

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

2-2016

Multiagent-based route guidance for increasing the chance of arrival on time

Zhiguang CAO

Singapore Management University, zgcao@smu.edu.sg

Hongliang GUO

Jie ZHANG

Ulrich FASTENRATH

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#)

Citation

CAO, Zhiguang; GUO, Hongliang; ZHANG, Jie; and FASTENRATH, Ulrich. Multiagent-based route guidance for increasing the chance of arrival on time. (2016). *Proceedings of the 30th AAAI Conference on Artificial Intelligence, Arizona, United States of America, 2016 Feb 12-17*. 3814-3820.

Available at: https://ink.library.smu.edu.sg/sis_research/8130

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Multiagent-Based Route Guidance for Increasing the Chance of Arrival on Time

Zhiguang Cao^{1,2}, Hongliang Guo^{1,2}, Jie Zhang¹, and Ulrich Fastenrath³

¹School of Computer Engineering, Nanyang Technological University, Singapore

²Energy Research Institute @NTU, Nanyang Technological University, Singapore

³Department of Traffic Information Management and Routing Optimization, BMW Group, Germany
{caoz0005, guohl, zhangj}@ntu.edu.sg, Ulrich.Fastenrath@bmw.de

Abstract

Transportation and mobility are central to sustainable urban development, where multiagent-based route guidance is widely applied. Traditional multiagent-based route guidance always seeks LET (least expected travel time) paths. However, drivers usually have specific expectations, i.e., tight or loose deadlines, which may not be all met by LET paths. We thus adopt and extend the probability tail model that aims to maximize the probability of reaching destinations before deadlines. Specifically, we propose a decentralized multiagent approach, where infrastructure agents locally collect intentions of concerned vehicle agents and formulate route guidance as a route assignment problem, to guarantee their arrival on time. Experimental results on real road networks justify its ability to increase the chance of arrival on time.

Introduction

Route guidance for vehicles is a challenging problem in transportation and mobility, which is crucial to the sustainable development of any city. It attracts broad and deep attention from the government, industry and research community due to its high relevance to people's daily life. Multiagent-based approaches are widely applied in route guidance (France, Ghorbani, and others 2003; Wilt and Botea 2014), because agent metaphor for modeling a traffic participant or decision-maker can capture complex constraints connecting all problem-solving phases, especially in the cooperative vehicle routing (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 infrastructure agents at road intersections (Jiang, Zhang, and Ong 2014).

Although multiagent-based approaches have achieved big success in route guidance, a critical issue still remains to be addressed: most of them only seek LET (least expected travel time) paths for vehicles (Yamashita et al. 2005; Claes, Holvoet, and Weyns 2011; Wang, Djahel, and McManis 2014). The attractive aspect about LET path is that it can be transformed into a deterministic routing problem

and solved by efficient path finding algorithms like Dijkstra (Hall 1986). However, 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 catch up important appointments, deadlines might be tight; if they go shopping, deadlines might be loose. Simply seeking LET paths for all drivers is not necessary and may cause some drivers with tight deadlines to miss their deadlines due to the influence from other drivers with loose deadlines. This will then increase these drivers' frustration and impatience, and in consequence the accident rate.

To address this issue, we propose to adopt the probability tail model which aims to maximize the probability of reaching destination before a deadline (Fan, Kalaba, and Moore II 2005; Lim et al. 2013). This model is more reasonable as it considers specific demands of drivers and is consistent with real-world travel behavior. One common query could be that "I want to reach airport in 40 minutes. Please find a path with maximum chance". However, the probability tail model is originally designed for each single vehicle, which independently pre-computes a path before each vehicle departs. Traffic is known to be dynamic, so optimality of a pre-computed path may not hold once all vehicles are en-route. It is then desirable to extend the probability tail model to consider the intentions of other vehicles, with the purpose of increasing the chance of arrival on time for all vehicles. To achieve this, we propose a decentralized multiagent approach, where infrastructure agents locally collect intentions of concerned vehicle agents and formulate route guidance as a route assignment problem, to guarantee their arrival on time. Furthermore, its efficiency is enhanced by reformulating route assignment as a mixed integer linear programming (MILP) problem, and its performance is improved by allowing communication between neighboring infrastructure agents. Experimental results on real road networks show that our approach outperforms the traditional methods.

