IPSJ Transactions on Mathematical Modeling and Its Applications Vol.12 No.2 1–11 (July 2019)

Regular Paper

Modeling and Evaluating Taxi Ride-sharing for Event Trips

Taketo Yoshida[1,a) Masaki Yano[1] Kenichiro Horikawa[2] Keita Sato[2] Shota Minami[1] Maiko Shigeno[1]

Received: January 31, 2018, Revised: March 24, 2018,
Accepted: January 15, 2019

Abstract: While ride-sharing systems have received great interest and spread widely in recent years, taxi ride-sharing is expected to be highly effective at ride-sharing. This research is an investigation into the possibility of taxi ride-sharing for passengers having a common purpose, such as an event trip, in which passengers having the same reason for taking a trip are handled. Although solutions to taxi ride-sharing problems are usually evaluated on the basis of the distance traveled by taxis, our proposed model minimizes the total trip distance of all passengers without lengthening the minimum total distance traveled by the taxi. This taxi ride-sharing problem is formulated as a mixed integer linear programming (MILP) problem. For this problem, an exact algorithm under the restriction of ride capacity and a heuristic algorithm that solves general cases are proposed. Moreover, numerical experiments were done to assess the performance of our heuristic algorithm and evaluate solutions in terms of distances of routes and fare that is paid.

Keywords: taxi ride sharing, mathematical modeling, mixed integer linear programming, matching algorithm, heuristic algorithm

1. Introduction

Ride-sharing systems have received great interest and spread widely in recent years because of their possibility to reduce travel costs and to overcome problems with environmental pollution and traffic congestion. It is said that the beginning of organized ride-sharing systems was in WWII [6], [8]. During the last decade, new tools such as smartphones and the global positioning system (GPS) have likely been the cause of the success of ride-sharing systems. Great attention is paid nowadays to ride-sharing services such as those that arrange many drivers and passengers dynamically, that is, that repeatedly rearrange drivers and passengers when they send their information [2], [7], [11], [18]. There are various types of ride-sharing as we can see in reviews [1], [8]. As a practical matter, system must be designed and analyzed depending on the characteristics of the target ride-sharing service.

In Japan, ride-sharing businesses that use personal automobiles are not allowed under the law. However, taxi ride-sharing systems have been introduced in order to support the public transportation system in local communities. These transportation services are called "community taxis." Recently, smartphone applications that support taxi ride-sharing systematically have been released and are used in urban areas. Such applications provide sharing services that can be regarded as an extension of the unorganized ride-sharing that is done within personal relationships. Moreover, trial experiments on taxi ride-sharing have been performed in several regions. For example, the Ministry of Land, Infrastructure, Transport and Tourism has started the experiments in Tokyo from January, 2018. There is nevertheless very little research done through case studies on taxi ride-sharing in Japan from the aspect of optimization models.

This research is an investigation into the possibility of taxi ride-sharing under mutual consent of passengers having a common purpose. We focus only on event trips, where passengers have the same reason for taking a trip, for example, attending events or going to stadiums. Since we assume each event has a start time, each passenger must reach the destination just before this time. Thus, we do not need to consider time window constraints and do not need to deal with dynamical cases. In addition, having a common destination makes the problem simple. This model is equivalent to the problems discussed in Massobrio, Fagúndez and Nesmachnow [20] and Ben-Smida et al. [4]. Their models treat ride-sharing for passengers from the same origin to distinct destinations.

Our taxi ride-sharing problem is to find an assignment of passengers to taxis and a tour of each taxi. Usually, solutions to taxi ride-sharing problems are evaluated in terms of transportation costs and passenger satisfactions. Transportation costs are calculated on the basis of the total distance traveled by taxis. Passenger satisfaction is expressed in terms of short detours through ride-sharing, compatibility between riding partners, and so on. It is reported in [21] that event participants who shared a ride in practice said that a large detour made to pick up other riders for ride-sharing was unacceptable. Therefore, the objective of our model is to minimize the total trip distance over all passengers

[1] University of Tsukuba, Tsukuba, Ibaraki 305–8573, Japan
[2] DENSO CORPORATION, Nihonbashi, Tokyo 103–6015, Japan
[a)] s1720607@s.tsukuba.ac.jp

instead of the distance traveled by taxis. This objective can address the issue of both transportation distance and a detour. For this model, we show an exact algorithm that uses matching methods under the condition that at most two passengers can share a ride. In addition, we propose a heuristic algorithm for general cases in which three or more passengers can share one taxi. Having a common destination may be enable us to use a heuristic algorithm based on geographical decomposition, which tends to fail in general ride-sharing problems. Our heuristic algorithm is based on clustering methods that uses cosine similarity.

The rest of this paper is organized as follows. In Section 2, related work on taxi ride-sharing is reviewed. In Section 3, our taxi ride-sharing problem for event trips is described, and mixed integer linear programming (MILP) formulations are given. Section 4 gives algorithms for our taxi ride-sharing problem. Numerical experiments evaluating our algorithms are shown in Section 5. Finally, in Section 6, we summarize our results and discuss future work.

2. Related Work

Taxis provide a more flexible, comfortable, and faster transportation service than buses, railway, subways and so on. However, the cost of taxi fares is higher. Moreover, taxis usually have low occupancy rates, that is, they have many empty car seats, even at times of peak traffic. Hence, taxi ride-sharing is expected to be highly effective. Taxi ride-sharing problems have now become one of the major topics in ride-sharing.

