Traffic Management for Last-Mile Public Transportation Systems Using Autonomous Vehicles

Koji Hasebe* Kazuhiko Kato* Hirotake Abe* Ryutaro Akiya* Masayuki Kawamoto†

*Department of Computer Science, University of Tsukuba
†International Innovation Interface Organization, University of Tsukuba
1-1-1 Tennodai, Tsukuba 305-8573, Japan

***Abstract*—We introduce our research project aimed at developing a last-mile public transportation system with autonomous vehicles. In particular, we focus attention on traffic control techniques. Our vehicles can form fleets of autonomous vehicles, each of which consists of a lead vehicle driven manually and some driverless vehicles running behind another vehicle. In our system, the transport requests of all passengers are aggregated in real time to a central management server, and according to the information received, an optimized operation schedule is dynamically determined. Based on this system, we present a technique that by reorganizing vehicles does not require passengers to transfer to other fleets. In addition, to ensure fault tolerance of the system against failure of the central server, we propose a method to automatically switch to a provisional operation schedule for continuing transportation service. We also describe a scheme to implement our vehicle operation management adopting techniques of parallel computing and virtualization.**

***Index Terms*—Last-mile transportation, vehicle platooning, autonomous vehicle, traffic management, fault tolerance.**

I. INTRODUCTION

Enriching public transport plays a critical role in realizing the next generation smart city. Recently, the convenience of basic public transportation such as railroads and large-scale facilities is being improved, but the last-mile transportation network is still underdeveloped. Therefore, the creation and social implementation of a new transportation system with high convenience and accessibility for all people including elderly and disabled are required.

Many last-mile transportation systems are already in operation, and demonstration driving of its practical use has also been carried out in various places. Typical examples are ParkShuttle [1] running near Rotterdam in the Netherlands and CityMobil 2 [2] in La Rochelle, France. However, as platforms are often diversely dispersed, conventional transportation systems such as on-demand bus services suffer with problems of bringing passengers to stops where they can be picked up or dropped off. In addition, caused by elderly people and persons with disabilities who need more time to board or alight, stop time gets longer, and as a consequence, express services may be compromised.

To tackle these problems, the authors are developing a last-mile public transportation system with autonomous vehicles based on technologies of semi-autonomous driving. Our vehicles are divided into two types, those that lead a fleet and driven manually and those that are driverless and follow another vehicle. The greatest advantage is that compared with conventional transportation the rearrangement of fleets becomes easier.

Based on this technology, in this paper we propose a passenger transportation scheme that enables all passengers to reach their destinations without having to change vehicles and without having to wait for other passengers to board or alight.

In our system, we assume that the travel requests are aggregated in real time and the schedule of the vehicles is dynamically determined. To achieve such scheduling, our system has a single central server to performe all navigation management of the fleet by centralizing demand information, and instructs each vehicle from this central hub. However, if the central server malfunctions due to some unforeseen event, the entire traffic system may be stopped. To address this problem, we propose a method of automatically assigning a provisional route to each vehicle at the time of failure for maintaining the passenger transport service.

Finally, we illustrate the implementation of our vehicle operation management using techniques adopted from parallel computing and virtualization.

This paper is organized as follows. Section 2 overviews our transportation system and presents the proposed vehicle operation management. Section 3 explains the mechanism by which the transport service can continue against failure of the central server. Section 4 describes our implementation. Finally, Section 5 concludes the paper and discusses future research.

II. DESIGN OF THE TRANSPORTATION SYSTEM

A. Overview of the Autonomous Vehicles

Our transportation system consists of vehicles that are able to run in a row without having physical contact points. A conceptual picture is shown in Fig. 1. (See also Fig. 2 for a running test with golf carts as our prototype vehicles.)

The vehicles are classified according to their function into two types.

- *Lead vehicles*: vehicles driven manually at the head of a fleet of vehicles.
- *Trailing vehicles* (or *trailers*, for short): driverless vehicles that run autonomously behind another vehicle.

