Published in IET Intelligent Transport Systems
Received on 30th November 2009
Revised on 23rd May 2010
doi: 10.1049/iet-its.2009.0113

Special Issue – selected papers from the 16th
World Congress on ITS

ISSN 1751-956X

Innovative on-demand bus system in Japan

K. Tsubouchi[1] *H. Yamato*[1] *K. Hiekata*[2]
[1]Graduate School of Frontier Sciences, The University of Tokyo, 5-1-5, Kashiwa-no-ha, Kashiwa, Chiba, Japan
[2]Graduate School of Engineering, The University of Tokyo, 5-1-5, Kashiwa-no-ha, Kashiwa, Chiba, Japan
E-mail: tsubouchi@is.k.u-tokyo.ac.jp

Abstract: The innovative on-demand bus system is developed and the result of field tests shows that the system is valid for different city types. On-demand is a demand-responsive transit service where the vehicles transport users after they reserve their seats, and the vehicle does not move if there is no reservation. On-demand bus is an existing technology and it runs all over the world, but the high running cost is a problem. The aim of the developed system is practical introduction of the system from cost viewpoint. Local government can start the service in low cost with cloud computing technology. In order to build up the cloud computing system, schedule calculation system and the communication device in the car are required. In order to validate the developed system, simulation and field test are held. The result of computer simulation shows that the developed calculation algorithm works well as designed. The result of field test shows that the on-demand bus service in three different type cities are provided through the Internet with cloud computing technology and it is evaluated to enable new public transportation system.

1 Introduction

This paper presents the details of the innovative on-demand bus system and the evaluation of the developed service by field tests.

On-demand bus is a demand responsive transit service where the vehicles transport users after they reserve their seats, and the vehicle do not move if there is no reservation. On-demand bus is an existing technology and it runs all over the world, but the high running cost is a problem. The high running cost comes from two reasons, the computer system and the cost for hiring operators. Specifically, the local government has to purchase a set of large server system in order to start and keep the service, and they have to employ operators who have the following three tasks: to input the demand through the computer in behalf of the customers, to calculate and make the route, and to relay the new route to the bus.

The 'innovative' on-demand bus can resolve these problems by cloud computing technology. The on-demand bus service is provided through the Internet, so local government officers do not have to purchase and maintain the large-scale server. Every vehicle in the world can be part of the on-demand bus system by putting the communication device in the vehicle. Actually, the running cost of the introduction of on-demand bus service can be lowered to one-tenth compared with the existing on-demand bus system.

The 'innovative' on-demand bus system is not only able to run with low cost but is also able to give high-quality on-demand bus scheduling service. The developed scheduling algorithm can automatically update the schedule during reservation. So the local government does not have to employ the operators because the customers who can operate mobile phone or personal computers reserve by themselves.

To introduce the idea of cloud computing is easy but to make the system practical is difficult because the calculation speed of the scheduling algorithm is the problem. Solving problem is categorised as on-line capacitated dial-a-ride problem (DARP) with time window and this problem is proven as NP-hard problem. So, high calculation specification is required when the computer solves the problem and a single server cannot correspond multi-reservation from many accesses from plural cities. Fig. 1 shows the outline of this paper.

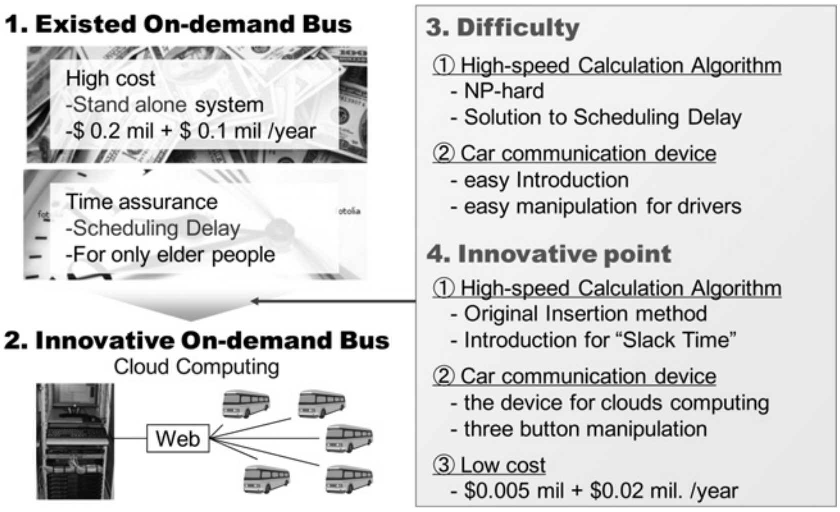

Figure 1 *Outline of the research of innovative on-demand bus project*

The rest of the paper is constructed as follows: Section 2 outlines the method used here. Section 3 sets out the results obtained, then conclusions in Section 4.

2 Developed on-demand bus system

2.1 Overview of the system

The main part of the developed on-demand bus system consists of four technologies: response gateway, schedule calculation system, communication device and database.

Fig. 2 shows the overview of the on-demand bus system. Passengers access the response gateway for bus reservation via phone or web. The response gateway relays the passengers' demand information to the schedule calculation system where the original scheduling algorithm is implemented. The algorithm updates the route after which the system announces the new route to the communication device in the vehicle. Finally, the communication device stores up actual moving time into the database.

The innovative points mentioned here are the high-quality scheduling method by the schedule calculation system and easy introduction in a city with the help of the communication device. These systems are provided with the cloud computing technology.