Taxi ride-sharing research is classified into two types. One considers coordination between drivers and passengers. To offer this service with smartphone applications, dynamical systems have been developed. For this type of service, Agatz et al. [2] proposed a method that iteratively solves the assignment problem. Bicocchi and Mamei [5] presented a system for finding appropriate ride-sharing partners automatically from mobile data. Ma et al. [17] and Huang et al. [11] formulated real-time ride-sharing problems and proposed heuristic algorithms that are efficient on a large scale. Schreieck et al. [26] developed matching algorithms for larger numbers of ride requests in real-time. Fu et al. [7] focused on matchings concerned with the cohesion of passengers in terms of social relations in order to improve passenger comfort and safety. As the commercial platforms for ride-sharing have become popular, progress has been make on high performance algorithms for matching in dynamic systems under many conditions in terms of practicality.

The other type of ride-sharing researche considers constructing a tour for each taxi by scheduling a pick up and drop off sequence for assigned passengers. This scheduling problem is regarded as a special case of dial-a-ride problems (see, for example, one review paper [22]). For the problem, both static and dynamic versions are considered. The typical objective function is to minimize the total travel distance. However, many evaluation criteria are considered. Lin et al. [16] adopted operation costs and passenger satisfaction, such as extra riding and waiting times. Hosni et al. [10] used benefits for drivers. Santi et al. [24] evaluated the trade-off between benefits and passenger inconvenience in ride-sharing. Santos and Xavier [25] introduced a multi-objective function consisting of maximizing the number of served passengers and minimizing the total cost of passengers. Lee and Savelsbergh [12] formulated a ride-sharing problem that compares the benefits and costs with ad-hoc drivers and dedicated drivers. Ma et al. [19] analyzed the distribution of benefits between passengers and drivers. As many evaluation criteria have been discussed, it is important to adopt appropriate criteria matched to the characteristics of ride-sharing situations.

To pursue efficiency in urban traffic, more complex models have been discussed, such as multi-modal sharing systems [14], [15], [18], [27] and multi-commodity sharing systems [13]. Meanwhile, one of the simplest but most significant cases of taxi ride-sharing is for passengers from the same origin heading to different destinations. For this problem, Massobrio et al. [20] proposed a non-deterministic method so called the evolutionary algorithm. Ben-Smida et al. [4] formulated the problem as a mixed integer linear programming (MILP) problem and compared it with Massobrio, Fagúndez and Nesmachnow's method in numerical experiments. The model discussed in Tao and Chen [28] also has one origin, but their model is dynamic in accordance with passengers' time windows. Recently, Qian et al. [23] designed a taxi group-ride problem in which passengers are grouped in a single ride. A similar idea is appeared in ride-sharing having transportation hubs in a paper by Lin et al. [15]. The taxi group-ride problem finds a group of passengers whose trips are close to each other in spatial terms, similar to finding groups of passengers having nearby destinations for sharing a taxi from the same origin. As many taxi ride-sharing models have been investigated, we should design a suitable model that represents the characterization of the services we consider.

In our research, we construct models for taxi ride-sharing problems with a focus on event trips. Corresponding to the result of a questionnaire survey [21], we focus not on only the total distance traveled by taxis, which directly affect the fare cost, but also on routes for passengers as the evaluation criteria.

3. Model Description and Formulation

Assume that there are n participants who go by taxi ride-sharing to an event for the purpose of meeting people having a common lifestyle and similar tastes, like parents circles for child care. In this case we do not need to consider preferences toward ride-sharing partners. That is to say, participants are matched only by spatial constraints.

Let N be a set of n participants. Participant $i (\in N)$ is characterized by the pick up location p_i and the number of passengers who go together with i, including i, f_i. For example, if participant i goes with his/her partner, f_i becomes 2. We call passengers going together with participant i "fellow passengers." The common destination where an event is held is denoted by p_g and a dummy starting place is denoted by p_o. The distance d_{ij} from location p_i to location p_j is known. For convenience, $\tilde{N} = N \cup \{o, g\}$ and $f_o = f_g = 0$ and $d_{oj} = 0$ for any $j \in \tilde{N}$. The ride capacity of each vehicle is F. Usually, F is set to 3 or 4 for taxis. A taxi ride-sharing problem for an event trip involves dividing participants into groups, to each of which a taxi is assigned, and deciding the visiting order for picking up participants in each group. In other words, it involves finding a set of tours that taxies travel along from p_o to p_g. The pick up location of each participant should belong to one of these tours.

First, we consider the problem of minimizing the total distance traveled by taxis. This taxi ride-sharing problem, called TRSP_T, is formulated as a MILP formulation. The following formulation (1)–(8) is slightly different from that adopted in [4].

$$\begin{aligned}
\text{minimize} \quad & \sum_{i,j\in\tilde{N}} d_{ij}x_{ij} && && (1)\\
\text{subject to} \quad & \sum_{\substack{i\in\tilde{N}\\ i\neq j}} x_{ij} = 1 && \forall j \in N && (2)\\
& \sum_{\substack{j\in\tilde{N}\\ j\neq i}} x_{ij} = 1 && \forall i \in N && (3)\\
& \sum_{i\in N} x_{io} = \sum_{j\in N} x_{gj} = 0 && && (4)\\
& u_i - u_j + F x_{ij} \leq F - f_i && \forall i,j \in \tilde{N} && (5)\\
& x_{ij} \in \{0,1\} && \forall i,j \in \tilde{N} && (6)\\
& u_i \geq 0 && \forall i \in N && (7)\\
& u_o = 0, && && (8)
\end{aligned}$$

where x_{ij} is a Boolean variable that equals 1 if a taxi picks up participant j just after participant i, and u_i is a nonnegative variable called a "potential." Objective function (1) minimizes the total distance traveled by taxis except for routes to visit the location of participants who ride first for each tour. Constraints (2)–(4) guarantee that several tours are constructed in which each participant is visited exactly once for any of them. Constraint (5) is a variant of the Miller-Tucker-Zemlin constraint for traveling salesman problems (TSP). The constraint eliminates subtours. That is to say, this ensures that all tours start from the dummy starting point p_o and finish at the common destination p_g. In addition, this constraint also ensures that u_i indicates the number of passengers in a taxi when it arrives at p_i, because inequality (5) becomes $u_i + f_i \leq u_j$ if $x_{ij} = 1$, that is, a taxi goes to p_j just after p_i. Moreover, the inequality (5) for $i = g$ and $j = o$, together with $x_{go} = 0$ and $u_o = 0$, becomes $u_g \leq F$, which implies that this constraint guarantees that the number of passengers sharing a taxi is not greater than ride capacity F.