Vehicles can be arbitrarily connected through electronic control to a fleet as long as the fleet capacity, i.e. the maximum

978-1-5386-2524-8/17/$31.00 © 2017 IEEE

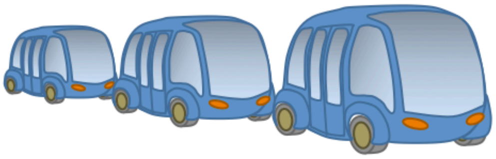

Fig. 1. Concept of a fleet of passenger-carrying vehicles.

Fig. 2. Traveling experiment with golf carts as prototype vehicles.

number of vehicles that can be connected to a lead vehicle, is not exceeded. Also, passengers can board any type of vehicle that has not already reached passenger capacity.

B. Traffic Network

In our transportation system, as for existing railroad and bus networks, the travel routes connect a plurality of prescribed stops (stations) where passengers may board or alight. Also, in this travel route network, several areas are provided depending on geographic location where stops overlap with one or more areas.

A concrete example of an operation route network is shown in Fig. 3. There are multiple stops in each area (circles and squares in the figure); a travel route passing through all the stops in an area is set for each area (arrow in the figure). Also, some stops (squares in the figure) are present in multiple areas. This is called a *node*. (In this example, there are two nodes indicated by "1" and "2".) There may be a plurality of nodes in one area as depicted in area B.

Here we note that travel routes in an area do not necessarily have to connect only different stops, but may include a route in which the same stop appears at several times, as shown in (A) of Fig. 4. Also, as shown in (B) of Fig. 4, a route that takes in all stops in the area need not be completed all at once. In the example of this figure, there are two separate routes circulating in order 1-2-3-4 and in order 1-5-6-7.

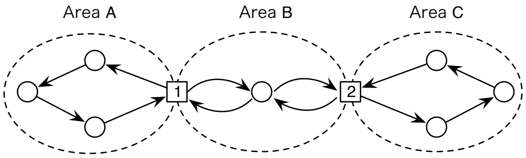

Fig. 3. Example of a transportation network with stops, nodes, and travel routes.

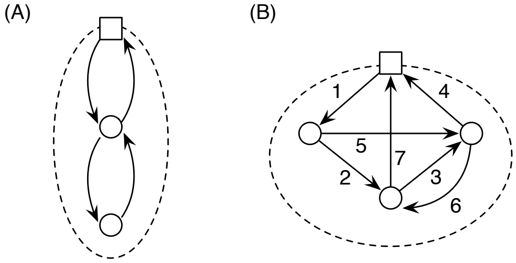

Fig. 4. Examples of route in an area.

C. Reorganization of Vehicles

Lead vehicles can disconnect any trailers at a node. In addition, a location for a *vehicle pool* is provided at each node for temporary parking of detached unmanned lead vehicles. Also, a lead vehicle can be provided with any number of trailers from the vehicle pool, as long as the fleet capacity.

D. Destination Board

All lead vehicles and trailers have a destination board, on which one of the area names in the traffic network can be displayed. In addition, the destination board also has a sign "Full" to notify waiting passengers that the vehicle is full. Vehicles in the same fleet may display different area names on the destination board, which can only be changed when the current position satisfies the following conditions.

- The current position is a node.
- The area in which the vehicle is currently located matches the area indicated on the destination board.

However, when the vehicle reaches capacity at a stop, the display on the destination board would be switched to "Full".

E. Passenger boarding and alighting

Passengers can travel along any route in the network up until the announced destination displayed on the vehicle's board. Hence, when boarding a vehicle, the passenger can only ride to the stop displayed. In particular, when the departure point and the destination point are in the same area, passengers can only ride on vehicles displaying that area. Moreover, no waiting passenger can board a vehicle displaying "Forwarded" or "Full" signs.