2.2 Cloud computing technology

Cloud computing is an Internet-based computing, whereby shared resources, software and information are provided to computers and other devices on demand, like the developed on-demand bus service.

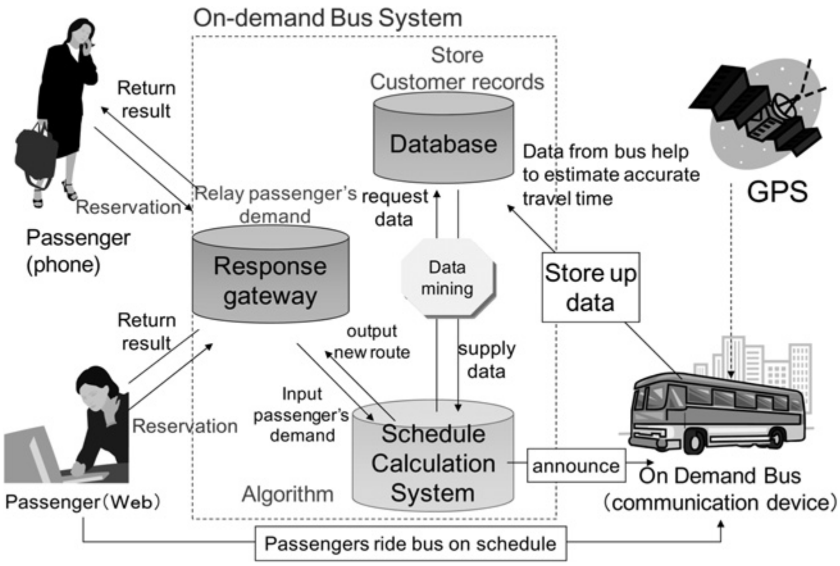

Figure 2 *On-demand bus system overview*

The merits of cloud computing technology are low cost and information security. First one is low cost introduction and low cost maintenance of computer system by sharing the server. In the existing on-demand bus system, the server is introduced in each local government and the system manager is hired to check the hardware and software. So, the budgets for purchase and maintenance of the server system and hiring the system manager or updating software should be prepared. However, these costs are disappeared with cloud computing technology because anyone can start the on-demand bus service just by writing application of software used. Local government can introduce the on-demand bus service through the Internet just by purchasing software license. There is an example that the local government can decrease the system cost of the on-demand bus service by one-twentieth in Japan.

The second merit of cloud computing technology is the information security. The personal information is stored in the server maintained by the local government officers, who have little knowledge of the computer maintenance when the local government purchases stand-alone server system. On the other hand, the specialist can check the centre server by cloud computing technology. It means that the cost for the professional person is also shared with some local government, and the information security is improved through professional maintenance. Thanks to the improvement of the computer specification, one computer can perform highly so that the other computer can supply the on-demand bus service to plural local area.

2.3 Schedule calculation system with new insertion algorithm

When new customers reserve for the bus, customers input their demand to the schedule calculation system through the response gateway. The information of 'demand' consists of five elements: origin, destination, expected time of arrival (or expected pick-up time), date and the number of seats for reservation. The system calculates the best route that satisfies the new demand without violating a set of constraints that reflect the previously planned transportation demands. The calculation time should be shortened because the customer waits for completion of the calculating process over response gateway for finalisation of reservation.

The routing algorithm decides the speed and quality of calculation. In operational research field, this problem is classified as the DARP, especially on-line capacitated DARP with time windows. 'On-line' means routes change with every reservation; 'capacitated' means there is a volume constraint in the car, and 'with time windows' shows problem has some constraints of time windows. On-line C-DARP is proven to have NP-hard constraint [1], so calculation time increases exponentially by increasing the number of customers and the number of vehicles. In this case, the cloud computing system cannot be employed because of out of spec of the calculation server since the scale of the problem is large.

Stein [2] and Psaraftis [3, 4] examined DARP when no defined time windows exist during which the service is to begin, and also when the service is done with one vehicle only. Cullen *et al.* [5] dealt with DARP also in case of absence of time windows, but when the service is performed by a greater number of vehicles. Sexton [6] studied the static version of DARP with defined time windows and one vehicle service. Yet, the most common problems encountered in practice are static dial-a-ride systems with specified time windows and performed by a greater number of vehicles. These systems were, among the others, researched by Jaw *et al.* [7], Alfa [8], Rhee [9], Kikuchi and Rhee [10] and Vukadinovic [11].

Considering the requirements, the routing algorithm should be a heuristic algorithm to insert the new customer's pick-up and get-off events into the planned route within a second. Li and Lim [12] also solved on-line DARP by Insertion method but they took a lot of calculation time. So, it becomes infeasible when the number of customers or number of buses increase. Fabri and Recht [13] shortened the calculation time by assuming the first found feasible solution semi-optimal. In the developed solution, Fabri's idea is modified to be able to select the better vehicle to insert by comparing each bus-moving locus.

The routing algorithm used in the existing on-demand bus service also inserts the new reservation, but delay often occurs with every insertion because the new customer's reservation lengthens the travel time of the bus. The developed algorithm [14] can insert new customers' reservation without any delays in any of the informed arrival times. It is necessary to permit some latitude in the schedule in order to be able to insert the new customer's reservation. The suggested system has a time window called 'slack time' as elbowroom of the schedule. The slack time is a unique time window about the announced arrival time and it assures that the bus arrives at the bus stop in the time window. For example, in case when the announced arrival time is 9:00 a.m. and slack time is 15 min, the bus arrives at the bus stop between 8:45 and 9:00 a.m. With the introduction of the new insertion algorithm and slack time, the bus does not miss the announced arrival time and it can transport several customers to their various destinations by shared rides.