It is important that taxi tours are evaluated in terms of the total distance traveled because the taxi fare depends on its mileage. However, some tours are not accepted by participants. From a questionnaire survey [21], one of the complaints of participants was roundabout routes that were long, especially those that involve driving in a direction different from the destination. When we adopt tours minimizing the total distance traveled by taxis, there may be a long detour for participants. For example, we consider the locations shown in **Fig. 1**. The distance between locations, shown in **Table 1**, corresponds approximately to the Euclid distance and is symmetric. We assume that one person will catch a taxi at each location, i.e, $f_i = 1$ for all i, and that $F = 2$. The minimum total distance traveled by taxis is 60, where two tours, p_1-p_2-p_g and p_3-p_4-p_g, are adopted. However, a long detour is

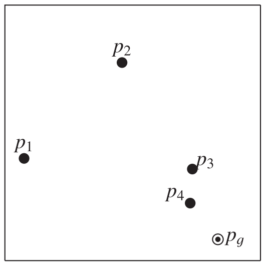


Fig. 1 Example of locations for 4 participants and common destination.

Table 1 Distance matrix for Fig. 1.

	p_2	p_3	p_4	p_g
p_1	19	24	25	30
p_2		18	23	28
p_3			5	10
p_4				8

imposed on participant 1. In this case, two other tours, p_1-p_4-p_g and p_2-p_3-p_g, are superior for every participant. To avoid long detours, we modify TRSP_T by adding a constraint that restricts the increase in the trip distance for each participant i from the distance d_{ig} that participant i takes to the destination directly [*1]. When dynamic situations are considered, such a maximum detour constraint is introduced in many cases [3], [26]. However, in our model, a solution that satisfies a maximum detour constraint usually requires more taxis than the original TRSP_T. Thus, the problem has parameters with respect to restricting increments in trip distance and increments in the number of taxis, which implies that we need to adjust such parameters in order to balance these restrictions. To not use such parameters, we now introduce another objective function for evaluating a set of tours. To evaluate extra trips taken by participants, we use the total trip distance of all participants. In the two tours p_1-p_4-p_g and p_2-p_3-p_g shown in the above example, trip distances for participants 1, 2, 3, and 4 are $d_{14} + d_{4g} = 33$, $d_{23} + d_{3g} = 28$, $d_{3g} = 10$, and $d_{4g} = 8$, respectively. Thus, the total trip distance of all participants is given by $33 + 28 + 10 + 8 = 79$. For the tours which is an optimal solution of TRSP_T, the total trip distance of all participants is given by $47 + 28 + 13 + 8 = 96$. If our objective were to minimize the total trip distance of all participants, the former tours would be adopted, even though this is not optimal for TRSP_T. Note that if the objective function (1) were changed to minimize the total trip distance of all participants under the constraints (2)–(8), no participants would ride with other participants in an optimal solution. That is to say, each participant would ride in a taxi alone. That would be meaningless. Thus, we set our objective to minimizing the total trip distance of all participants, without making the length of the minimum total distance traveled by taxis too long.

We now consider a taxi ride-sharing problem that minimizes the total trip distance of all participants, without counting their fellow passengers, under the condition that they ride separately in exactly k taxis. The problem is referred to as TRSP_P. To simplify our discussion, we assume that each participant has no fellow passengers, i.e., $f_i = 1$ for all $i \in N$ from now on. The problem is formulated as follows.

[*1] Strictly speaking, we need additional variables to represent the trip distance for each participant when we add such an additional constraint.

$$\text{minimize} \quad \sum_{i,j\in\tilde{N}} d_{ij}y_{ij} \tag{9}$$

$$\text{subject to} \quad \sum_{\substack{i\in\tilde{N}\\ i\neq j}} x_{ij} = 1 \qquad \forall j \in N \tag{10}$$

$$\sum_{\substack{j\in\tilde{N}\\ i\neq j}} x_{ij} = 1 \qquad \forall i \in N \tag{11}$$

$$\sum_{i\in N} x_{io} = \sum_{j\in N} x_{gj} = 0 \tag{12}$$

$$\sum_{j\in N} x_{oj} = \sum_{i\in N} x_{ig} = k \tag{13}$$

$$v_i - v_j + (F+2)x_{ij} \leq F+1 \qquad \forall i,j \in \tilde{N} \tag{14}$$

$$y_{ij} \geq v_i + x_{ij} - 1 - F(1-x_{ij}) \qquad \forall i,j \in \tilde{N} \tag{15}$$

$$x_{ij} \in \{0,1\} \qquad \forall i,j \in \tilde{N} \tag{16}$$

$$v_i \geq 1 \qquad \forall i \in N \tag{17}$$

$$v_o = 0 \tag{18}$$

$$y_{ij} \geq 0 \qquad \forall i,j \in \tilde{N}, \tag{19}$$

where y_{ij} stands for the number of participants riding a taxi when the taxi drives from location p_i to location p_j directly, and v_i represents the number of participants in a taxi when it departs from location p_i. Constraint (14), which is a variant of Constraint (5), eliminates subtours and ensures that v_i is the number of participants in a taxi. In addition, when $x_{ig} = 1$, the constraint becomes $v_i \leq v_g - 1$, which, together with $v_g \leq F + 1$ obtained by this constraint for $i = g$ and $j = o$, indicates that $v_i \leq F$. Constraint (15) represents y_{ij} as the number of participants between locations p_i and p_j. It becomes $y_{ij} \geq v_i$, if $x_{ij} = 1$, and $y_{ij} \geq v_i - F - 1$ which is trivially satisfied, otherwise. If we need to consider fellow passengers, all it takes is to add constraints (5), (7) and (8) to this MILP formulation (9)–(19). Note that tours obtained by this formulation depend on the number of taxi vehicles k shared by participants. In our problem, we employ the optimal number of taxis in TRSP_T as k.