F. Vehicle Traffic Management

To realize our vehicle traffic management scheme, the operation schedule needs to be arranged according to the current requests and delivered to each vehicle. Here we assume a certain centralized control. Specifically, we set up a server for vehicle operation management (hereinafter referred to as the *central server*) in the system's hub, receiving requests for travel from passengers using smartphones or other means of

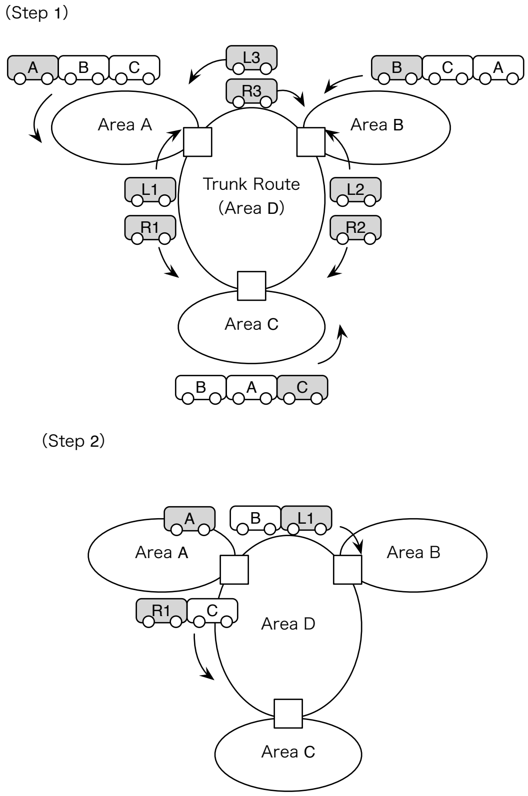

Fig. 5. Example of traffic control: Steps 1 and 2.

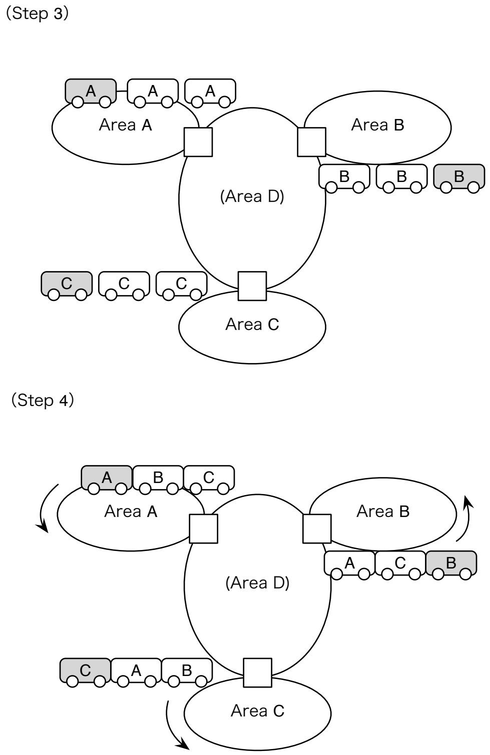

Fig. 6. Example of traffic control: Steps 3 and 4.

communication, minimizing passenger waiting time, determining a targeted optimum operation schedule, and instructing each lead vehicle.

Moreover, in addition to the central server, a server at each node (hereinafter referred to as a *node server*) is maintained. The configuration and the operation route of a fleet as determined by the central server are transmitted once to each vehicle via the node server within the communication network.

G. Example

We now present a simple example of vehicle management and demonstrate that a passenger can be transported to any destination without a changeover operation (see also Figs. 5 and 6). In this example, we consider a network consisting of four areas, labeled A, B, C, and D, consisting of a trunk route connecting them. Areas A, B, and C are geographically remote, and the network provides a *rapid service link* via the trunk route where the lead vehicles travel at higher speeds. To simplify our discussion, we assume that for each of the four areas, lead vehicles do not go outside their own areas. In the figure, squares marked “A”, “B”, and “C” represent vehicles displaying the destination boards as “A”, “B”, and “C”, respectively, with lead vehicles in gray and trailers in white.