2.4 Problem formulation

2.4.1 Mathematical notation

N: total number of customers requesting service

n: indicates a customer

DPT_n (DDT_n): desired pick-up time of customer n

EPT_n (EDT_n): earliest pick-up time of customer n

doi: 10.1049/iet-its.2009.0113

LPT_n (LDT_n): latest pick-up time for customer n

APT_n (ADT_n): actual (scheduled) pick-up time for customer n

$D(x, y)$: direct travel time of the shortest route from point x to point y

$+n$ ($-n$): the event 'pick-up (delivery) customer n'; '$+n$' ('$-n$') also denotes the point of origin (destination) of customer n

TT(x, y, hour): travel time from point x to point y at a specific time

$p(x)$: pick-up or drop-off point, i.e. $p(+n)$ or $p(-n)$

Bus_n: bus number which will serve customer n

DRT_n: direct ride time of customer n, that is $\mathrm{DRT}_n = D(+n, -n)$

MRT_n: maximum acceptable ride time for customer n

ST_n: slack time of customer n

IPT_n (IDT_n): informed pick-up (delivery) time of customer n

V: total number of vehicles

CAP_v: maximum number of passengers that vehicle v can carry.

2.4.2 Problem: In this problem, N customers have to be transported by a maximum of V vehicles. Each customer, customer n, has to specify pick-up bus stop, $p(+n)$, and delivery bus stop, $p(-n)$. The customer also has to specify either the desired pick-up time (DPT_n) or a desired delivery time (DDT_n). Most individuals are constrained in the morning by a desired 'delivery' time (e.g. work start time) and select their start of trip time accordingly. Such dial-a-ride customer will be a 'DDT-specified' customer and will rely on the system to tell him at what time he will be picked up so that he will be delivered by DDT. The reverse is true for DPT-specified customers. After system calculation, the pick-up time (IPT_n) and informed delivery time (IDT_n) will be relayed to the customer.

2.4.3 Initial time window setting: Given a subscription list of N customers, each specifying either a DPT_i or DDT_i ($i = 1, 2, \ldots, N$) and a fleet of V vehicles, find an effective allocation of customers among vehicles and associated time schedule of pick-ups and deliveries such that:

1. for all customer n

$$\mathrm{EPT}_n \le \mathrm{APT}_n (= \mathrm{IPT}_n) \le \mathrm{LPT}_n \tag{1}$$

$$\mathrm{EDT}_n \le \mathrm{ADT}_n \le \mathrm{IDT}_n \le \mathrm{LDT}_n \tag{2}$$

2. For DPT-specified customers

$$\mathrm{EPT}_n = \mathrm{DPT}_n \tag{3}$$

$$\mathrm{LPT}_n = \mathrm{EPT}_n + \mathrm{ST}_n \tag{4}$$

$$\mathrm{DRT}_n = \mathrm{TT}(p(+n), p(-n), \mathrm{hour}(\mathrm{DPT}_n)) \tag{5}$$

$$\mathrm{EDT}_n = \mathrm{EPT}_n + \mathrm{DRT}_n \tag{6}$$

$$\mathrm{LDT}_n = \mathrm{LPT}_n + \mathrm{DRT}_n \tag{7}$$

3. For DDT-specified customers

$$\mathrm{LDT}_n = \mathrm{DDT}_n \tag{8}$$

$$\mathrm{EDT}_n = \mathrm{LDT}_n - \mathrm{ST}_n \tag{9}$$

$$\mathrm{DRT}_n = \mathrm{TT}(p(+p), p(-n), \mathrm{hour}(\mathrm{DDT}_n)) \tag{10}$$

$$\mathrm{EPT}_n = \mathrm{EDT}_n - \mathrm{DRT}_n \tag{11}$$

$$\mathrm{LPT}_n = \mathrm{EPT}_n + \mathrm{ST}_n \tag{12}$$

Time window is set as shown in Fig. 3.

2.4.4 Time window setting after reservation completion: When one reservation is completed, the time window is set by the constraints below

$$\mathrm{IPT}_n = \mathrm{APT}_n = \mathrm{EPT}_n \tag{13}$$

$$\mathrm{LDT}_n = \mathrm{IDT}_n \tag{14}$$

$$\mathrm{EDT}_n \le \mathrm{ADT}_n \le \mathrm{LDT}_n (= \mathrm{IDT}_n) \tag{15}$$

$$\mathrm{LPT}_n = \mathrm{LDT}_n - \mathrm{DRT}_n \tag{16}$$

In order to guarantee that the actual riding time does not exceed the maximum acceptable riding time, the following constraint is also introduced

$$0 \le \mathrm{ADT}_n - \mathrm{APT}_n \le \mathrm{MRT}_n \tag{17}$$

New time window is set as shown in Fig. 4. Since IPT_n and IDT_n are introduced by calculation and confirmed by customers, time windows are narrowed by calculation because the bus cannot leave the bus stop before IPT_n and may not delay at IDT_n. The time windows solidly painted are new time windows setting after reservation completion.