4. Algorithms

We first consider algorithms for the taxi ride-sharing problem under the condition that at most two participants can share a vehicle, that is, $F = 2$. When the event that participants go to by ride-sharing is a parent circle for child care, participants take their children along. In such a case, at most two participants and their children can share a vehicle, where the pairing possibility of participants is determined in accordance with the number of children. Since the problem is to find appropriate pairs of participants, it can be induced in a matching problem on a general graph. Let $G = (N, E)$ be a graph, where the vertex set corresponds to the set of participants N, and edge set E is composed of edges representing possible pairs of two participants. Note that, for a feasible solution x of TRSP_T under the condition that at most two participants can share a vehicle, $\{(i, j) \in N \times N \mid x_{ij} = 1\}$ is a matching in G. A matching M of G corresponds to solution x^M, which gives the following. $x^M_{oi} = x^M_{ig} = 1$ if node i is not incident to any edge in M; $x^M_{oi} = x^M_{ij} = x^M_{jg} = 1$ or $x^M_{oj} = x^M_{ji} = x^M_{ig} = 1$ if $(i, j) \in M$; $x^M_{ij} = 0$ for remaining pairs of i and j in $\tilde{N}$. Since edge (i, j) in matching M is associated with two tours through i and j, we assign the one with the minimum distance to it. For TRSP_T, the weight of edge $(i, j) \in E$ is given by the distance saved due to pairing of participants i and j, that is,

$$d_{ig} + d_{jg} - \min\{d_{ij} + d_{jg}, d_{ji} + d_{ig}\}.$$

Since $d_{oj} = 0$ for any $j \in \tilde{N}$, the weight of matching M can be transformed to the weight of the corresponding feasible solution x^M of M.

$$\begin{aligned}
&\sum_{(i,j)\in M} \Big(d_{ig} + d_{jg} - \min\{d_{ij} + d_{jg}, d_{ji} + d_{ig}\}\Big) \\
&= \sum_{i\in N} d_{ig} - \sum_{i\notin M} d_{ig} - \sum_{(i,j)\in M} \min\{d_{ij} + d_{jg}, d_{ji} + d_{ig}\} \\
&= \sum_{i\in N} d_{ig} - \sum_{i\notin M}\sum_{j\in\tilde{N}} d_{ij}x^M_{ij} \\
&\quad - \sum_{(i,j)\in M} \Big((d_{ij}x^M_{ij} + d_{jg}x^M_{jg}) + (d_{ji}x^M_{ji} + d_{ig}x^M_{ig})\Big) \\
&= \sum_{i\in N} d_{ig} - \sum_{i,j\in\tilde{N}} d_{ij}x^M_{ij}
\end{aligned}$$

where, for convenience, we denote by $i \notin M$ such that vertex i is not incident to any edge in M. Therefore a maximum weight matching in G corresponds to an optimal solution for TRSP_T.

In regard to TRSP_P, we consider a graph $\hat{G} = (N \cup \hat{N}, E \cup \hat{E})$, where $\hat{N}$ is a set of $2k - n$ dummy vertices and $\hat{E} = \{(i, j) \mid i \in N, j \in \hat{N}\}$. A perfect matching in $\hat{G}$ corresponds to a solution to ride sharing using exactly k taxis. Edge $(i, j) \in \hat{E}$ is contained in a perfect matching, if and only if participant i rides alone. Edge $(i, j) \in E$ is contained in a perfect matching, if and only if participants i and j share the same taxi, where the taxi tour chosen is the one with the minimum total trip distance. Since $2k - n$ dummy vertices are added, there are $2k - n$ participants who ride alone. Thus, the remaining $n - (2k - n) = 2(n - k)$ participants share taxis in pairs. Then, the number of vehicles needed for sharing is $(2k - n) + \frac{2(n-k)}{2} = k$. We give a weight of edge $(i, j) \in E$ by calculating the total tour distance of passengers i and j, that is,

$$\min\{d_{ij} + 2d_{jg}, d_{ji} + 2d_{ig}\},$$

and a weight of edge $(i, j) \in \hat{E}$ by using the trip distance when participant i rides in a taxi alone, that is, d_{ig}. Then the weight of a perfect matching is equivalent to the total trip distance of all participants corresponding to the ride-sharing solution. Thus, a minimum weight perfect matching corresponds to an optimal solution for TRSP_P. From the above, under the condition that at most two participants can share a vehicle, both TRSP_T and TRSP_P can be solved in polynomial time by matching algorithms.

