda_i(\sum_{j \in J}(f_j(\vec{\mathbf{x}}) - T_{ij}^r) \cdot x_{ij} - \xi_i))$. Since any feasible solution to the dual problem is a lower bound to the primal problem [Boyd and Vandenberghe 2004], we assume that $\vec{\nu} = \vec{\mathbf{0}}$, the lower bound can be easily calculated by solving $g(\vec{\lambda}, \vec{\mathbf{0}}) = \min_{\vec{\mathbf{x}}, \vec{\xi}} \mathbf{Card}(\vec{\xi}) + \tau_i \cdot \phi_i(\vec{\mathbf{x}}) + \sum_{i \in I}(\lambda_i(\sum_{j \in J}(f_j(\vec{\mathbf{x}}) - T_{ij}^r) \cdot x_{ij} - \xi_i))$ (the second and third terms can be expressed as linear functions, see Section 5.2), where λ_i is known and $\lambda_i \geq 0$. In addition, the accuracy for solving the general cardinality minimization problem can be improved by some variants of ℓ_1-norm minimization, e.g., iterated weighted ℓ_1-norm heuristic [Candes et al. 2008]. We do not elaborate the details here since they are beyond the interest of this paper.

4.3. Pseudo-Code Summary

We summarize the proposed multiagent-based route guidance considering both arriving on time and total travel time in Algorithm 1. Lines 1-2 initialize infrastructure agents and vehicle agents. In Lines 3-33, each infrastructure agent recursively assigns paths to vehicle agents who need route guidance at intersections, until they all reach destinations. Particularly, in Lines 4-12, each vehicle agent travels along a current route and updates its deadline if it has not reached destination. In Lines 14-23, during the red-color phase, each infrastructure agent recursively finds the set of vehicle agents who need route guidance and collects their intentions including deadlines and destinations. In Lines 25-29, upon the completion of the red-color phase, each infrastructure agent computes the optimal road links for all concerned vehicle agents to enter next based on the route assignment in Eq. (4), and accordingly updates theirs routes. We would like to note that Algorithm 1 will only consider arriving on time if we set all τ_i in Eq. (6) as 0.

5. FURTHER IMPROVEMENTS

In this section, we further improve the previously proposed route guidance approach, in the following aspects: (1) predicted travel time on assigned road links, through it-

ALGORITHM 1: Multiagent-based Route Guidance

Input : $V = \{v_1, ..., v_Q\}$, a set of vehicle agents;
$R = \{r_1, ..., r_G\}$, a set of infrastructure agents;
$V^g = \emptyset$, vehicle agents that need guidance at r_g;
TL_g, traffic lights associated with r_g;
T_{gn}, T_{rd}, green light and red light duration;
$flg = 0$, indicator of route guidance computation;

Each $r_g \in R$ turns on its associated traffic lights TL_g;
Each $v_i \in V$ determines destination d_i, deadline T_i, deadline coefficient α_i (determined by T_i), and computes an initial path P_i using Eq. (1);
while $|V| > 0$ **do**
  foreach $v_i \in V$ **do**
    if v_i reaches d_i **then**
      Deletes itself from V: $V = V - v_i$;
    end
    else
      Travels along P_i;
      Updates deadline T_i;
    end
  end
  foreach $r_g \in R$ **do**
    foreach $tl_k \in TL_g$ **do**
      Runs according to T_{gn} and T_{rd};
      if tl_k is in red color phase **then**
        r_g finds v_k facing tl_k: $V^g = V^g + v_k$;
        r_g collects d_k and T_k from v_k;
      end
      if tl_k is at end of red color phase **then**
        $flg = 1$;
      end
    end
    if $flg == 1$ **then**
      foreach $v_i \in V^g$ **do**
        Computes relative deadline for v_i based on the latest T_i;
        Computes its new path $P_i^{'}$ via Eq. (6);
        Updates its route: $P_i = P_i^{'}$;
      end
      Resets parameters: $V^g = \emptyset$, $flg = 0$;
    end
  end
end

erative linearization for more accurate prediction; (2) computational efficiency of our approach, by introducing new variables and additional linear constraints; (3) real-time traffic information acquirement, by allowing communications between infrastructure agents.

5.1. Refinement of Predicted Travel Time

In view of the analysis in Section 4.1.1, it is beneficial to assume that $f_j(\vec{x})$ in Eq. (3) is linear, because in this case route assignment in Eq. (6) can be formulated as a canonical optimization problem, i.e., MIQP, which enables tractable computation. However, by visualizing the statistical relationship between amount of vehicle agents and ex-

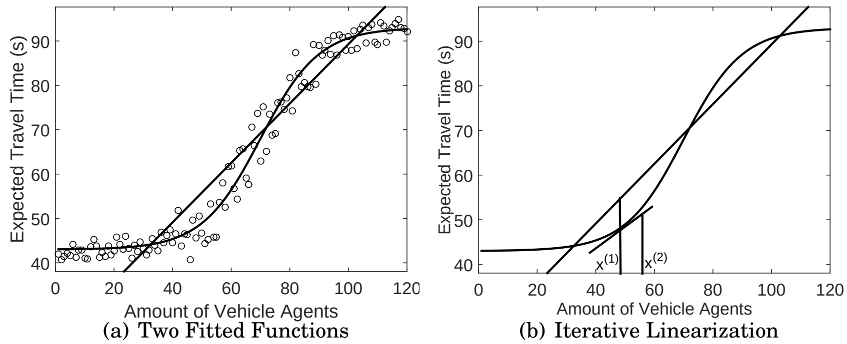


Fig. 2: Illustration of the Refinement of Travel Time Prediction Function.

pected travel time on some road links (i.e., Fig 2.(a)), it is obvious that a non-linear function fits the relationship better than a linear function. Therefore, to guarantee both tractable computation and accurate route assignment, we adopt the non-linear function to predict the travel time on assigned road link, but solve the optimization problem in Eq. (6) by iterative linearization. Before showing this refinement, we would like to reformulate the problem in Eq. (6) as Eq. (9):

$$\min_{\vec{\mathbf{x}},t} t \left| \begin{array}{l} \sum_{i=1}^{Q}(\xi_i + \tau_i \cdot \sum_{j \in J}(f_j(\vec{\mathbf{x}}) + \psi_j(\vec{\mathbf{x}})) \cdot x_{ij}) \leq t; \\ \sum_{j \in J}(f_j(\vec{\mathbf{x}}) - T_{ij}^r) \cdot x_{ij} \leq \xi_i, \forall i \in I; \\ \sum_{j \in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0,1\}, \end{array} \right. \tag{9}$$