In Areas A, B, and C, a route circling a plurality of stops is established, and lead vehicles for the area circulate repeatedly along it. For the sake of simplicity, we consider each area to have one lead vehicle and the individual stops are omitted in the figure. Also, the trunk route consists of a closed route along which lead vehicles circulate, either clockwise or counterclockwise. Here, we assume there are six lead vehicles with those moving counterclockwise labeled “R1”–“R3” and those moving clockwise labeled “L1”–“L3”. This route connects each of the areas A, B, and C.

In this network, passengers are transported in the following way. (See also Figs. 5 and 6.) In Step 1, in each of Areas A, B, and C, the destination board of the lead vehicle displays the area to which the vehicle belongs, while the destination boards of the trailers display an outside area. For example, in Area A, the destination board of the lead vehicle displays “Area A” while the trailers’ display “Area B” and “Area C”, respectively. During a route around each area, passengers board the vehicle displaying the destination board in an area of their destination.

Next, in Step 2, as each vehicle goes around in its own area, trailers detach at the node. For legibility in the figure, we have focused attention on the behavior of vehicles at the node of Area A. Each detached trailer is then connected to a lead vehicle in Area D going to the same destination.

In Step 3, trailers arriving at the node of their destination area are detached and then connect to the lead vehicle of that area. (For readability, the vehicles in “Area D” are omitted in the figure.)

Furthermore, in Step 4, the trailers update the destination board to display an outside area and continue around the route of the current area picking up and dropping off passengers. For example, in Step 3, the trailers, which have been taken from Areas B and C via Area D, have updated their destination boards to Areas B and C in Step 4.

We note here that the trailer whose destination board has been updated to Area B or C may be carrying passengers whose destinations are stops in Area A. Therefore, as the trailer goes around Area A picking up new passengers in this area, it is dropping off passengers already on board, and thus the vehicle is being used efficiently. Moreover, as illustrated in the above example, passengers having boarded a vehicle displaying the destination of their choice arrive at their destination without performing a changeover operation.

III. FAULT TOLERANCE OF THE TRANSPORTATION SERVICE

A. Central Idea

We now present a method of fault tolerance in the management of this transportation service against central server failure. To ensure the service availability of the transportation, our central idea is to switch the current schedule (i.e., routing of lead vehicles) determined by the central server to a predetermined provisional schedule when a node server detects a failure of the central server.

To realize this mechanism, a simple way is to determine in advance the assignment of each vehicle to a provisional route. However, if the traffic network is relatively wide spread, it takes time to reach the assigned travel route if the distance between the current position during the disruption and the assigned travel route is large.

To avoid this problem, another possible idea is to dynamically allocate a route to each lead vehicle arriving at its own node. However, when the central server fails, the network between the node servers fails simultaneously, and the node servers cannot reach a consensus about current assignments.

As a method to assign routes while solving these problems, we propose the idea that each node has a set of provisional routes as well as a predetermined number of vehicles to be assigned to each route.

B. Assumptions regarding Failure

Here we assume only a failure of the central server, and specifically a scenario in which information from the central server does not reach any node server. That is, we are not considering partial failures for which the central server can communicate with some but not all node servers. Also, if the central server fails, all lead vehicles are able to detect it immediately. Furthermore, we assume that the disruption may last a long time but that the communication with the central server will be restored eventually.

C. Proposed Method

First, in our method, it is necessary to determine the following items (1)–(4) in advance.

1) Determine a set of provisional routes to be assigned to lead vehicles. At this time, the routes satisfy conditions (A) and (B) below.
   - (A) In each area, at least one provisional route travels around the area.
   - (B) By following one or more provisional routes, the vehicle can establish a route starting from one of the two arbitrary nodes to the other.
2) For each node, determine a rule concerning the updating of trailers when a new fleet arrives at the node. That is, a rule is set so that the trailers to be connected to a lead vehicle are selected among the active trailers on routes and parked vehicles in the vehicle pool. As an example, we may consider a rule such that the trailers are selected to attend to passengers who have the longest waiting time.
3) For each node, set rules on how to change the destination board when a new fleet arrives at a node. Here, as described above, when a lead vehicle arrives at a node, the destination board is changed if the area currently displayed and the area of the node coincide. The rule concerning a change in display of the destination board must satisfy the following condition.
   - (C) At each stop, vehicles of every destination visit regularly.
4) Set to which route the number of nodes is allocated to each node server. At this point, the sum of the number of vehicles assigned by each node server needs to match the number of lead vehicles in the whole system.