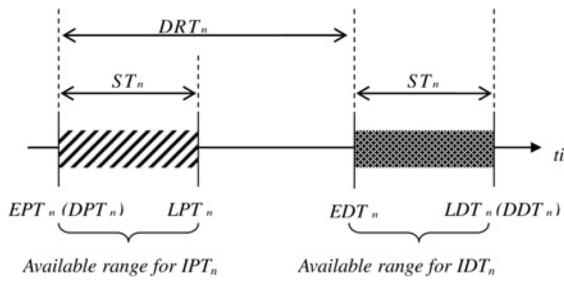

Figure 3 *Initial time windows before defining IPT_n*

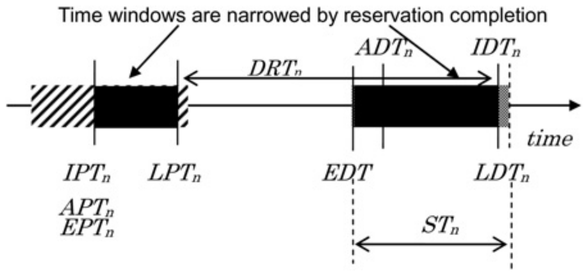

Figure 4 *Time windows after defining IPT_n*

2.5 Algorithm outline

In the developed on-demand bus system, two algorithms are combined together as shown in the following figure. Using the first algorithm, vehicle choosing algorithm, the decision is made about which vehicle accepts the new request. The second algorithm, routing algorithm, is used to design the new route and schedule for the vehicle chosen to serve the new request (Fig. 5).

2.5.1 Vehicle choosing algorithm: In the vehicle choosing algorithm, an effective algorithm with less calculation time is introduced, especially when solving big problems. The direction variable as a decision criterion is proposed.

First, direction vector ($\boldsymbol{A}_n$) of customer n is declared as

$$\boldsymbol{A}_n = \overrightarrow{p(-n)} - \overrightarrow{p(+n)} \tag{18}$$

At the same time, bus direction vector ($\boldsymbol{B}_i$) is defined as

$$\boldsymbol{B}_i = \overrightarrow{p(s_i^*)} - \overrightarrow{p(t_i^*)} \tag{19}$$

In this case, $p(s_i)$ and $p(t_i)$ are considered to be nearest to the LPT_n and LDT_n, respectively.

The direction decision variable (θ_i) is then defined as

$$\cos\theta_i = \frac{\boldsymbol{A}_n \cdot \boldsymbol{B}_i}{|\boldsymbol{A}_n||\boldsymbol{B}_i|} \tag{20}$$

When a new reservation comes into the system, the vehicle choosing algorithm is executed for each available bus. Since the bus with the most value of cos θ is the one with the closest direction to the new demand, that bus is selected first in the execution of the next algorithm that is the routing algorithm.

2.5.2 Routing algorithm: *Routing algorithm outline:* For the routing algorithm, a heuristic algorithm was developed which can be described as follows. In this example, there are $n-1$ passengers who have already reserved the bus. Then comes a new reservation from customer n. The proposed 'insertion and time adjustment algorithm' inserts event $p(+n)$ and $p(-n)$ into the planned route. After insertion, some passengers' APT_i or ADT_i ($i = 1, 2, 3, \ldots n-1$) are changed within their respective time windows.

Declaration of variables: In this 'insertion and time adjustment algorithm', four new variables are defined. TimeLimit and Repeat define conditions that the calculation is finished.

S(e): feasibility of event e; explained in number 4

TimeLimit: limitation of search time

Repeat: maximum number for iterations

C(e): capacity check in event e; explained further in Section 6.

Insertion and time adjustment algorithm: The procedures of the algorithm are shown by the following flowchart. There are completed route for $n-1$ customers. A new customer n requests for a bus. The time windows of all events $\{P(+1), P(-1), P(+2), P(-2), \ldots, P(+n-1), P(-n-1)\}$ are calculated. With this algorithm, the APT_n (IPT_n) and ADT_n (IDT_n) are decided.

First, the algorithm sets APT_n and try insertion. After insertion, the feasibility check is performed. If all events are feasible, the route is decided. If there are some infeasible events, the TimeLimit and Repeat are checked. If the two variables do not exceed the limitation, the iteration is executed by adjusting the departure time of infeasible

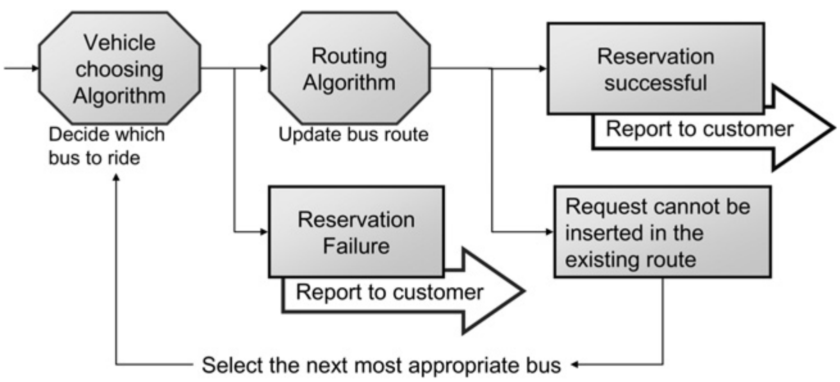

Figure 5 *Algorithm outline*

doi: 10.1049/iet-its.2009.0113

events. However, the processes stops if one or both of the variables exceed the defined limitation (Fig. 6).