To illustrate the refinement of the travel time prediction, we take the two functions in Fig 2 (b) as an example, and denote the linear and non-linear functions as $\mathcal{F}_j(\vec{\mathbf{x}})$ and $\mathbb{F}_j(\vec{\mathbf{x}})$ respectively. In the first iteration, we use $\mathcal{F}_j(\vec{\mathbf{x}})$ to replace $f_j(\vec{\mathbf{x}})$, and we get a solution, i.e., $\vec{\mathbf{x}}^{(1)}$ to Eq. (6). In the next iteration, we use the first order Taylor expansion of $\mathbb{F}_j(\vec{\mathbf{x}})$ at $\vec{\mathbf{x}}^{(1)}$ to replace $f_j(\vec{\mathbf{x}})$, which is expressed as:

$$f_j^{(k)}(\vec{\mathbf{x}}) = \mathbb{F}_j(\vec{\mathbf{x}}^{(k-1)}) + \nabla\mathbb{F}_j(\vec{\mathbf{x}}^{(k-1)})(\vec{\mathbf{x}} - \vec{\mathbf{x}}^{(k-1)}), \tag{10}$$

where $\nabla\mathbb{F}_j(\vec{\mathbf{x}}^{(k-1)})$ refers to the first order derivative of $\mathbb{F}_j(\vec{\mathbf{x}})$ at $\vec{\mathbf{x}}^{(k-1)}$, and k is the iteration number (k is equal to 2 in this context). Consequently, we get an optimal solution to Eq. (6), i.e., $\vec{\mathbf{x}}^{(2)}$. We continue to search for a better solution around $\vec{\mathbf{x}}^{(2)}$ on $\mathbb{F}_j(\vec{\mathbf{x}})$, until the objective function value in Eq. (6) does not significantly decrease. To guarantee that the algorithm locally searches for the optimum on the linearized straight line in each iteration, we limit the searching area by adding a constraint.

Therefore, the optimization problem in the k-th iteration is formulated as:

$$\min_{\vec{\mathbf{x}},t} t \left| \begin{array}{l} \sum_{i=1}^{Q}(\xi_i + \tau_i \cdot \sum_{j\in J}(f_j^k(\vec{\mathbf{x}}) + \psi_j(\vec{\mathbf{x}})) \cdot x_{ij}) \leq t; \\ \sum_{j\in J}(f_j^{(k)}(\vec{\mathbf{x}}) - T_{ij}^r) \cdot x_{ij} \leq \xi_i, \forall i \in I; \\ \sum_{j\in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0,1\}; \\ -\epsilon_2 \leq \sum_{i\in I} x_{ij} - \sum_{i\in I} x_{ij}^{(k-1)} \leq \epsilon_2, \forall j \in J, \end{array} \right. \tag{11}$$

where ϵ_2 is a positive integer, denoting local search range.

5.2. Improvement on Computational Efficiency

In this subsection, we improve the computational efficiency by reformulating the route assignment as a mixed integer linear programming (MILP) problem. To better illustrate the idea, we take the route assignment considering both arriving on time and total travel time in Eq. (9) as an example, which is completely equal to Eq. (6). Route assignment in Eq. (9) is an MIQP problem mainly due to $f_j(\vec{\mathbf{x}}) \cdot x_{ij}$ in the first two constraints. After unfolding, quadratic part comes from the term $x_{kj} \cdot x_{ij}$ ($k \in I$, and x_{kj} is a component of $\vec{\mathbf{x}}_j$). Since both $x_{kj}, x_{ij} \in \{0,1\}$, $x_{kj} \cdot x_{ij}$ can be replaced by x_{ij} if $k = i$. Therefore, the term $x_{kj} \cdot x_{ij}$ is quadratic only if $k \neq i$. However, $x_{kj} \cdot x_{ij}$ ($k \neq i$) can also be replaced by a binary variable with two additional linear constraints.

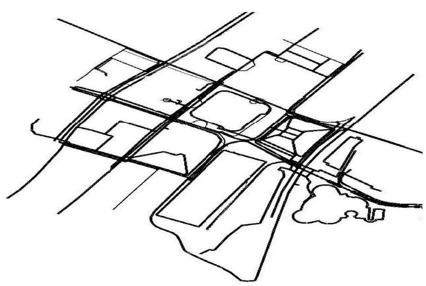
There are four correct permutations for vector (x_{kj}, x_{ij}, $x_{kj} \cdot x_{ij}$), i.e., (0,0,0), (0,1,0), (1,0,0) and (1,1,1). And we introduce a new variable $y_{kij} \in \{0,1\}$ to replace $x_{kj} \cdot x_{ij}$ ($k \neq i$), where eight permutations for vector (x_{kj}, x_{ij}, y_{kij}) exist. Therefore we add two linear cuts, i.e., $x_{kj} + x_{ij} + y_{kij} \leq 1$ and $-x_{kj} - x_{ij} + 2y_{kij} \leq 0$, to filter out the four faulty permutations [Yang et al. 2013]. Then we reformulate the MIQP problem in Eq. (9) as follows:

$$\min_{\vec{\mathbf{x}},t} t \left| \begin{array}{l} \sum_{i=1}^{Q}(\xi_i + \tau_i \cdot \sum_{j\in J}(g_j(\vec{\mathbf{z}}) + (\gamma_j + \psi_j(\vec{\mathbf{x}})) \cdot x_{ij})) \leq t; \\ \sum_{j\in J}(g_j(\vec{\mathbf{z}}) + (\gamma_j - T_{ij}^r)x_{ij}) \leq \xi_i, \forall i \in I; \\ -x_{kj} - x_{ij} + 2y_{kij} \leq 0, ..., \\ x_{kj} + x_{ij} + y_{kij} \leq 1, \forall i, k \in I, k < i, \forall j \in J; \\ \sum_{j\in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0,1\}, \end{array} \right. \tag{12}$$

where $g_j(\vec{\mathbf{z}}) = c_j \sum_{i\in I} z_{ij}$; size of $\vec{\mathbf{z}}$ is same with that of $\vec{\mathbf{x}}$; z_{kj} is equal to x_{ij} if $k = i$, and y_{kij} if $k \neq i$; y_{kij} is equal to y_{ikj} in this scenario; and $\psi_j(\vec{\mathbf{x}})$ can be determined through a look-up table based on x_{ij}. Thus, Eq. (12) is reduced to an MILP problem, which can be solved much more efficiently than MIQP of similar scale [Fuchs et al. 2013]. We wish to note that the MIQP-to-MILP transformation approach also directly applies to the refinement of travel time prediction in Eq. (11).