Related Work

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 inter-

section accordingly. 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. In another centralized approach (de B do 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. Decentralized multiagent approaches have the capability of adaptively updating routes according to dynamic traffic, however they do not consider specific demands of vehicle agents, i.e., preferred deadlines.

Several optimization approaches are proposed to take drivers' preferred deadlines into account. Particularly, the probability tail model is widely adopted for vehicle route guidance (Fan, Kalaba, and Moore II 2005; Lim et al. 2009; 2013), which aims to maximize the probability of arriving at destination before deadline and is formally expressed as (Lim et al. 2013; Cao et al. 2014):

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

where \vec{w} denotes travel time for each road link; \mathbf{M} is node-arc incidence matrix of the road network; T is the preferred deadline; \vec{b} is an O-D vector, all elements of which are zeros except those for origin ("1") and destination ("-1"); \vec{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{x} is a connected path from origin to destination. This model is more consistent with real-world travel queries. However the existing approaches incorporating this model independently pre-computes a path for each individual vehicle before it departs, without considering the intentions of others. 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 of others.

It is thus desirable to leverage both the advantages of a decentralized multiagent approach (that deals with traffic dynamics by considering intentions of vehicle agents) and the probability tail model (that considers different deadlines of vehicles), which is what we propose in this paper.

Multiagent-based Route Guidance

We propose a decentralized multiagent-based approach involving two types of agents, vehicle agents and infrastructure agents. Vehicle agents representing drivers, travel on a road network by following route guidance. Infrastructure agents, located at road intersections, collect intentions (i.e., deadlines and destinations) from vehicle agents, and formulate route guidance as a route assignment problem, to increase the chance of arrival on time for all concerned vehicle agents. Then the vehicle agents accordingly update routes.

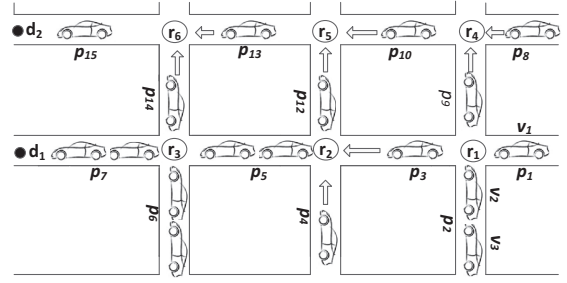


Figure 1: Two Types of Agents and Intention Collection

Intention Collection

In traffic, vehicles may influence each other due to limited road capacity and rush hour effect. To consider such influence, intention collection is necessary (Yamashita et al. 2005; Li, Wu, and Zhu 2009; Claes, Holvoet, and Weyns 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.

Take a one-way road network in Fig. 1 as an example, where r_i is infrastructure agent, v_j is vehicle agent, and p_k is 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.

Route Assignment

Vehicles generally have different types of deadlines. Simply seeking LET paths for all vehicles may cause some vehicles with tight deadlines to miss their deadlines due to the influences from other vehicles with loose deadlines. Motivated by this concern, a desirable approach is to distribute vehicles with loose deadlines to detour crowded paths, to make ways to those with tight deadlines. On the other hand, vehicles 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 incorporates the collected intentions.

Take infrastructure agent r_1 and vehicle agents v_1 , v_2 and v_3 in Fig. 1 again 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_{13}^p and

T_{19}^p , 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_{22}^e and T_{42}^e . 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_{13}^r and T_{19}^r . Since deadlines should always be non-negative, we have $T_{13}^r = \max\{0, T_1 - T_{22}^e\}$, and $T_{19}^r = \max\{0, T_1 - T_{42}^e\}$. Thus there would be a potential delay ξ_1 for v_1 , satisfying $T_{13}^p - T_{13}^r \leq \xi_1$ and $T_{19}^p - T_{19}^r \leq \xi_1$ (i.e., ξ_1 is non-negative, and there is no delay only if $\xi_i = 0$). To guarantee arrival on time for the three vehicle agents, r_1 should minimize the cardinality¹ 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) $I = \{1, \dots, Q\}$ and $J = \{1, \dots, L\}$, indexes of vehicle agents and road link choices associated with the infrastructure agent. Then maximizing the chance of arrival on time for all concerned vehicle agents can be expressed as minimizing the cardinality of the potential delay $\vec{\xi}$, as follows:

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

where $\vec{\xi} = \{\xi_1, \dots, \xi_Q\}$, and delay occurs for v_i if $\xi_i > 0$; $\vec{x}_j = (x_{1j}, \dots, x_{Qj})$, indicates assignments to l_j ; $f_j(\vec{x})$, a linear function, denotes predicted travel time on l_j ; T_{ij}^r 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 $\sum_{j \in J} (f_j(\vec{x}) - T_{ij}^r) \cdot x_{ij}$. Particularly, the linear function $f_j(\vec{x})$ for l_j is expressed as:

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

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:

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

Eq. (4) is a mixed integer quadratic programming (MIQP) problem in nature, which can be solved by existing solvers. Since ℓ_1 -norm minimization in Eq. (4) is minimizing the sum of delay duration, our approach actually also reduces the delay duration for individual vehicles.

¹Cardinality is the number of non-zero components in a vector.

input : $V = \{v_1, \dots, v_Q\}$, a set of vehicle agents;
 $R = \{r_1, \dots, r_G\}$, a set of infrastructure agents;
 $V^i = \emptyset$, vehicle agents that need guidance at r_i ;
 TL_i , traffic lights associated with r_i ;
 T_g, T_r , green light and red light duration;
 $flg = 0$, indicator of route guidance computation;

```

1 Each  $r_i \in R$  turns on its associated traffic lights  $TL_i$ ;
2 Each  $v_j \in V$  determines destination  $d_j$  and deadline  $T_j$ , and computes an initial path  $P_j$  using Eq. (1);
3 while  $|V| > 0$  do
4   foreach  $v_j \in V$  do
5     if  $v_j$  reaches  $d_j$  then
6       Deletes itself from  $V$ :  $V = V - v_j$ ;
7     else
8       Travels along  $P_j$ ;
9       Updates deadline  $T_j$ ;
10  foreach  $r_i \in R$  do
11    foreach  $tl_k \in TL_i$  do
12      Runs according to  $T_g$  and  $T_r$ ;
13      if  $tl_k$  is in red color phase then
14         $r_i$  finds  $v_k$  facing  $tl_k$ :  $V^i = V^i + v_k$ ;
15         $r_i$  collects  $d_k$  and  $T_k$  from  $v_k$ ;
16      if  $tl_k$  is at end of red color phase then
17         $flg = 1$ ;
18      if  $flg == 1$  then
19        foreach  $v_j \in V^i$  do
20          Computes relative deadline for  $v_j$  based on the latest  $T_j$ ;
21          Computes its new path  $P_j'$  via Eq. (4);
22          Updates its route:  $P_j = P_j'$ ;
23          Resets parameters:  $V^i = \emptyset, flg = 0$ ;

```

Algorithm 1: Multiagent-based Route Guidance

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_{ij}^r . 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.

Pseudo-Code Summary

We summarize the proposed multiagent-based route guidance approach in Algorithm 1. Lines 1-2 initialize infrastructure agents and vehicle agents. In Lines 3-23, each infrastructure agent recursively assigns paths to vehicle agents

who need route guidance at intersections, until they all reach destinations. Particularly, in Lines 4-9, each vehicle agent travels along a current route and updates its deadline if it has not reached destination. In Lines 11-17, 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 18-22, 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 Eq. (4), and accordingly updates their routes.

Further Improvements

We further improve computational efficiency of our approach by introducing new variables and additional linear constraints to Eq. (4), and route guidance performance by allowing communications between infrastructure agents.