We now turn to general cases of taxi ride-sharing problems, that is, three or more participants can share one vehicle. Since taxi ride-sharing problems are NP-hard for general cases [4], we developed a heuristic algorithm. Since TRSP_T is used to find a suitable number of taxis, our target problem is TRSP_P. Our heuristic algorithm finds a suitable number of taxis as well as the tour of each taxi that is assigned a cluster of participants, which is obtained by geographical decomposition centering on a common destination like a classical heuristic algorithm of the sweep method for vehicle routing problems (VRPs) [9]. Let S be a cosine similarity matrix derived from the relative degrees of latitude and longitude of each location p_i for $i \in N$ from the destination p_g. Namely, by denoting the latitude and longitude of p_i as lat_i and long_i, the (i, j)-element s_{ij} of S is given by

$$\frac{(\mathrm{lat}_i-\mathrm{lat}_g)(\mathrm{lat}_j-\mathrm{lat}_g)+(\mathrm{long}_i-\mathrm{long}_g)(\mathrm{long}_j-\mathrm{long}_g)}{\sqrt{(\mathrm{lat}_i-\mathrm{lat}_g)^2+(\mathrm{long}_i-\mathrm{long}_g)^2}\sqrt{(\mathrm{lat}_j-\mathrm{lat}_g)^2+(\mathrm{long}_j-\mathrm{long}_g)^2}}.$$

Then, we partition the set of participants N into $X_1, X_2, \ldots, X_{k'}$ i.e., $\bigcup_{l=1}^{k'} X_l = N$ and $X_l \cap X_{l'} = \emptyset$ for all$1 \le l < l' \le k'$, by using an appropriate clustering algorithm for S. Clustering algorithms, however, do not consider the ride capacities of taxis F. Thus, we apply a clustering algorithm iteratively until the ride capacity is satisfied. A detailed description of our algorithm is given in Algorithm 1. Here, two threshold parameters, θ and α, are used. Recall that, for convenience, we assume that $f_i = 1$ for all $i \in N$ in this algorithm.

For the partition $\mathcal{X} = \{X_1, X_2, \ldots, X_{k'}\}$ of N obtained by Algorithm 1, each tour of a taxi that is assigned to X_l $(l = 1, \ldots, k')$ is found by calculating the distance of every visiting sequence for participants in X_l when F is not that large. When F is too large to enumerate all visiting sequences for participants in a taxi, we employ an appropriate method for traveling salesman problems (TSPs). The total distance traveled by taxis in this solution tends to become long because the obtained partition $\mathcal{X}$ does not consider the constraints on the number of taxis and the number of partitions is usually large. Thus, we modify the obtained partitions by combining clusters. We again adopt matching algorithms. Let $G_{\mathcal{X}} = (\mathcal{X}, E_{\mathcal{X}})$ be a graph with vertex set $\mathcal{X}$ and edge set $E_{\mathcal{X}} = \{(X_l, X_{l'}) \in \mathcal{X}\times\mathcal{X} \mid |X_l|+|X_{l'}| \le F, s_{ij} \ge \eta, i \in X_l, j \in X_{l'}\}$, where η is a threshold parameter less than θ. A weight of edge $(X_l, X_{l'}) \in E_{\mathcal{X}}$ is given by the distance saved due to merging a pair of clusters X_l and $X_{l'}$, that is,

$$M - \mathrm{dist}(X_l \cup X_{l'}),$$

where $\mathrm{dist}(X)$ stands for the minimum trip distance of all participants in X, and M is a large constant number so that as many edges as possible are chosen in matching. Then, we find a maximum weight matching in $G_{\mathcal{X}}$ and combine a pair of clusters X_l and $X_{l'}$ if edge $(X_l, X_{l'})$ is contained in the maximum weight matching. According to necessity, we repeatedly find maximum weight matchings and combine clusters. Summarizing the above, our heuristic algorithm is described in Algorithm 2.

Algorithm 1 iterative clustering

```
[initialization] h := 2, X := ∅ and Nr := N;
repeat
    obtain a partition X_1, X_2, ..., X_h of Nr by using a clustering algorithm
    for S;
    if |X_l| > F for all l = 1, ..., h then
        h ← h + 1;
    else
        for all X_l (l = 1, ..., h) such that |X_l| ≤ F do
            if |X_l| ≤ α or s_ij ≥ θ for any i, j ∈ X_l then
                add X_l to X and update Nr ← Nr \ X_i;
            end if
        end for
        h ← 2;
    end if
until |Nr| ≤ h
output X;
```

Algorithm 2

```
find partition X of N by using the iterative clustering algorithm (Algorithm
1);
construct a tour of a taxi to which each X ∈ X is assigned;
repeat
    find a maximum weight matching in G_X and combine pairs of clusters
    corresponding to the matching;
    update a tour of a taxi to which each combined cluster is assigned;
until there are no appropriate combinations of clusters
```

5. Numerical Experiments

We performed numerical experiments in order to investigate the performance of our heuristic algorithm and to evaluate the solution to TRSP_P in terms of both transportation distance and detours for each participant. In addition, the fares of each participant was also compared.

5.1 Experimental Setup

Our instances were based on three city areas: Kashiwa, Totsuka, and Sendai. The taxi fare differs slightly among these areas. Typical fares for these city areas are shown in **Table 2**. We generated a location p_i for each participant on the basis of facilities for primary education, such as elementary schools and private tutors, extracted from Google Maps by setting a 10-kilometer radius arond the main station of each city. We constructed an instance with n participants by choosing n from 25 facilities extracted in advance. The main station was set as the common destination. The distance between locations was obtained by using the Google Map API.

We implemented the algorithms discussed in Section 4 with Python 3.5.2. For our iterative clustering algorithm (Algorithm 1), we employed k-medoids as the clustering method. Since the solution depends on an initial solution for k-medoids, our heuristic algorithm output the best solution among the results by performing the algorithm 15 times. That is to say, it returned a solution with the shortest total trip distance for all participants among solutions that achieve the minimum number of vehicles. Threshold parameters were given by $\theta = 0.9, \alpha = F, \eta = 0.75$.