5.3. Performance Improvement via Communication

In the proposed approach, we use historical expected travel time to evaluate the remaining path from the assigned road link to destination. Since the infrastructure agent is always located at an intersection, it can obtain real-time traffic conditions (i.e., travel time) on directly connected road links. It is thus reasonable for an infrastructure agent to communicate with neighboring infrastructure agents to obtain real-time traffic conditions further away. The expectation is that real-time traffic condition can better evaluate a route than the historical traffic condition. We use E (i.e., $E \in \mathbb{Z}_0^+$) to denote the number of communication hops, and there is no communication if $E = 0$. In Fig. 1, if $E = 1$, r_1 only communicates with its neighbors, e.g., r_2 and r_4. Thus it can obtain real-time traffic conditions on p_4, p_5, p_{12}, p_{10} and p_8, which can be used to evaluate the paths from r_2 and r_4 to destinations when r_1 performs route assignment for v_1, v_2 and v_3. As E increases to 2, r_1 is able to communicate with infrastructure agents one more hop away, e.g., r_3, thus r_1 can obtain real-time traffic conditions on p_6, p_7 and r_{14} as well. However, as the number of communication hops becomes larger, additional communication and storage costs also incur. The dynamics of traffic may also cause real-time traffic information to be outdated by the time vehicle agents reach the intersection, if the location is far away.



(a) Singapore

(b) New York

Fig. 3: Two Testing Road Networks

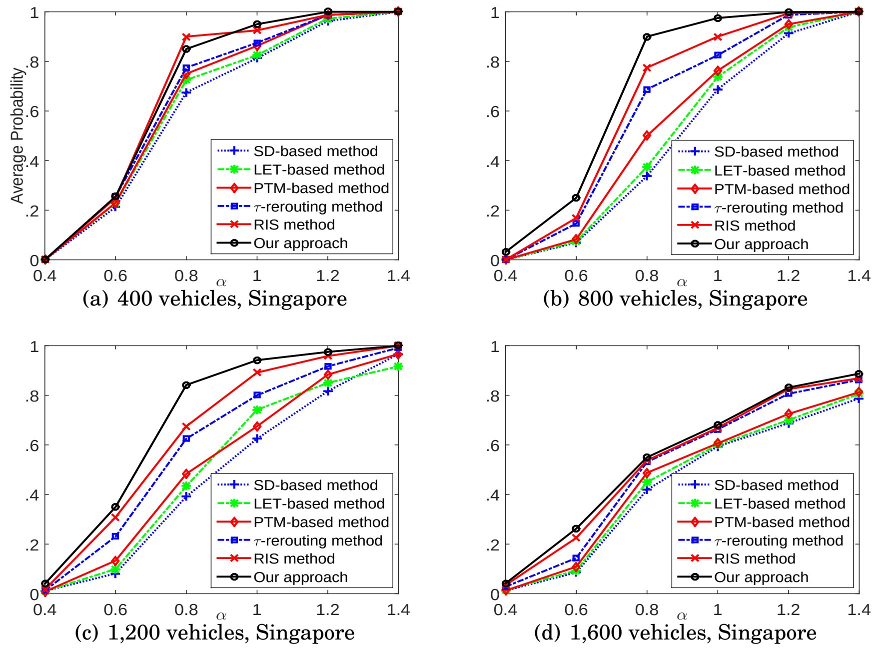
6. EXPERIMENTATION

In this section, we conduct experiments in various settings to extensively compare our route guidance approach with existing methods, showing its advantages of increasing the chances of reaching destination before deadline and reducing total travel time for all vehicles. Accordingly, we first introduce the experimental settings, then we focus on evaluating the performance of arriving on time. After that, we test the performance when considering both arriving on time and total travel time. Finally, we verify all the improvements (proposed in Section 5).

6.1. Road Networks and Parameter Settings

All experiments are conducted on the popular simulation of urban mobility platform, SUMO [Behrisch et al. 2011]. The two testing road networks are parts of two very dense cities, Singapore and New York respectively. Each road has 2 lanes, and their maps are given in Fig. 3, with the following properties summarized by SUMO: (1) network areas are 65,300m^2 and 218,000m^2; (2) numbers of road links are 507 and 1,121; (3) numbers of intersections are 98 and 352.

The configurations of vehicles are as follows: length is 5m; minimal gap is 2.5m; car following model is Krauss [Behrisch et al. 2011]; origins and destinations are randomly generated; traffic light duration: $T_g = T_r = 20s$; vehicles will park and not occupy road resources when reaching destinations. In addition, once origin and destination of a vehicle are determined, an expected travel time T_e can be derived based



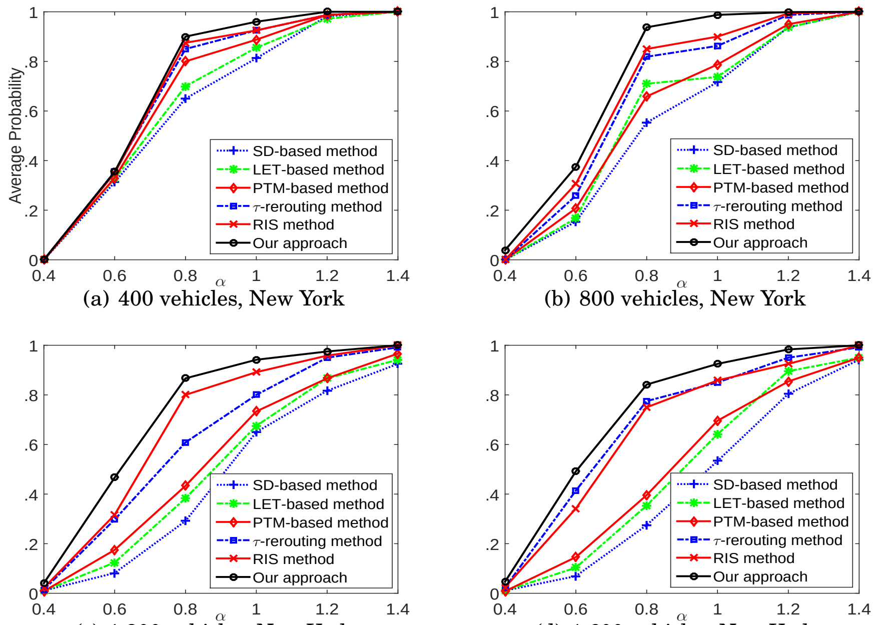
(a) 400 vehicles, Singapore (b) 800 vehicles, Singapore (c) 1,200 vehicles, Singapore (d) 1,600 vehicles, Singapore