Lastly, the number of passengers is checked in all events. If the number of passengers exceeds the capacity of vehicle, the routing process is reported as failed and the algorithm searches the next bus or changes the desired arrival time of all the passengers.

Feasibility of event or S(e): $S(e)$ was set up to watch for feasibility of events. There are possible values of $S(e)$ with different meanings as shown in Table 1.

Time adjustment for infeasible events: When there is an infeasible event but the iteration does not violate the TimeLimit and Repeat constraints, time adjustments are performed. Table 2 shows the process of time adjustment by each $S(e)$ value.

Capacity check for event or C(e): $C(e)$ is set up to check the volume of each vehicle in event e. $C(e)$ is defined as follows

$$\begin{cases} C(e) = 0 & \text{in} \quad \mathrm{CAP}(e)_{\mathrm{Bus}_n} \le \mathrm{CAP}_{\mathrm{Bus}_n} \\ C(e) = -1 & \text{in} \quad \mathrm{CAP}(e)_{\mathrm{Bus}_n} > \mathrm{CAP}_{\mathrm{Bus}_n} \end{cases} \quad (21)$$

where $\mathrm{CAP}(e)_v$ indicates the number of passengers in vehicle v at event e. So, $C(e)$ is 0 when the number of passengers in event e does not exceed the capacity of vehicle v.

Conditions that signal the end of calculation: There are two conditions that signal the end or finished calculation. First,

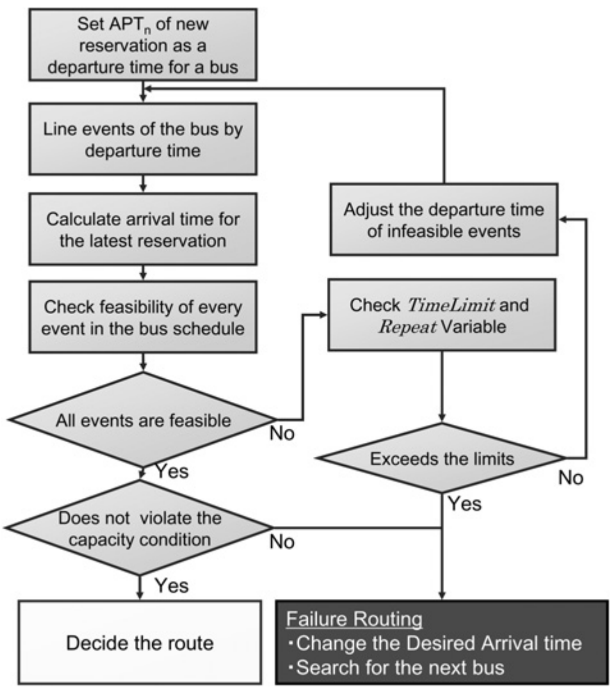

Figure 6 *Insertion and time adjustment algorithm*

Table 1 $S(e)$ explanation

$S(e)$	Explanation
0	event e is feasible
−1	event e is infeasible because of faulty events' consequence ($\mathrm{APT}_e > \mathrm{ADT}_e$)
−2	event e is infeasible because of late delivery time ($\mathrm{ADT}_e > \mathrm{LDT}_e$)
−3	event e is infeasible because of early delivery time ($\mathrm{ADT}_e < \mathrm{EDT}_e$)
−4	event e is infeasible because of late pick-up time ($\mathrm{APT}_e > \mathrm{LPT}_e$)
−5	event e is infeasible because of early pick-up time ($\mathrm{APT}_e < \mathrm{EPT}_e$)

Table 2 Time adjustment by $S(e)$ value

$S(e)$	Adjustment
0	no adjustment
−1	$\mathrm{APT}_e^* = \mathrm{LPT}_e$
−2	$\mathrm{ADT}_{e-1}^* = \mathrm{ADT}_{e-1} - (\mathrm{ADT}_e - \mathrm{LDT}_e)$
−3	$\mathrm{ADT}_{e-1}^* = \mathrm{ADT}_{e-1} + (\mathrm{EDT}_e - \mathrm{ADT}_e)$
−4	$\mathrm{APT}_e^* = \mathrm{LPT}_e$
−5	$\mathrm{APT}_e^* = EPT_e$

a feasible route should be found.

$$\sum_e^N S(e) = 0 \quad \cap \quad \sum_e^N C(e) = 0 \quad (22)$$

Another one is when the calculation time exceeds TimeLimit or the calculation iteration exceeds Repeat.

2.5.3 Cancellation: The schedule calculation system updates the bus travelling plan not only when the new customers reserve their sheet but also when the reserved customers cancel their reservation. When the reserved customers cancel their sheet, the system just deletes their reservation data from the decided plan.