Improvement on Computational Efficiency

Eq. (4) is a MIQP problem mainly due to $(f_j(\vec{x}) - T_{ij}^r) \cdot x_{ij}$ in the first constraint. After unfolding, quadratic part comes from the term $x_{kj} \cdot x_{ij}$ ($k \in I$, and x_{kj} denotes assignment to l_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. (4) as follows:

$$\min_{\vec{x}} \sum_{i=1}^Q \xi_i \begin{cases} \sum_{j \in J} (g_j(\vec{z}) + (\gamma_j - T_{ij}^r) x_{ij}) \leq \xi_i, \forall i \in I; \\ -x_{kj} - x_{ij} + 2y_{kij} \leq 0, \dots, \\ x_{kj} + x_{ij} + y_{kij} \leq 1, \forall i, k \in I, k < i, \forall j \in J; \\ \sum_{j \in J} x_{ij} = 1, \forall i \in I; \xi_i \geq 0; x_{ij} \in \{0, 1\}, \end{cases} \quad (5)$$

where $g_j(\vec{z}) = c_j \sum_{i \in I} z_{ij}$; size of \vec{z} is same with that of \vec{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. Thus, Eq. (5) is reduced to a mixed integer linear programming (MILP) problem, which can be solved much more efficiently than MIQP of similar scale.

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.

Experimentation

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 for all vehicles.

Road Networks and Parameter Settings

All experiments are conducted on 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 65,300m² and 218,000m²; (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 not occupy road resources when reaching destinations. In addition, we introduce a positive parameter α to denote different levels of deadlines. Specifically, once O-D are determined, an expected travel time T_e can be derived based on historical traffic data. Thus, $T = \alpha \cdot T_e$ implies a tight deadline if $\alpha < 1$, and a loose deadline if $\alpha > 1$. Moreover, the proposed approach needs historical expected travel time to evaluate some parts of a route. So, 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 (Chang and Lin 2011) to learn the coefficients of $f_j(\vec{x})$ through those random simulations. Particularly, all experiments are conducted on an ordinary PC with Intel Core i7-3540M processor and 8.00 GB RAM.

Comparative Performance

We compare 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,

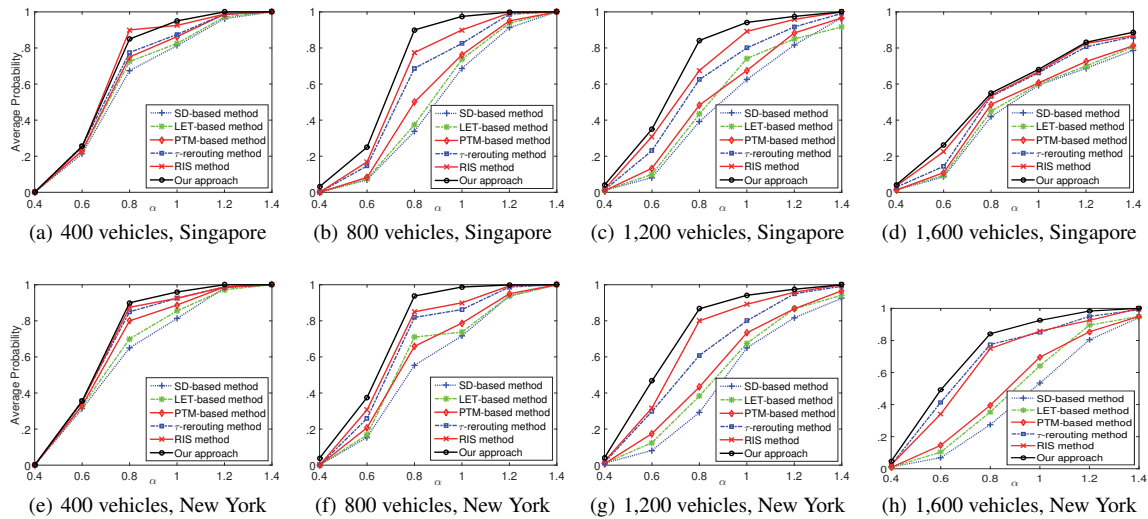


Figure 2: Different Types of Deadlines. [Best Viewed in Color]

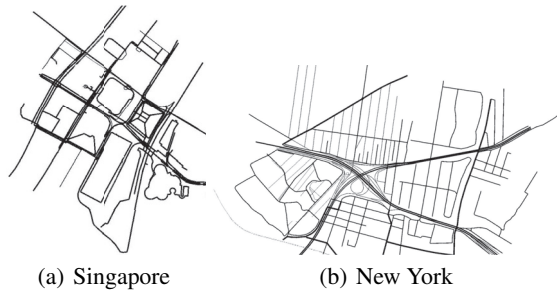


Figure 3: Two Testing Road Networks

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, Zhang, and Ong 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.