Fig. 4: Different Types of Deadlines on Singapore Map. [**Note:** α- deadline coefficient.]

on historical traffic data, which can be used to describe different levels of deadlines as in Eq. (5). Moreover, the proposed approach needs historical expected travel time to evaluate some parts of a route. Therefore, before testing our approach, we first randomly run the simulation for 250 times to get an expected travel time of each road link, where vehicles simply travel along the shortest distance routes. Additionally, we also use SVR (i.e., support vector regression) [Chang and Lin 2011] to learn the linear and non-linear functions of predicted travel time through those random simulations, which are described in Section 5.1. Particularly, all experiments in this paper are conducted on an ordinary PC with Intel Core i7-3540M processor and 8.00 GB RAM.

6.2. Comparative Performance When only Considering Arriving on Time

To only consider the arriving on time property, we set all τ_i in Eq. (6) as 0, which is equal to the route assignment in Eq. (4). Then we compare our algorithm with five different route guidance methods: (1) SD (i.e., shortest distance) based method, which pre-computes a path of shortest distance; (2) LET-based method, which pre-computes a path of least expected travel time based on historical traffic conditions; (3) PTM-based method [Lim et al. 2013], which pre-computes a path by Eq. (1); (4) RIS (i.e., route information sharing) method, which constantly computes LET paths for vehicles at each intersection by cooperatively exploring their latest intentions of routes [Yamashita et al. 2005]; (5) τ-rerouting method, claimed to be the best-performing vehicle rerouting strategy [Jiang et al. 2014]. Note that the first three methods pre-compute route guidance before vehicle departure, while the last two and our approach adaptively provide route guidance for vehicles en-route. Although O-D pairs are randomly generated, for each specified O-D pair of a vehicle, we respectively adopt the six methods to provide route guidance.



(a) 400 vehicles, New York (b) 800 vehicles, New York

(c) 1,200 vehicles, New York (d) 1,600 vehicles, New York

Fig. 5: Different Types of Deadlines on New York Map. [**Note:** α- deadline coefficient.]

6.2.1. Different Levels of Deadlines. This experiment varies different levels of deadlines α: 0.4, 0.6, 0.8, 1.0, 1.2 and 1.4, with different numbers of vehicles: 400, 800, 1,200 and 1,600 on both networks. We run the simulation for 500 times under each setting, record the probability of arriving on time for each vehicle, and plot the average in Fig. 4 and Fig. 5 respectively. We can observe that on both networks, the average probabilities always increase with α for all six methods. It is natural because a vehicle with a very loose deadline has higher chance to arrive on time even if it does not follow any smart route guidance. Generally, the three pre-computation methods are inferior to the three adaptive methods, because the optimality of pre-computed paths may not hold, especially in highly dynamic traffic, e.g., New York network with 1,600 vehicles. However, the inferiority is not obvious in extremely sparse or saturated traffic, such as Fig. 4 (a), (d) and Fig. 5 (a). In sparse traffic, vehicles rarely influence each other, and shortest distance path is sufficiently satisfactory. In over-saturated traffic, vehicles almost cannot proceed even if they receive adaptive guidance. Among the three pre-computation methods, PTM-based method achieves the highest overall performance because it takes deadline into account, although in an independent manner. As for the three adaptive methods, our approach is always better than the other two in terms of overall probabilities of arriving on time, especially in Fig. 4 (b), (c), Fig. 5 (b), (c) and (d), where the traffic densities are moderate. In most cases, RIS method is better than τ-rerouting method, because it is centralized, where a global server constantly predicts the LET path for each individual vehicle, based on latest intentions of routes. This superiority does not hold for New York network with 1,600 vehicles, because the traffic density is comparatively high, and τ-rerouting method is especially effective where congestion is likely to occur. However, both methods do not care about whether vehicles would be late regarding their preferred deadlines. On the other hand, our approach cooperatively explores the deadlines of other vehicles, and recursively provides

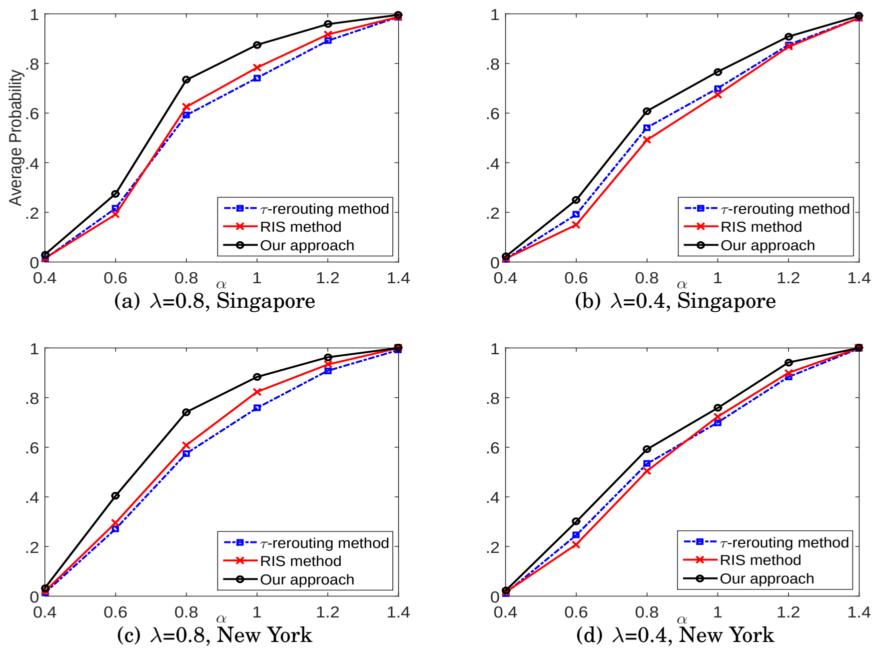


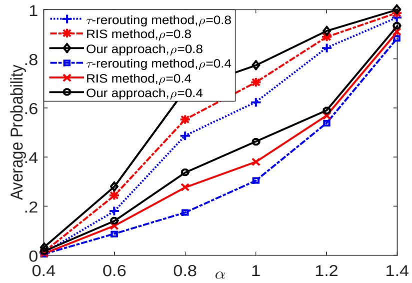
Fig. 6: Different Penetration Rates. [**Note:** α- deadline coefficent; λ- penetration rate.]

guidance by solving an optimization problem, which aims to guarantee arriving at destination before deadline for all concerned vehicles, thus achieving the best overall performance.