We here focus some attention on Conditions (A), (B), and (C) presented above. These three conditions are sufficient to ensure that if a failure occurs traffic control can continue without any passengers willing to transfer to another vehicle in the transportation system. The reason is that by satisfying Conditions (A) and (B), if there is a sufficient number of trailers for passengers, by appropriately switching the connection of the vehicles (more specifically, by being connected and taken by the lead vehicle heading towards a node nearer to the destination node), the trailers can move between any two arbitrary stops. Furthermore, by satisfying Condition (C), it is necessary for the trailers to halt at any stop in its destination area.

According to the setting of 1 to 4 explained above, the node server assigns a route every time an unassigned lead vehicle arrives until the assignment of the routes in its charge is completed. As a result, the assignment as intended can be performed without information being exchanged between nodes.

D. Procedure

If a schedule determined by the central server does not arrive at the node server due to a certain failure, each node server detects the failure and then shift to its operation mode described below. While in this operation mode, each time a fleet arrives at a node, the server instructs the lead vehicle, in

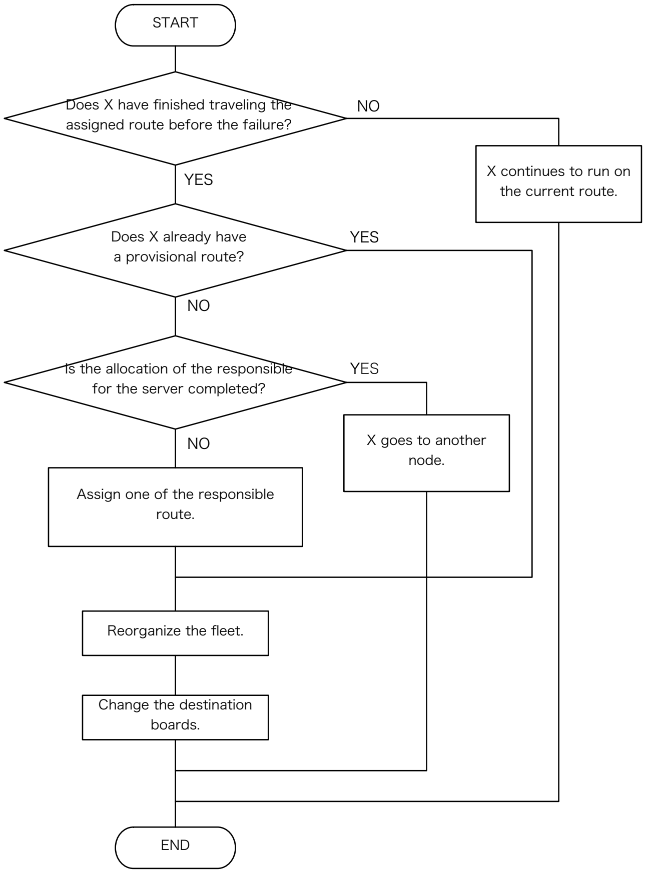

Fig. 7. Flowchart of the operation mode following a failure.

accordance with the procedures of Steps 1 to 3 below, which switches the fleet to a provisional route. (See also Fig. 7 for the flowchart of this procedure.)

Step 1. When the lead vehicle (called X) arrives at the node, the next operation is performed.