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 arrival on time for each vehicle, and plot the average in Fig. 2. 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 vehi-

cles. However, the inferiority is not obvious in extremely sparse or saturated traffic, such as Fig. 2 (a), (e) and (d). 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. 2 (b), (c), (f), (g) and (h), 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 guidance by solving an optimization problem, which aims to guarantee arrival on time, thus achieving the best overall performance.

Different Penetration Rates We test the three adaptive approaches 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 $\lambda=0.8$ and 0.4. From Fig. 4, 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 $\lambda=0.4$, especially for

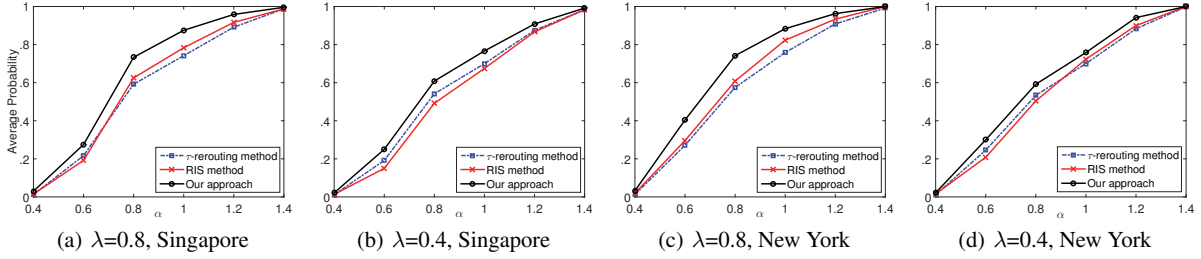


Figure 4: Different Penetration Rates. [Best Viewed in Color]

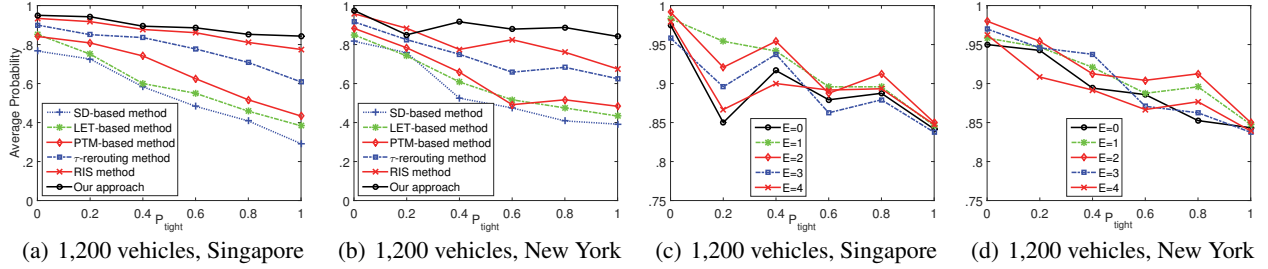


Figure 5: Different Percentages of Tight Deadlines and Improvement via Communication. [Best Viewed in Color]

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, 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.

Different Percentages of Tight Deadlines We adopt both networks with 1,200 vehicles to further test our approach against different percentages of tight deadlines, where 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. 5 (a) and (b), 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.

Improved Performance

We also conduct experiments to confirm the improvements on computation efficiency and route guidance performance proposed in the ‘‘Further Improvements’’ section.

Improvement of Computation Efficiency Original route assignment is formulated as an MIQP problem in Eq. (4), and we reformulate it as an MILP problem in Eq. (5).

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 time for both networks in Table 1. 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 10 times faster.

Table 1: Average Computation Time (s)

	Singapore				New York			
	400	800	1,200	1,600	400	800	1,200	1,600
MIQP	2.31	3.19	7.42	11.61	1.16	1.96	3.29	6.98
MILP	0.38	0.58	0.95	1.19	0.21	0.34	0.62	0.72

Improvement via Communication To evaluate the benefits brought by communication, we test our approach against different communication hops (i.e., E). We again study both networks with 1,200 vehicles for different percentages of tight deadlines. From Fig. 5 (c) and (d), we find that, $E = 1, 2$ on Singapore network, and $E = 1, 2, 3$ on New York network always achieve higher overall probabilities than that of $E = 0$, 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, e.g., 3 and 4 on Singapore network, and 4 on New York network, 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 (de B do 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.