The MILP formulations for TRSP_T and TRSP_P were solved by using FICO Xpress Optimizer 27.01.02 on an HP Pavilion HPE h8-1090JP with an Intel Core i7, 3.20 GHz-CPU and 12.0 GB of RAM. Both MILP formulations were solved by adding valid inequalities in order to accelerate the computational time. We adopted three types of valid inequalities

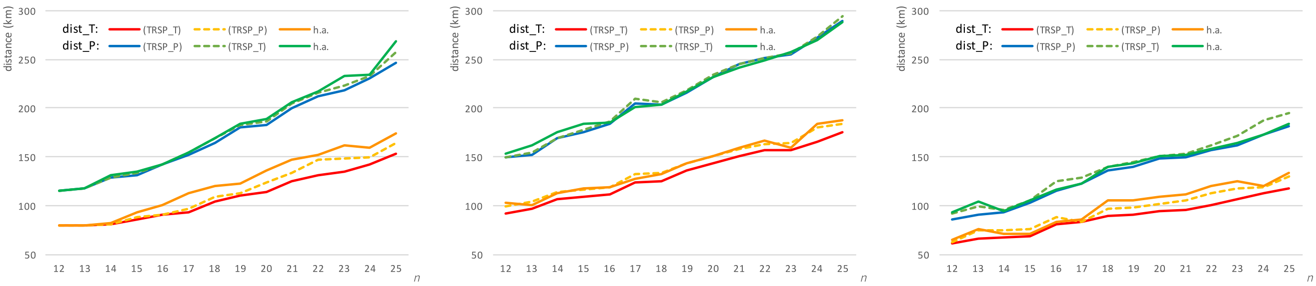
Table 2 Typical taxi fares in Totsuka, Kashiwa and Sendai.

	Kashiwa	Totsuka	Sendai
base fare	730 yen	730 yen	680 yen
maximum distance by the base fare	2.0 km	2.0 km	1.7 km
added fare	90 yen per 290 m	90 yen par 293 m	80 yen per 238 m

IPSJ Transactions on Mathematical Modeling and Its Applications Vol.12 No.2 1–11 (July 2019)

Table 3 Comparing evaluation values when $F = 2$.

		n	12	13	14	15	16	17	18	19	20	21	22	23	24	25
Kashiwa	TRSP_T	k	7	7	7	8	8	9	10	10	11	11	12	12	12	13
		dist_T	79.66	79.51	81.24	85.30	90.17	93.60	103.61	110.75	114.07	125.07	131.17	134.58	142.35	153.31
		dist_P	115.23	118.11	129.05	133.11	142.02	153.80	169.33	182.77	186.09	204.48	215.87	223.38	232.46	257.93
	TRSP_P	dist_T	79.69	79.51	81.24	88.72	90.17	97.02	109.47	113.18	124.21	134.11	146.79	148.12	149.97	163.74
		dist_P	115.07	118.11	129.05	131.38	142.02	152.07	164.52	179.69	182.58	199.32	212.00	218.16	229.85	246.45
	h.a.	k+	0	0	0	0	+1	+1	0	0	0	0	0	0	+1	0
		dist_T	79.69	79.51	82.43	93.16	100.82	113.20	119.85	123.04	135.70	146.44	152.47	162.04	159.43	174.16
		dist_P	115.07	118.11	131.09	134.44	141.90	154.48	169.66	184.31	188.31	206.39	217.49	232.43	233.92	268.91
	total distance by riding alone		119.90	123.12	132.33	136.39	143.20	155.62	168.53	182.01	185.34	210.24	222.92	227.75	237.33	255.99
Totsuka	TRSP_T	k	7	8	8	9	9	9	10	11	11	11	12	12	13	13
		dist_T	92.29	96.58	106.53	109.16	111.99	123.23	125.43	136.27	143.67	151.18	156.49	157.45	165.65	174.71
		dist_P	149.57	153.86	169.68	178.11	186.81	210.18	205.55	218.57	233.92	245.72	251.92	256.37	273.96	293.71
	TRSP_P	dist_T	99.51	103.98	113.76	116.57	119.39	132.60	133.00	143.84	151.08	157.82	163.13	164.09	180.59	184.00
		dist_P	149.37	151.40	169.48	175.65	184.34	204.28	202.89	215.91	231.46	244.94	251.13	255.58	272.08	289.27
	h.a	k+	0	−1	0	−1	0	+1	0	0	0	+1	+1	0	+1	+1
		dist_T	102.84	100.55	112.30	117.42	119.34	127.82	132.78	143.62	151.02	158.71	166.52	159.25	183.28	187.63
		dist_P	152.84	162.26	175.59	183.37	184.93	201.20	203.67	216.69	232.04	241.53	249.35	257.29	269.59	287.72
	total distance by riding alone		138.95	143.24	157.35	167.44	173.31	187.09	192.03	207.76	222.16	233.71	240.82	244.31	260.81	276.49
Sendai	TRSP_T	k	7	8	8	8	9	9	10	10	10	11	11	12	12	13
		dist_T	61.56	66.31	67.81	69.00	80.77	82.87	89.85	90.77	94.65	96.06	100.35	106.40	112.41	118.16
		dist_P	92.21	99.24	95.52	105.66	125.04	129.23	140.19	144.31	151.32	153.19	162.24	172.17	187.73	194.47
	TRSP_P	dist_T	62.49	74.16	74.07	75.26	87.68	83.80	97.34	98.25	101.25	105.26	112.96	117.78	119.05	129.29
		dist_P	85.54	90.65	92.94	103.08	115.50	122.55	136.09	140.21	148.48	148.93	157.42	162.24	173.05	181.67
	h.a.	k+	0	−1	+1	+1	+1	+1	+1	+1	+1	+1	+2	+1	+1	+1
		dist_T	65.23	75.39	70.65	70.81	83.09	85.94	105.26	105.46	109.13	110.89	119.73	125.41	120.57	133.56
		dist_P	92.65	103.59	94.77	105.09	116.16	122.64	139.89	143.30	150.56	151.42	157.80	164.79	172.55	183.26
	total distance by riding alone		81.61	87.32	89.70	99.88	112.30	116.67	131.16	134.36	141.43	143.15	149.77	157.50	167.04	177.66