- Case 1: When a provisional route has not yet been assigned to X,
- Case 1-1: When the node server has not allocated the responsible section,
- Case 1-1-1: When X has finished traveling the indicated route before the failure occurred,
- Case 1-1-1-1: When there remains provisional routes at the node, if there is anything in it that includes or is near the current location, one of them is assigned the provisional route of X. (Go to Step 2)
- Case 1-1-1-2: Otherwise, instruct X to head to another node. At this point, it is assumed that the node destinations have been determined in advance taking into consideration the efficiency associated with the allocation. (Go back and wait for the arrival of the next lead vehicle.)
- Case 1-1-2: When X has not completed the indicated route before the failure,
- Case 1-1-2-1: When there remain provisional routes at the node, if there are routes that includes or is close to the final destination instructed prior to failure, one of them is assigned as the provisional route of X. (Go to Step 2)
- Case 1-1-2-2: Otherwise, do the same processing as Case 1-1-1-2.
- Case 1-2When the node server has already assigned the route, do the same processing as in Case 1-1-1-2.
- Case 2: When a route has already been assigned to X, continue running on the current assigned route. (Go to Step 2)

Step 2. Select vehicles in the order of longest waiting times among passengers from the trailers currently connected to the lead vehicle and the trailer which is stopping at the node. Then reorganize the fleet consisting of the selected vehicles. (Go to Step 3)

Step 3. If all passengers alight in the area, change the destination board by the predetermined method. As mentioned earlier, the rule for changes at this time is set in advance so as to satisfy Condition (C) stated in the previous subsection.

When the central server is restored back into operation, each node server orders the lead vehicles to follow the instruction issued by the central server.

IV. IMPLEMENTATION

We often face a common dilemma when we design an algorithm and try to implement it in the real world, especially in an application area that requires a high level of reliability. In the designing phase, we have tried to keep the algorithm as simple as possible to make it easy to be effectively validated. However, in the implementation phase, we have needed to add extra details to overcome realistic obstacles such as incompatibility of software stacks, poor performance, and security threats. These prevent us from performing frequent validations on a modified algorithm and therefore delays incremental improvements in terms of performance and/or adaptations to emerging situations.

As a realistic compromise regarding this dilemma, we introduce an idea which we call the *deployable simulator*. (See also Fig. 8 for a screenshot of our simulator.) The concept underpinning this simulator is that simulation codes, which are supposed to be as simple as possible, should also be able to run on a real system in the wild. In other words, we would like to expand the range of application of DevOps even to a system requiring high reliability.

Our solution to realize a deployable simulator is to model the entire system as a collection of actors. The actor model [11] is a fundamental model for concurrent computation. An actor is a computational entity that has a message queue and an ability to send messages asynchronously to other actors. Incoming messages are received through the message queue

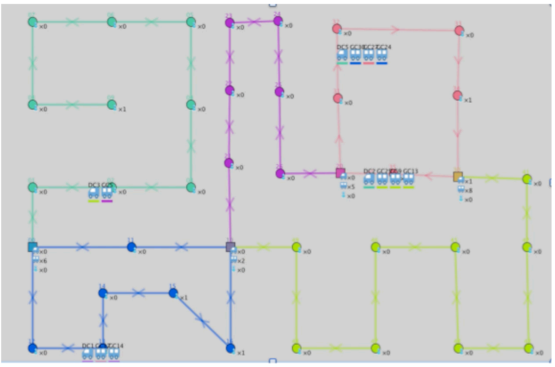

Fig. 8. Screenshot of the deployable simulator

one by one and computation will proceed as a series of responsive reactions to the messages. Simulation codes written in the actors model can run not only on a single computer but also on a distributed computing environment through the function called Remote Actors [3]. Combining this feature and software container technology such as Docker [4], we run core algorithms as on a real system.

V. RELATED WORK

As mentioned in the introduction, improving convenience in last-mile transportation is one of the important tasks in modern society, and there are many studies on such transportation modes such as bus demand.

With the rapid advent of self-driving technology, there have been a number of studies on fleets of autonomous vehicles [10], [12], [5]. However, these studies focused on how to organize vehicles into a single fleet and how to longitudinally control each vehicle using VANET (Vehicular Ad-hoc Network), whereas we are focusing on how to organize and operate multiple fleets. Fernandes et al. [10] proposed intra-fleet communication strategies that exploit anticipatory information for improving stability. Jia et al. [12] proposed a driving strategy that mitigates the impact of traffic disturbance. Amoozadeh et al. [5] developed a novel protocol that uses both CACC (Cooperative Adaptive Cruise Control) and VANET.