Conclusion

We proposed a decentralized multiagent-based route guidance approach to enhance the chances of reaching destination before deadline for vehicles. It is formulated as a route assignment problem at each road intersection by leveraging intentions of the vehicles. Experiments confirm its superior performance over existing methods. As the chance of vehicles' arrival on time is increased, drivers' satisfaction gets improved, which also reduces accident rate due to drivers' frustration and impatience, an important mission of intelligent transportation and sustainable urban development.

Acknowledgment

This work is supported by the BMW Group, Germany, and also partially supported by the MOE AcRF Tier 1 funding (M4011261.020) awarded to Dr. Jie Zhang. Hongliang Guo and Zhiguang Cao are corresponding authors to this paper.

References

- Bazzan, A. L., and Klügl, F. 2014. A review on agent-based technology for traffic and transportation. *The Knowledge Engineering Review* 29(03):375–403.
- Behrisch, M.; Bieker, L.; Erdmann, J.; and Krajzewicz, D. 2011. SUMO – simulation of urban mobility: an overview. In *Proceedings of the 3rd International Conference on Advances in System Simulation (SIMUL)*, 55–60.
- Cao, Z.; Guo, H.; Zhang, J.; Niyato, D.; and Fastenrath, U. 2014. A data-driven method for stochastic shortest path problem. In *Proceedings of the IEEE 17th International Conference on Intelligent Transportation Systems (ITSC)*, 1045–1052.
- Chang, C.-C., and Lin, C.-J. 2011. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2:27:1–27:27. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- Claes, R.; Holvoet, T.; and Weyns, D. 2011. A decentralized approach for anticipatory vehicle routing using delegate multiagent systems. *IEEE Transactions on Intelligent Transportation Systems* 12(2):364–373.
- de B do Amarante, M., and Bazzan, A. L. 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.
- Fan, Y.; Kalaba, R.; and Moore II, J. 2005. Arriving on time. *Journal of Optimization Theory and Applications* 127(3):497–513.
- France, J.; Ghorbani, A.; et al. 2003. A multiagent system for optimizing urban traffic. In *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology (IAT)*, 411–414.
- Hall, R. W. 1986. The fastest path through a network with random time-dependent travel times. *Transportation science* 20(3):182–188.
- Jiang, S.; Zhang, J.; and Ong, Y.-S. 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.
- Kim, S.-J.; Koh, K.; Boyd, S.; and Gorinevsky, D. 2009. ℓ_1 trend filtering. *Siam Review* 51(2):339–360.
- Li, J.; Wu, Q.; and Zhu, D. 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.
- Lim, S.; Balakrishnan, H.; Gifford, D.; Madden, S.; and Rus, D. 2009. Stochastic motion planning and applications to traffic. In *Algorithmic Foundation of Robotics VIII*. Springer. 483–500.
- Lim, S.; Sommer, C.; Nikolova, E.; and Rus, D. 2013. Practical route planning under delay uncertainty: Stochastic shortest path queries. *Robotics* 249.
- Pan, J.; Sandu Popa, I.; Zeitouni, K.; and Borcea, C. 2013. Proactive vehicular traffic re-routing for lower travel time. *IEEE Transactions on Vehicular Technology*, 3551–3568.
- Papageorgiou, M. 1990. Dynamic modeling, assignment, and route guidance in traffic networks. *Transportation Research Part B: Methodological* 24(6):471–495.
- Wang, S.; Djahel, S.; and McManis, J. 2014. 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.
- Wilt, C., and Botea, A. 2014. Spatially distributed multiagent path planning. In *Proceedings of the 24th International Conference on Automated Planning and Scheduling (ICAPS)*.
- Yamashita, T.; Izumi, K.; Kurumatani, K.; and Nakashima, H. 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.
- Yang, R.; Jiang, A. X.; Tambe, M.; and Ordóñez, F. 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.