2.6 Communication device

Communication device shown in Fig. 7 is used to exchange information between the bus and the server. This device tells the driver the information of when and where the bus should go next, as well as the information of who gets on/off the bus. Commercial wireless network is used to exchange the data between server and the bus. So, the communication device can get the new plan through the internet when the travelling plan is changed by new

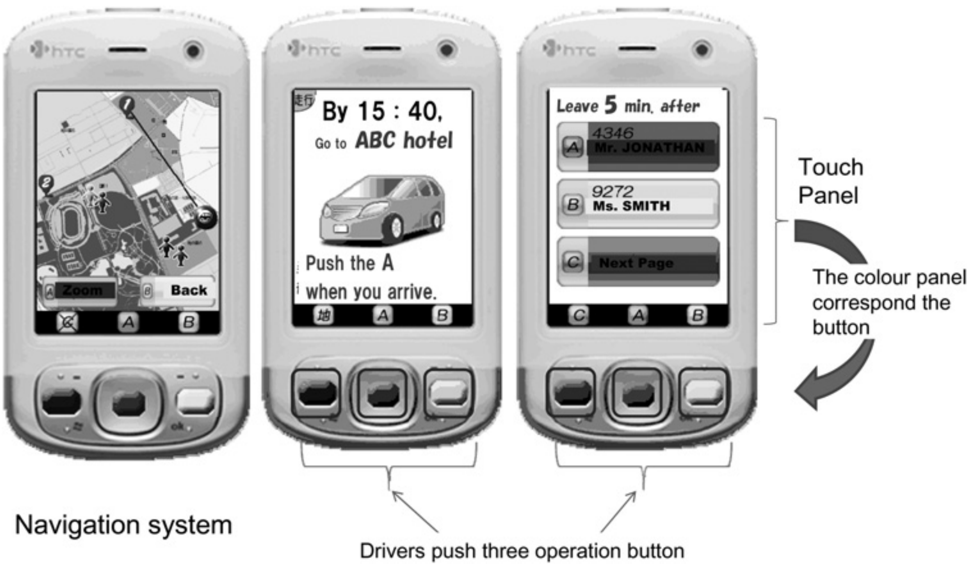

Figure 7 *Communication device (navigation page, schedule order, passenger list)*

reservation or new cancellation. Software is implemented on personal digital assistant and drivers are able to operate it by touching the display buttons. Customers get the accurate position of bus by global positioning system. The drivers know the bus route from the two windows: main window and navigation window. The passengers' information is displayed on the main window, whereas the navigation window tells drivers the path to the next destination.

3 Test result

3.1 Computation test result

In order to validate the scheduling algorithm, a computer simulation is executed. Simulator creates on-demand vehicle and passengers on the computer. In the virtual world, passengers reserve their seats and vehicles pick up and deliver passengers as scheduled. In the simulation, the scheduling algorithm is validated by calculation time and the violation of time window constraints. In the computation test, TimeLimit is not defined because measuring the calculation time is one of the objectives of this test.

By computational simulation, it is proven that the developed scheduling algorithm works well, and that the on-demand bus with developed algorithm performs efficiently. Calculation time does not increase exponentially but is almost linear when the problem scale becomes large.

Fig. 8 shows the result of calculation time. *X*-axis indicates the reservation case composed of the number of on-demand vehicles (*V*)## and the calculation result (OK/NG). *Y*-axis shows the distribution of the calculation time in each case. Numbers in the graph indicate average calculation time in computer simulation of calculation cases denoted by dots in

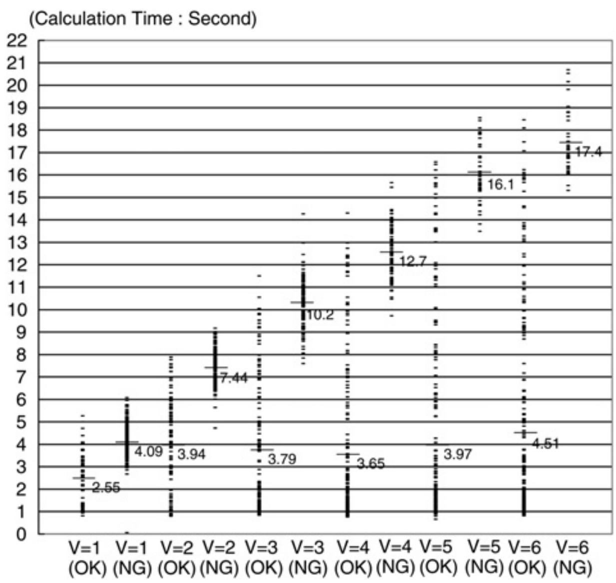

Figure 8 *Calculation time (s) results*

the figure. Calculation time increases when there are more reservations added passing through the 'insertion and time adjustment algorithm' until such time that the calculation result returns NG. The figure also shows that a longer calculation time is needed when the number of vehicles increases. However the increasing rate is almost linear, although the calculation time for on-line DARP usually increased exponentially in previous research. It means that this scheduling algorithm is practical because passengers can get the results immediately and at minimal waiting time. Moreover, the average calculation time does not increase when the system can find the route and it increases when the system cannot find any feasible route. It means that the system can update the route by setting the TimeLimit variable even if the number of vehicle increases.

IET Intell. Transp. Syst., 2010, Vol. 4, Iss. 4, pp. 270–279

doi: 10.1049/iet-its.2009.0113

It is also confirmed that the calculation does not violate the time window constraints.

APT_n should not be earlier than IPT_n because passengers arrive at the bus stops after the bus leaves from the bus stops. On the other hand, ADT_n should not be later than IDT_n because this service ensures the arrival time.

Fig. 9 shows the distribution of APT_n−IPT_n and this indicates that the positive numbers imply that APT_n is later than IPT_n. There is no negative figure so the vehicle did not leave from the bus stop before the passengers arrived there.

Fig. 10 shows the distribution of ADT_n−IDT_n. The negative figures denote that ADT_n is earlier than IDT_n. There is no positive figure so the vehicles did not arrive at passengers' destination before the arrival time informed to the passengers by the system.