Kashiwa Totsuka Sendai

Fig. 2 Comparing dist_T and dist_P for $F = 2$.

$$u_i - u_j + Fx_{ij} + (F - f_i - f_j)x_{ji} \leq F - f_i, \quad \forall i, j \in N$$
$$1 + (1 - x_{oi}) \leq u_i, \quad \forall i \in N$$
$$u_i \leq F - f_i - (1 - x_{ig}) - (F - 2)x_{oi}, \quad \forall i \in N$$

for TRSP_T, and six types of valid inequalities

$$v_i - v_j + (F + 2)x_{ij} + Fx_{ji} \leq F + 1, \quad \forall i, j \in N$$
$$1 + (1 - x_{oi}) \leq v_i, \quad \forall i \in N$$
$$v_i \leq F - (1 - x_{ig}) - (F - 2)x_{oi}, \quad \forall i \in N$$
$$x_{ij} \leq y_{ij} \leq Fx_{ij}, \quad \forall i, j \in \tilde{N}$$
$$y_{ij} \leq v_i, \quad \forall i \in N, \forall j \in \tilde{N}$$
$$v_i = \sum_{j \in \tilde{N}} y_{ij}, \quad \forall i \in N$$

for TRSP_P.

5.2 Comparing Performance of Algorithms

We compared the performance between our heuristic and exact algorithms by evaluating the obtained tours and computational time for the cases with a ride capacity of $F = 2$ and 3. To evaluate the tours obtained by each algorithm, we measured three numerical values: the total distance traveled by all taxis (dist_T), the total trip distance of all participants (dist_P), and the number of taxis needed for sharing (k). Recall that our main purpose is to minimize dist_P without making the length of dist_T too long.

We first show the results when $F = 2$. In this case, the exact algorithms for TRSP_T and TRSP_P were implemented by using the matching algorithm described in Section 4. For each area, we picked the top 12 facilities from among those prepared in advance and added facilities one by one until $n = 25$. Therefore, we solved 14 instances for each area. The results are shown in **Table 3** and **Fig. 2**, where "h.a." means our "heuristic algorithm," and "k+" stands for the difference in the number of vehicles used in the solutions of our heuristic algorithm and TRSP_P. The total distance when all participants rode alone is also shown in Table 3. Through comparison with this distance, we verified the effect of ride-sharing. From this distance, dist_T decreased to 61.4%, 65.3% and 77.1% on average in Kashiwa, Totsuka, and Sendai, respectively, although dist_P are decreased to 99.8% and increased to 106.6% and 106.7% on average in each area. In a few cases for Kashiwa, dist_P was less than the total distance when all participants rode alone. This would not occur if we made a distance matrix on the basis of the shortest path distance. However, such an inconsistency is caused by the fact that we created the

Table 4 Comparing evaluation values when $F = 3$.

		n	12	13	14	15	16	12	13	14	15	16	12	13	14	15	16
Kashiwa	TRSP_T	k	5	5	6	6	6	5	5	5	6	6	5	5	6	6	6
		dist_T	65.34	66.86	76.07	76.07	78.38	58.15	68.52	73.55	80.98	86.21	66.93	68.76	72.68	74.64	75.66
		dist_P	124.19	130.44	139.65	143.71	154.74	107.45	131.69	151.19	158.62	171.63	120.38	131.58	138.97	144.24	153.10
	TRSP_P	dist_T	66.02	71.34	80.73	80.73	82.51*	59.28	73.82	79.41	90.49	91.71*	70.40	72.23	76.62	77.22	75.66*
		dist_P	123.52	128.78	134.14	138.21	148.58*	105.02	125.78	143.35	148.17	161.76*	120.27	131.47	136.61	143.34	153.10*
	h.a.	k+	0	+1	0	0	+1	0	+1	+2	+1	+2	+1	0	0	+1	0
		dist_T	65.34	71.56	76.38	77.02	83.36	59.28	74.94	82.52	90.23	92.81	70.14	71.19	79.27	83.99	84.22
		dist_P	123.52	124.26	135.78	140.48	145.30	105.02	123.99	131.56	142.53	153.14	114.64	136.43	137.06	139.70	159.16
	total distance by riding alone		119.90	123.12	132.33	136.39	143.20	101.38	120.04	131.49	144.40	155.19	116.96	124.49	133.71	137.77	144.59
Totsuka	TRSP_T	k	5	5	5	5	6	4	5	6	6	6	4	5	6	6	6
		dist_T	84.07	85.79	90.19	94.81	98.71	77.86	84.34	94.41	99.59	100.42	74.48	89.11	95.34	98.02	98.97
		dist_P	169.85	177.56	196.91	219.84	221.04	158.79	172.10	181.78	198.50	210.26	160.45	175.08	181.31	190.22	205.56
	TRSP_P	dist_T	88.35	90.84	94.24	100.13*	107.61*	77.86	89.86	99.93	107.72	108.56*	74.48	96.70	101.51	102.39	103.33*
		dist_P	160.12	169.40	190.69	209.31*	208.59*	158.79	171.09	180.77	196.34	208.10*	160.45	172.55	173.01	182.80	198.14*
	h.a.	k+	+1	+1	+2	+2	+1	+2	0	0	+1	0	+1	+1	−1	+1	+1
		dist_T	97.47	94.92	106.67	107.51	108.01	85.73	88.43	94.41	103.30	100.42	81.59	96.22	111.05	105.52	112.30
		dist_P	154.28	164.41	177.73	188.66	195.54	143.91	172.82	181.78	190.76	210.26	149.73	164.37	203.17	182.06	200.63
	total distance by riding alone		138.95	143.24	157.35	167.44	173.31	134.38	148.98	163.61	175.55	185.64	133.78	148.42	154.65	161.76	176.16
Sendai	TRSP_T	k	6	6	6	6	6	5	5	5	6	7	5	5	6	6	7
		dist_T	54.28	58.60	60.48	61.67	66.86	54.17	62.77	63.96	67.72	68.09	56.40	59.03	61.01	67.40	68.76
		dist_P	95.81	104.61	109.37	119.52	134.86	99.37	110.35	122.52	123.44	118.06	93.578	108.80	110.27	119.89	122.78
	TRSP_P	dist_T	57.20	61.52	62.58	63.39	68.58*	57.06	64.73	65.93	73.66	69.88	56.40	59.94	66.40	71.87	75.53
		dist_P	88.12	96.92	101.02	111.51	126.85*	91.64	109.20	121.37	121.80	110.34	93.58	104.72	102.04	115.74	113.08
	h.a.	k+	−1	0	0	+2	+1	+1	+1	+1	+1	0	+1	+1	+1	+1	0
		dist_T	60.17	64.92	65.74	67.21	77.16	61.43	74.63	74.41	77.39	75.76	64.20	67.73	68.55	74.02	73.41
		dist_P	98.63	105.66	108.63	112.04	131.46	87.72	107.49	118.35	117.55	115.29	93.83	104.98	103.07	114.26	116.85
	total distance by riding alone		81.61	87.32	89.70	99.88	112.30	78.99	87.94	98.12	103.79	104.63	81.43	89.16	91.54	102.16	105.75