Various methods related to the optimization of operation schedules aimed at minimizing waiting times of passengers, for example, have been proposed within fields of application of mathematical optimization. These are based on the Vehicle Routing Problem or its variant such as the Dial-a-Ride Problem, School Bus Routing Problem (for an overview of these problem, see [14], [8], and [13], respectively). Among them are schedule optimization of transportation systems similar to that of our study dealing with delivery problems for dynamic planning of multiple vehicles [6] and scheduling with transfers between passenger vehicles [9], [7]. However, in these research fields, a last-mile transportation system that runs fleets of vehicles has been rarely dealt with so far.

VI. CONCLUSION

We have described a last-mile public transportation system with autonomous vehicles. In particular, we focused attention on methods of vehicle traffic management. We proposed a method of efficiently transporting passengers by dividing the stop into several areas and appropriately updating the destination board provided for each vehicle at the node in the system. In addition, in the event of a failure of the central server, we proposed a method for the continuation of transportation services for passengers by automatically shifting each vehicle to a provisional operation route. Moreover, we described an implementation of our vehicle operation management using techniques adopted from parallel computing and virtualization.

One of the important tasks for the future is the development of an optimization method for scheduling. In particular, minimizing the waiting time of passengers (the difference between the shortest travel time and the actual travel time taken) and minimizing the cost of moving vehicles are considered important in practical applications. In the study, we assumed a server to perform the centralized operations management and a server at each node in the transport network, albeit from the viewpoint of fault tolerance and autonomous distributed types that only determine operation scheduling between vehicles, and excluded the installation of such servers. Considerations regarding control methods would be beneficial in advancing these developments.

REFERENCES

[1] http://www.2getthere.eu/projects/rivium-grt/

[2] http://www.citymobil2.eu/

[3] http://doc.akka.io/docs/akka/snapshot/scala/remoting.html

[4] https://www.docker.com/

[5] M. Amoozadeh, H. Deng, C.-N. Chuah, H. M. Zhang, and D. Ghosal. Platoon management with cooperative adaptive cruise control enabled by VANET. *Vehicular Communications*, vol.2(2), pp.110-123, 2015.

[6] A. Attanasio, J-F. Cordeau, G. Ghiani, and G. Laporte. Parallel Tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, vol.30, issue 3, pp.377-387, 2004.

[7] M. Bögl, K. F. Doerner, S. N. Parragh. The School Bus Routing and Scheduling Problem with Transfers. *Networks*, vol.65, no.2, pp.180-203, 2015.

[8] J-F. Cordeau and G. Laporte. The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, vol.153, pp.29-46, 2004.

[9] C. E. Cortés, M. Matamala, and C. Contardo. The pickup and delivery problem with transfers: Formulation and a branch-and-cut solution method. *European Journal of Operational Research*, vol.200, pp.711-724, 2010.

[10] P. Fernandes and U. Nunes. Platooning With IVC-Enabled Autonomous Vehicles: Strategies to Mitigate Communication Delays, Improve Safety and Traffic Flow. *IEEE Transactions on Intelligent Transportation Systems*, vol.13(1), pp.91-106, 2012.

[11] C. Hewitt, P. Bishop, and R. Steiger. A universal modular ACTOR formalism for artificial intelligence. *3rd international joint conference on Artificial intelligence*, pp.235-245, 1973.

[12] D. Jia, K. Lu, and J. Wang. A Disturbance-Adaptive Design for VANET-Enabled Vehicle Platoon. *IEEE Transactions on Vehicular Technology*, vol.63(2), pp.527-539, 2014.

[13] J. Park and B. Kim. The school bus routing problem: A review *European Journal of Operational Research*, vol.202, pp.311-319, 2010.

[14] P. Toth and D. Vigo (eds.) *Vehicle Routing: Problems, Methods, and Applications (2nd ed.)*, Society for Industrial and Applied Mathematics, 2014.