3.2 Field test result

In order to validate the proposed on-demand bus service, field tests are conducted in three different type cities in Japan in 2009. These cities are Kashiwa City, Chiba Prefecture, Sakai City, Osaka Prefecture, Moriyama City and Shiga Prefecture. Table 3 shows the difference between overview of the three field tests.

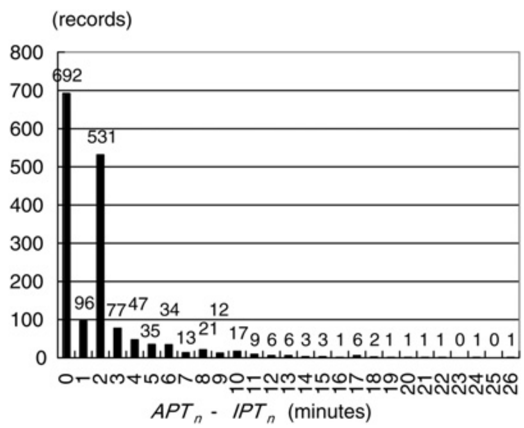

Figure 9 *Distribution of APT_n minus IPT_n*

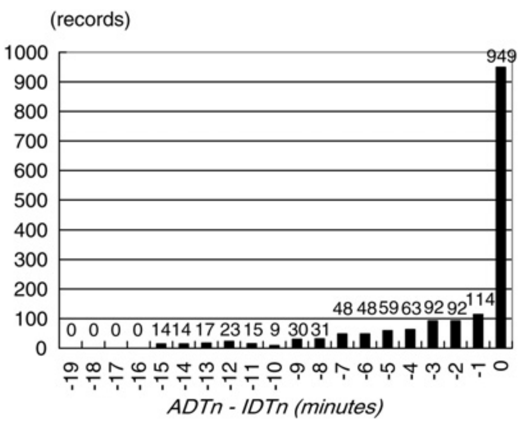

Figure 10 *Distribution of ADT_n minus IDT_n*

The accesses to the server are occurred from one month ago of starting service because there are some actual test and demonstration to the people and drivers. All accesses concerning these field tests in different cities receive a single server and the on-demand bus service in each city is provided. There is no trouble of cloud computing system.

3.2.1 Experiment in Kashiwa city: In the experiment in Kashiwa city, which is recognised as a commuters' town of Tokyo metropolitan area, 15 451 passengers used the five on-demand bus vehicles in 96 days (1248 h). In northern part of Kashiwa city, whose area is about 21.5 km^2, 384 bus stops are prepared for passengers.

As a result, the average ratio of operation time was 60.9%. Operation time means the time that the vehicle moved with passengers. So, vehicles moved without passengers in their 39.1% of travelling time. The average share-ride ratio was 38.2% of their operation time. This result shows that the on-demand bus reduced the travelling time by 38.2% in total, so the on-demand bus operation is friendly with the environment.

Table 3 Overview of field test

	Kashiwa	Sakai	Moriyama
area, km^2	about 21.5	about 4.8	about 54.8
vehicle	five vehicles	two vehicles	four vehicles
number of bus stops	384	209	300
date	2008/10/1–2009/2/28	2008/10/15–2008/11/14	2008/11/12–2008/12/25
period, days	96	30	44
running time in a day, h	13	8	8
total number of passengers	15 451	1331	1793
average ratio of operation time, %	60.9	32.9	64.8
average ratio of share-ride, %	38.2	48.8	36.5
willingness to pay	353 yen	220 yen	404 yen

It was shown that the characteristics of this system worked very well, and more than 45.9% of passengers were satisfied with the on-demand bus service because they can make a reservation via personal computers and they were able to specify dependable arrival time or pick-up time. About 36.1% of passengers answered that their chances of going out increased by the introduction of on-demand bus service. The result of willingness to pay (WTP) analysis is more than twice that of fixed buses (353 yen/ride). In summary, the result of the experiment showed that the developed on-demand bus service is a practical alternative mode for public transportation.

3.2.2 Experiment in Sakai city: A field test was held in Sakai city, which has a huge population but is a small size city in Osaka prefecture. A total of 209 bus stops are placed in 4.8 km^2 in the centre part of Sakai city. About 1331 passengers moved by the on-demand bus in 30 days.

Average ratio of operation time was 32.9%, and the ratio of share-ride was 48.8%. The share-ride ratio in Sakai city was higher than that in Kashiwa city. On the other hand, the operation time ratio is 32.9%, which is lower than that in Kashiwa city. The WTP in Sakai city was 220 yen, which is almost same with the fare of fixed buses. However, passengers evaluated the on-demand bus service as a practical alternative mode for public transportation.

3.2.3 Experiment in Moriyama city: Another test was also held in Moriyama city, which has little population over a large area. Field is very wide and four vehicles ran in 54.8 km^2 area for 44 days as on-demand bus. On-demand vehicles were operated 8 h a day and 1793 passengers rode the bus.

High operation time ratio of 64.8% shows that on-demand bus ran with little empty running. However, the share-ride ratio is 36.5%, so the share-ride rarely occurred. WTP in Moriyama city was 404 yen, which is almost twice the fare of fixed buses. In this case, passengers evaluated the on-demand bus service as a practical alternative mode for public transportation.

3.2.4 Comparison of the influence to the transportation mode in the city: The introduction of new transportation mode in the city influences other existing transportation system because people switch their transportation modes. Fig. 11 shows the result of questionnaire of the transportation mode which the on-demand bus passengers used before on-demand bus introduction to each areas.