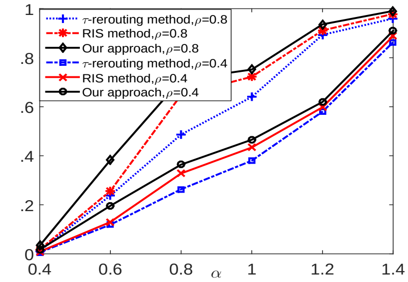
6.2.2. Different Penetration Rates. We test the three adaptive approaches[3] with different penetration rates λ defined as the percentage of vehicles sharing their intentions. We take both networks with 1,200 vehicles as study cases, and adopt two penetration rates, λ=0.8 and 0.4. From Fig. 6, we notice that average probabilities for the three methods decrease as λ becomes smaller. It is natural since route guidance in them all reply on intentions. Particularly, RIS method is centralized, and missing intentions of routes may globally influence route guidance of others, so τ-rerouting method and our approach achieve better performance regarding λ=0.4, especially for tight deadlines on both networks, i.e., from $\alpha = 0.6$ to 0.8. Although τ-rerouting method is decentralized, missing intentions may make infrastructure agents unable to report congestion timely, which causes more vehicles to miss their deadlines. On the other hand, in our approach, predicted travel time on assigned road link relies on both vehicle amount and road link length. Missing intentions only partially influences our approach. Moreover, our approach always takes deadline into account, thus it achieves best overall performance for different penetration rates.

6.2.3. Different Compliance Rates. We further test the three adaptive approaches with different compliance rates ρ defined as the percentage of vehicles following the route guidance by infrastructure agent. We take both networks with 1,200 vehicles as study cases, and adopt two compliance rates, ρ=0.8 and 0.4, the results of which are plotted in Fig. 7 (a) and (b) respectively. We would like to note that the three adaptive

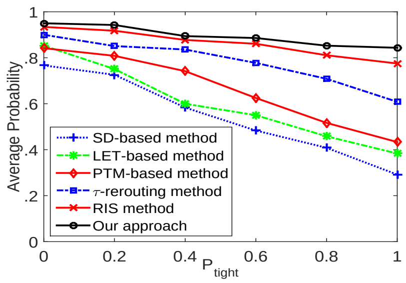
[3]The three pre-computation methods do not involve intention sharing, and it makes no sense to test penetration rates for them.



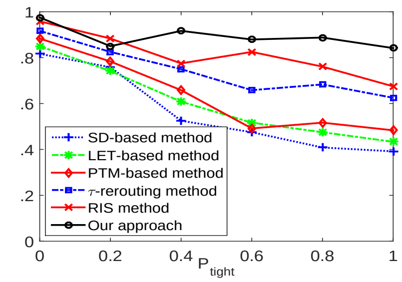
(a) 1,200 vehicles, Singapore



(b) 1,200 vehicles, New York



(c) 1,200 vehicles, Singapore

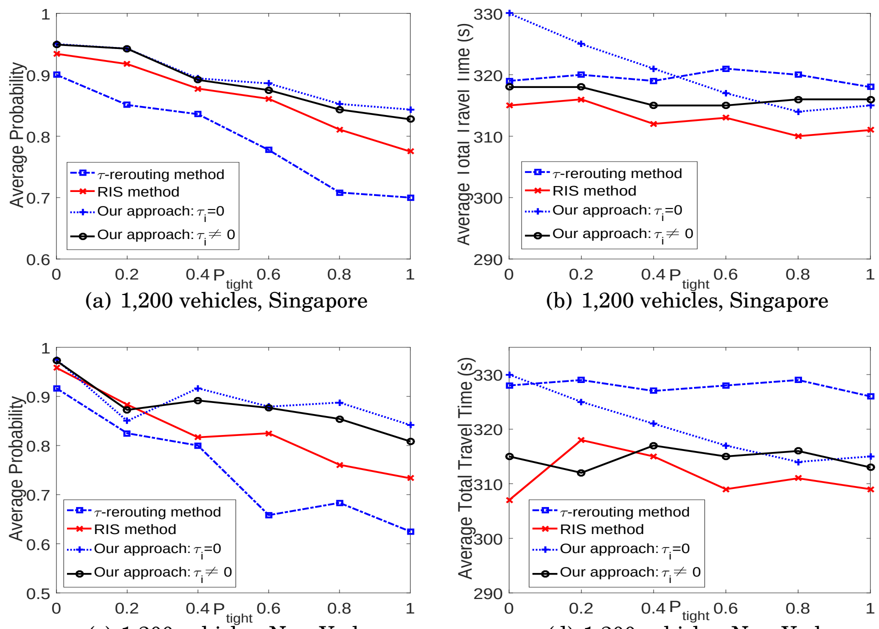


(d) 1,200 vehicles, New York

Fig. 7: Different Compliance Rates and Different Percentages of Tight Deadlines. [**Note:** α- deadline coefficent; P_{tight}- percentage of tight deadlines; ρ- compliance rate.]

approaches in Fig. 4 (c) and Fig. 5 (c) refer to ρ=1. Combining those figures together, we notice that average probabilities for the three methods on both networks decrease as ρ becomes smaller. However, our approach achieves highest overall performance for each setting because it takes arriving on time into account, although some of vehicles may not follow the guidance. We also observe that, τ-rerouting method in Fig. 7 (a) has lower probability, especially for α=0.8 and ρ=0.4. It happens because vehicle density is comparatively higher on Singapore network, where congestion is likely to happen. If most of vehicles do not follow the rerouting, they may have to stay in congestion for a longer time. Generally, our approach will achieve higher probability of arriving on time as ρ increases, which provides convincing incentives for vehicles to follow the route guidance.

6.2.4. Different Percentages of Tight Deadlines. We take both networks with 1,200 vehicles as study cases to further test our approach against different percentages of tight deadlines. In this situation, α is set as 0.8 for tight deadline, and 1.2 for loose deadline. Their percentages are P_{tight} and 1-P_{tight}. In Fig. 7 (c) and (d), as P_{tight} increases, the average probabilities for the three pre-computation methods drop more quickly on Singapore network because its traffic density is comparatively high, where vehicles always influence each other. There is only slight decrease for our approach on both networks, which is better than the other two adaptive methods. Although RIS method on Singapore network is competitive to our approach, we highlight that RIS method is centralized, becoming prohibitively time-consuming as network size and vehicle number scale up.



(a) 1,200 vehicles, Singapore (b) 1,200 vehicles, Singapore

(c) 1,200 vehicles, New York (d) 1,200 vehicles, New York

Fig. 8: Performance Considering both Arriving on Time and Total Travel Time. [**Note:** P_{tight}- percentage of tight deadlines; τ_i- weight for total travel time.]

6.3. Overall Performance When Considering both Arriving on Time and Total Travel Time

We further test our approach considering both the arriving on time and total travel time at the same time, where τ_i in Eq. (6) is 0 only if α_i is 0. To simplify the illustration, we only compare all the adaptive approaches. To show the advantages brought by considering both arriving on time and total travel time, we also compare it with the one that only considers arriving on time (i.e., we enforce τ_i to be 0). Moreover, we use the two networks with 1,200 vehicles as study cases. Note that $\tau_i \neq 0$ refers to the route assignment in Eq. (6), which considers both arriving on time and total travel time. And corresponding average probabilities of arriving on time and average total travel time against different percentages of tight deadlines are all plotted in Fig. 8.

From the four figures, we observe that the results for Singapore network and New York network share the similar pattern. Therefore, we emphasize on the analysis for Singapore network. From Fig. 8 (a), we notice that, as P_{tight} increases from 0 to 1.0, the average probability of arriving on time decreases for all methods. It is natural since the deadlines of most vehicle agents become more tight as P_{tight} increases. However, the average probabilities of arriving on time for $\tau_i = 0$ and $\tau_i \neq 0$ are much higher than those of the other two methods. It happens because, τ-rerouting method and RIS method do not take arriving on time into account. On the other hand, we notice that the average probabilities of $\tau_i = 0$ are higher than that of $\tau_i \neq 0$ as P_{tight} increases from 0.6 to 1. It happens because, all vehicle agents will reduce the total travel time to some extent, and those tight deadlines are likely to be missed due to influence from each other for $\tau_i \neq 0$. However, the difference of the probabilities is very slight for the two methods. On the other hand, from Fig. 8 (b), we observe that, comparing the method of $\tau_i \neq 0$ with $\tau_i = 0$, the average total travel time of the former is much lower than that of the latter, especially as P_{tight} increases from 0 to 0.6. That is because, the route assignment in Eq. (4) does not consider the total travel time. And if the deadlines of most vehicle agents are loose, Eq. (4) will randomly output a route as long as it guarantees arriving at the destination before the deadline, which is likely to be a detoured path and cause additional travel time. To the contrary, the second objective term in Eq. (6) will help to reduce the additional travel time while increasing the chance of arriving on time, especially for the case of loose deadlines. As P_{tight} continues to increase, the two methods will achieve similar performance, because in the case of tight deadlines, the method of $\tau_i = 0$ generally will not cause the additional travel time any more. We do not observe significant changes with P_{tight} for the τ-rerouting method and RIS method, because they do not provide route guidance based on the deadline. At the same time, we notice that, only the RIS method achieves slightly better performance than our method of $\tau_i \neq 0$ regarding the average total travel time, because the former is centralized, which relies on the global real-time information to directly reduce the total travel time. And it is more costly compared with our method of $\tau_i \neq 0$. Besides, the average probabilities of arriving on time for the RIS method is also lower than that of our method.

6.4. Improved Performance When Considering both Arriving on Time and Total Travel Time

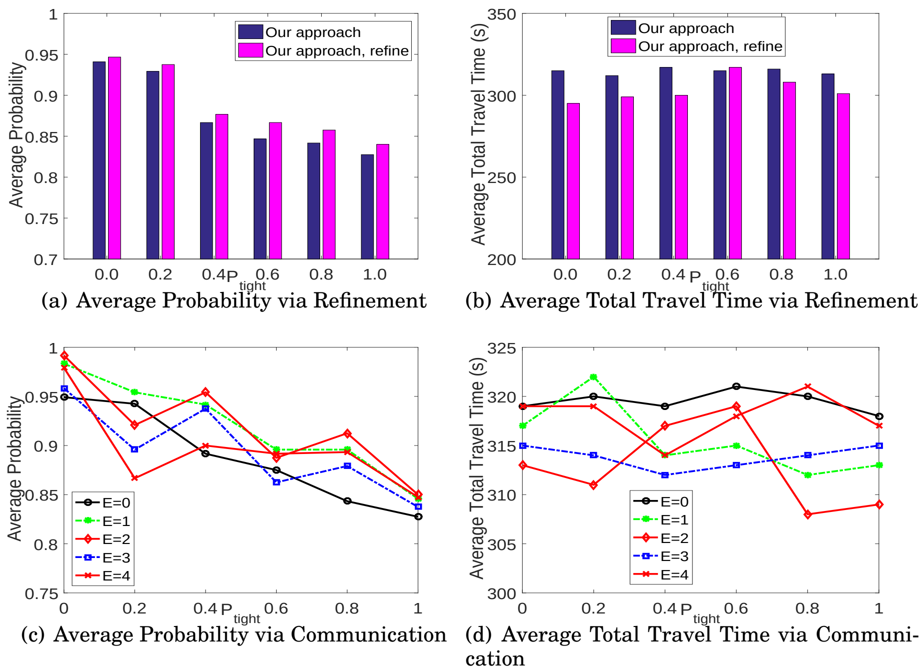
We conduct experiments to verify the improvements on travel time prediction, computation efficiency and real-time traffic condition acquirement proposed in Section 5.

6.4.1. Refinement of Travel Time Prediction. We proposed to refine the predicted travel time on the assigned road link by iteratively linearizing a more accurate learned function. We adopt the Singapore network with 1,200 vehicles as a study case, to investigate this improvement to the route assignment (i.e., the original route assignment is formulated as Eq. (6), and the refinement is formulated as Eq. (11)). All the results regarding the average probabilities of arriving on time and average total travel time are plotted in Fig. 9 (a) and (b) respectively.