First of all, the figure shows that over 30% of passengers switch their transportation mode from 'route bus' to on-demand bus. The proportion of switched passengers from 'bicycle' and 'car' is almost same in three field test. The result that 13–15% of passengers are from car user shows that this system is environmental friendly because it can promote to stop using car.

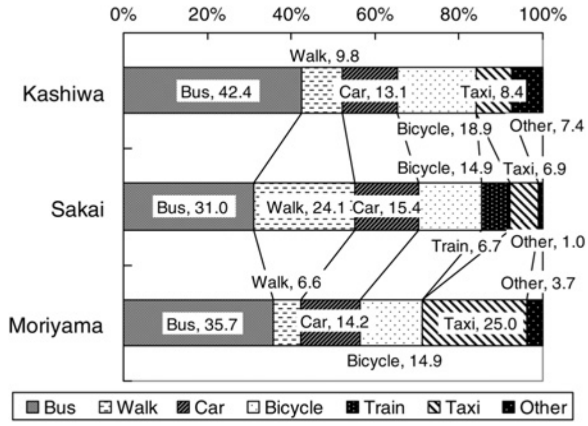

Figure 11 *Transportation mode before the introduction of on-demand bus service*

It is really interesting that similar tendency could be confirmed in three field tests.

4 Conclusions

The Innovative on-demand bus system is developed and the results of field tests show that the system is valid for different types of cities.

The developed on-demand bus system is totally different from the existing on-demand bus system from the viewpoint of cloud computing technology. Cloud computing system consists of four important parts: schedule calculation system, communication device, reservation interface and database. These software are implemented on the remote server and the local government officers can introduce the On-demand Bus service without any server system.

The schedule calculation system is valid by the computer simulation. The result of test proves that the calculation time does not increase exponentially even if the scale of problem becomes large. Moreover, the calculation time does not change by setting time limit of calculation when the new route can be found. These mean that the developed scheduling algorithm is practical in cloud computing technology.

The field test of the on-demand bus is held in Kashiwa city, Sakai city and Moriyama city in Japan at the same time. The result shows that the system provides the on-demand bus service through the Internet, and the result of every experiments concludes that the developed on-demand bus system is evaluated as practical and efficient. Moreover, passengers in each test answer the on-demand bus service promotes them to change into public transportation system.

To promote the developed system and to solve the transportation problem especially in the rural area are future work of this research. The utilisation of data accumulated in on-demand bus server is also one of the future works, so

doi: 10.1049/iet-its.2009.0113

that more efficient transportation system in the area can be divided. The route and schedule of the route bus can be optimised with the data in on-demand bus server.

This system is expected to apply for car sharing, taxi services, freight pickups and so on because the principles of them are really similar with the on-demand bus service. To expand the application of this system is also one of the future works.

5 Acknowledgments

The authors would like to express their sincere gratitude to the support of Core Research for Evolutional Science and Technology (CREST) and the Japan Science and Technology Agency (JST).

6 References

[1] DIANA M., DESSOUKY M.M., XIA N.: 'A model for the fleet sizing of demand responsive transportation services with time Windows', *Transp. Res.*, 2006, **40B**, (8), pp. 651–666

[2] STEIN D.: 'Scheduling dial-a-ride transportation system', *Transp. Sci.*, 1978, **12**, pp. 232–249

[3] PSARAFTIS H.: 'A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem', *Transp. Sci.*, 1980, **14**, pp. 130–154

[4] PSARAFTIS H.: 'An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time window', *Transp. Sci.*, 1983, **17**, pp. 351–357

[5] CULLEN F., JARVIS J., RATLIFF H.: 'Set partitioning based heuristics for interactive routing', *Networks*, 1981, **11**, pp. 125–144

[6] SEXTON T.: 'The single vehicle many to many routing and scheduling problem'. PhD thesis, State University of New York at Stony Brook, 1979

[7] JAW J., ODONI A., PSARAFTIS H., WILSON N.: 'A heuristic algorithm for the multi-vehicle advance-request dial-a-ride problem with time windows', *Transp. Res.*, 1986, **20B**, pp. 243–257

[8] ALFA A.S.:: 'Scheduling of vehicles for transportation of elderly', *Transp. Plan. Technol.*, 1986, **11**, pp. 203–212

[9] RHEE J.H.:: 'Vehicle routing and scheduling strategies for demand responsive transportation systems'. PhD dissertation, University of Delaware, Newark, 1987

[10] KIKUCHI S., RHEE J.H.:: 'Scheduling method for demand responsive transportation system', *J. Transp. Eng.*, 1989, **115**, pp. 630–645

[11] VUKADINOVIC N.:: 'Fuzzy relation based vehicle scheduling algorithm: application to the demand responsive system', MSc thesis. University of Delaware, Newark, 1992

[12] LI H.B., LIM A.: 'Local search with annealing-like restarts to solve the VRPTW', *Eur. J. Oper. Res.*, 2003, **150**, (1), pp. 115–127

[13] FABRI A., RECHT P.: 'On dynamic pickup and delivery vehicle routing with several time windows and waiting times', *Transp. Res.*, 2006, **40B**, (4), pp. 335–350

[14] KOTA T., KAZUO H., HIROYUKI Y.: 'Scheduling algorithm for on-demand bus system'. Proc. Sixth Int. Conf. on Information Technology: New Generations (ITNG), 2009, pp. 189–194