From Fig. 9 (a) and (b) we observe that, the refinement of travel time prediction improves the average probability of arriving on time and reduces the average total travel time (except for $P_{tight} = 0.6$ regarding the total travel time[4]. Nevertheless, the difference is not significant, which is acceptable). This advantage comes from the fact that the proposed method approximates a non-linear function, which is more accurate than the previous linear prediction. We also notice that, compared with other values, the improvement for $P_{tight} = 0$ and $P_{tight} = 0.2$ regarding the average probabilities of arriving on time are not that obvious. It happens because most vehicles have loose deadlines for $P_{tight} = 0$ and $P_{tight} = 0.2$, and they still have higher chances of arriving on time although the infrastructure agents perform route assignment based on the inaccurate linear prediction of travel time. However, as P_{tight} increases, this improvement becomes more obvious. To the contrary, the reduction of the total travel time is much more obvious for $P_{tight} = 0, 0.2$ and 0.4 in comparison with others. It happens because, if most of the deadlines are loose, there would be much room to reduce the travel time, where a more accurate travel time prediction can better result in less total travel time compared with an inaccurate prediction function.

6.4.2. Improvement of Computation Efficiency. Original route assignment considering both arriving on time and total travel time is formulated as an MIQP problem in Eq. (6), and we reformulate it as an MILP problem in Eq. (12). To show the efficiency improvement, we use Pyomo (www.pyomo.org) to respectively solve the two problems regarding the same route assignment at each intersection, and record the average computation

[4]It might be caused by the fact that the refinement takes more effects on the arriving on time than the total travel time



(a) Average Probability via Refinement (b) Average Total Travel Time via Refinement

(c) Average Probability via Communication (d) Average Total Travel Time via Communication

Fig. 9: Refinement of Prediction and Improvement via Communication, Singapore. [**Note:** P_{tight}- percentage of tight deadlines; E- communication hop.]

time for both networks in Table I. We see that as vehicle number increases, the average computation time becomes longer for both problems. This happens because more vehicles are likely to request for route guidance at an intersection if traffic density is larger, thus the scales of the two optimization problems both increase, and longer computation time is needed. That also explains why the computation time on Singapore network is longer than that of New York network. However, for both networks, MILP problem can be more efficiently solved than MIQP problem of similar scale, especially for Singapore network with 1,600 vehicles, which is around 6 times faster.

Table I: Average Computation Time (s)

	Singapore				**New York**			
	400	800	1,200	1,600	400	800	1,200	1,600
MIQP	3.29	3.81	7.49	13.34	2.18	1.87	3.87	7.75
MILP	0.62	0.75	0.86	2.55	0.39	0.46	0.77	0.96

6.4.3. Improvement via Communication. To evaluate the benefits brought by communication, we test our approach against different communication hops (i.e., E). We again study the Singapore network with 1,200 vehicle agents for different percentages of tight deadlines. From Fig. 9 (c) and (d), we find that, $E = 1, 2$ achieve higher overall probabilities and shorter total travel time than that of $E = 0$ in most of cases, this is reasonable in that, if $E > 0$, our approach uses real-time traffic conditions to evaluate the first E road link(s) of the path from assigned road link to destination. If $E = 0$, it only uses historical traffic conditions. As E increases to 3 and 4, they do not achieve dominant performances over that of $E = 0$, because the traffic is always dynamic, and knowing real-time traffic conditions far away may not yield desirable route guidance [Amarante and Bazzan 2012]. Moreover, large communication hop also incurs additional cost to dynamically obtain and store traffic information. Therefore, $E = 1$ and $E = 2$ are sufficient to achieve satisfactory route guidance in our approach.

7. CONCLUSION AND FUTURE WORK

In this paper, we propose a decentralized multiagent-based route guidance approach to consider both arriving on time and total travel time. It is formulated as a route assignment problem at each road intersection by leveraging intentions of the vehicle agents. Besides, we also improve the proposed route guidance approach in the aspects of travel time prediction, computational efficiency and real-time traffic condition acquirement respectively. Experimental results confirm its superior performance over existing methods. As the chance of vehicles' arriving on time is increased, drivers' satisfaction gets improved, which also reduces accident rate due to drivers' frustration and impatience. At the same time, as the average total travel time is decreased, fuel consumption and air pollution can also be accordingly reduced. These are the important missions of the intelligent transportation and sustainable urban development.

In the future, we will develop a more intelligent algorithm to calculate the weight τ_i for total travel time, which can reduce the total travel time while not decreasing the chance of arriving on time. At the same time, we will also consider a personalized routing service, such as assigning customized weights to different vehicle agents according to their own preferences on arriving on time or total travel time. Besides, we will try road networks with less traffic lights or intersections, e.g., Paris and Berlin, to investigate the influence of network topology and density of infrastructure agents.

REFERENCES

Maicon de B. do Amarante and Ana L. C. Bazzan. 2012. Agent-based simulation of mobility in real-world transportation networks: effects of acquiring information and replanning en-route. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1351–1352.

Ana LC Bazzan and Franziska Klügl. 2014. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review* 29, 03 (2014), 375–403.

Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. 2011. SUMO–simulation of urban mobility: an overview. In *Proceedings of the Third International Conference on Advances in System Simulation*. 55–60.

Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

Emmanuel J Candes, Michael B Wakin, and Stephen P Boyd. 2008. Enhancing sparsity by reweighted 1 minimization. *Journal of Fourier analysis and applications* 14, 5-6 (2008), 877–905.

Zhiguang Cao, Hongliang Guo, Jie Zhang, and Ulrich Fastenrath. 2016a. Multiagent-based route guidance for increasing the chance of arrival on time. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI)*. 1180–1187.

Zhiguang Cao, Hongliang Guo, Jie Zhang, Dusit Niyato, and Ulrich Fastenrath. 2016b. Finding the shortest path in stochastic vehicle routing: a cardinality minimization approach. *IEEE Transactions on Intelligent Transportation Systems* 17, 6 (2016), 1688–1702.

Zhiguang Cao, Siwei Jiang, Jie Zhang, and Hongliang Guo. 2016c. A Unified Framework for Vehicle Rerouting and Traffic Light Control to Reduce Traffic Congestion. *IEEE Transactions on Intelligent Transportation Systems* PP, 99 (2016), 1–16. DOI: http://dx.doi.org/10.1109/TITS.2016.2613997

Chih-Chung Chang and Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2 (2011), 27:1–27:27. Issue 3.

Shan Shan Chen, Xin Zhao, and Yu Sheng. 2014. Research on Dynamic Route Guidance for an Emergency Vehicle Considering the Intersection Delay. In *Applied Mechanics and Materials*, Vol. 641. 848–852.

Mikhail V Chester. 2015. Can Disruptive Technologies, On-Demand Mobility, and Biofuels Improve Transportation Environmental Sustainability? A Review of Recent Research. *Current Sustainable/Renewable Energy Reports* 2, 3 (2015), 63–65.

Rutger Claes, Tom Holvoet, and Danny Weyns. 2011. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *IEEE Transactions on Intelligent Transportation Systems* 12, 2 (2011), 364–373.

YY Fan, RE Kalaba, and JE Moore II. 2005. Arriving on time. *Journal of Optimization Theory and Applications* 127, 3 (2005), 497–513.

John France, Ali Ghorbani, and others. 2003. A multiagent system for optimizing urban traffic. In *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT)*. 411–414.

Alexander Fuchs, Daniel Axehill, and Manfred Morari. 2013. Efficient evaluation of mp-MIQP solutions using lifting. *arXiv preprint arXiv:1311.4752* (2013), 1–23.

Jiarui Gan, Bo An, and Chunyan Miao. 2015. Optimizing efficiency of taxi systems: Scaling-up and handling arbitrary constraints. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. International Foundation for Autonomous Agents and Multiagent Systems, 523–531.

Hongliang Guo, Zhiguang Cao, Jie Zhang, Dusit Niyato, and Ulrich Fastenrath. 2014. Routing multiple cars in large scale networks: Minimizing road network breakdown probability. In *Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*. 2180–2187.

Siwei Jiang, Jie Zhang, and Yew-Soon Ong. 2014. A pheromone-based traffic management model for vehicle re-routing and traffic light control. In *Proceedings of international conference on Autonomous agents and multi-agent systems (AAMAS)*. 1479–1480.

Seung-Jean Kim, Kwangmoo Koh, Stephen Boyd, and Dimitry Gorinevsky. 2009. ℓ_1 Trend Filtering. *Siam Review* 51, 2 (2009), 339–360.

Jin Li, Qinmin Wu, and Daoli Zhu. 2009. Route Guidance Mechanism with Centralized Information Control in Large-scale Crowd's Activities. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 7–11.

Zilu Liang and Yasushi Wakahara. 2014. A route guidance system with personalized rerouting for reducing traveling time of vehicles in urban areas. In *Proceedings of IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*. 1541–1548.

Sejoon Lim, Hari Balakrishnan, David Gifford, Samuel Madden, and Daniela Rus. 2009. Stochastic motion planning and applications to traffic. In *Algorithmic Foundation of Robotics VIII*. 483–500.

Sejoon Lim, Christian Sommer, Evdokia Nikolova, and Daniela Rus. 2013. Practical route planning under delay uncertainty: Stochastic shortest path queries. *Robotics* (2013), 249–256.

Marin Lujak, Stefano Giordani, and Sascha Ossowski. 2015. Route guidance: Bridging system and user optimization in traffic assignment. *Neurocomputing* 151 (2015), 449–460.

Yu Marco Nie and Xing Wu. 2009. Shortest path problem considering on-time arrival probability. *Transportation Research Part B: Methodological* 43, 6 (2009), 597–613.

Evdokia Nikolova, Jonathan A Kelner, Matthew Brand, and Michael Mitzenmacher. 2006. Stochastic shortest paths via quasi-convex maximization. In *Algorithms–ESA 2006*. Springer, 552–563.

Jean Oh and Stephen F Smith. 2008. A few good agents: multi-agent social learning. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems-Volume 1*. International Foundation for Autonomous Agents and Multiagent Systems, 339–346.

J Pan, I Sandu Popa, Karine Zeitouni, and Cristian Borcea. 2013. Proactive Vehicular Traffic Re-routing for Lower Travel Time. *IEEE Transactions on Vehicular Technology* (2013), 3551–3568.

Markos Papageorgiou. 1990. Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research Part B: Methodological* 24, 6 (1990), 471–495.

Zachary Rubinstein, Stephen Smith, and Laura Barbulescu. 2012. Incremental Management of Oversubscribed Vehicle Schedules in Dynamic Dial-A-Ride Problems.. In *Proceedings 26th AAAI Conference on Artificial Intelligence (AAAI)*. 1–7.

Stephen F Smith, Anthony Gallagher, Terry L Zimmerman, Laura Barbulescu, and Zachary B Rubinstein. 2006. Multi-Agent Management of Joint Schedules.. In *AAAI Spring Symposium: Distributed Plan and Schedule Management*. 128–135.

Shen Wang, Soufiene Djahel, and Jennifer McManis. 2014a. A Multi-Agent based vehicles re-routing system for unexpected traffic congestion avoidance. In *Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*. 2541–2548.

Yilun Wang, Yu Zheng, and Yexiang Xue. 2014b. Travel time estimation of a path using sparse trajectories. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 25–34.

Mathijs M. de Weerdt, Enrico H Gerding, Sebastian Stein, Valentin Robu, and Nicholas R Jennings. 2013. Intention-aware routing to minimise delays at electric vehicle charging stations. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI)*. 83–89.

Christopher Wilt and Adi Botea. 2014. Spatially Distributed Multiagent Path Planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*. 332–340.

Tomohisa Yamashita, Kiyoshi Izumi, Koichi Kurumatani, and Hideyuki Nakashima. 2005. Smooth traffic flow with a cooperative car navigation system. In *Proceedings of the 4th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 478–485.

Rong Yang, Albert Xin Jiang, Milind Tambe, and Fernando Ordonez. 2013. Scaling-up security games with boundedly rational adversaries: A cutting-plane approach. In *Proceedings of the Twenty-Third international joint conference on Artificial Intelligence (IJCAI)*. 404–410.

Jing Yuan, Yu Zheng, Chengyang Zhang, Wenlei Xie, Xing Xie, Guangzhong Sun, and Yan Huang. 2010. T-drive: driving directions based on taxi trajectories. In *Proceedings of the 18th SIGSPATIAL International conference on advances in geographic information systems*. 99–108.

Fuzheng Zhang, Nicholas Jing Yuan, David Wilkie, Yu Zheng, and Xing Xie. 2015. Sensing the pulse of urban refueling behavior: A perspective from taxi mobility. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6, 3 (2015), 37:1–37:23.

Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. 2014. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)* 5, 3 (2014), 38:1–38:55.

Yu Zheng and Xing Xie. 2011. Learning travel recommendations from user-generated GPS traces. *ACM Transactions on Intelligent Systems and Technology (TIST)* 2, 1 (2011), 2:1–2:29.

Mortaza Zolfpour-Arokhlo, Ali Selamat, Siti Zaiton Mohd Hashim, and Hossein Afkhami. 2014. Modeling of route planning system based on Q value-based dynamic programming with multi-agent reinforcement learning algorithms. *Engineering Applications of Artificial Intelligence* 29 (2014), 163–177.