distance matrix on the basis of real roads from the Google Map API, which extracts the distance giving the shortest travel time. In Fig. 2, the cold color lines represent dist_P, where the optimal values were obtained from TRSP_P, and the warm color lines represent dist_T, where the optimal values were obtained from TRSP_T. The broken lines are for reference. There were few gaps between the dist_P obtained by our heuristic algorithm and the exact values. In most of the cases, our heuristic algorithm improved dist_P in comparison with those obtained from TRSP_T, which is represented by broken lines. Sometimes, the obtained dist_P was slightly shorter than the exact value since the number of vehicles was one or two more than the exact solution. With respect to dist_P, the relative errors, $(d - d^*)/d^*$ for distance d obtained by our heuristic algorithm and the exact value d^*, were less than 15% for all instances and were 2% on average. Regarding to dist_T, the gaps between distances obtained by our algorithm and the exact values were not that small. The average of the relative errors was about 10%. In particular, for instances in Kashiwa, the average was about 13%. However, the gaps of distances between our heuristic algorithm and those obtained from the solutions of TRSP_P were not that large. The relative errors between dist_T obtained by our heuristic algorithm and obtained by TRSP_P were 3% on average. We also recognized that these relative errors did not depend on n. Thus, when $F = 2$, our heuristic algorithm provided good approximate solutions for TRSP_P although the number of vehicles exceeded the number for vehicles for TRSP_P sometimes.

For the case of $F = 3$, we found exact optimal solutions for TRSP_T and TRSP_P by solving MILP formulations with a solver. Even if we added valid inequalities as described in Section 5.1, the computational time became drastically long when the number of participants n was over 15. Thus, we examined instances with a small size only, i.e., the range of n was set up from 12 to 16. The first set of instances was the same as that used for the case of $F = 2$. In addition, we used two other sets of instances that were made by choosing 12 facilities randomly from those prepared in advance and by adding facilities one by one.

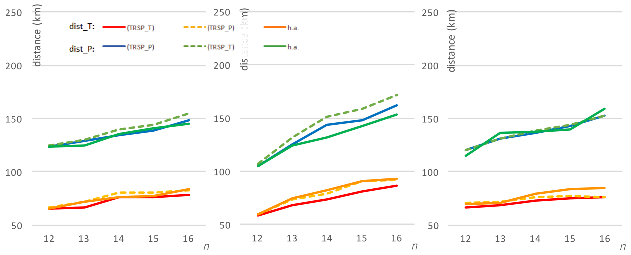


Fig. 3 Comparing dist_T and dist_P for $F = 3$ for Kashiwa.

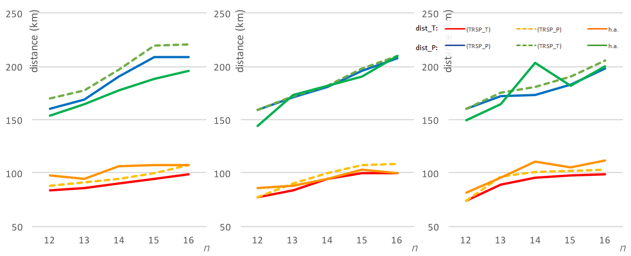


Fig. 4 Comparing dist_T and dist_P for $F = 3$ for Totsuka.

Table 4 and **Fig. 3**, **Fig. 4**, and **Fig. 5** indicate the results. Here, the computational time was limited by 3,600 seconds. In Table 4, the mark "*" indicates the best value among feasible solutions provided before the solver stopped due to a time limit, and it does not certify optimality. The tendency of the results was not that different from the case of $F = 2$. With respect to dist_P, the relative errors were less than 18% for all instances, and the average relative errors for Kashiwa, Totsuka, and Sendai were about −1.4%, −2.1% and 1.9%, respectively. When our heuristic algorithm used vehicles more than TRSP_P, dist_P obtained by

IPSJ Transactions on Mathematical Modeling and Its Applications Vol.12 No.2 1–11 (July 2019)

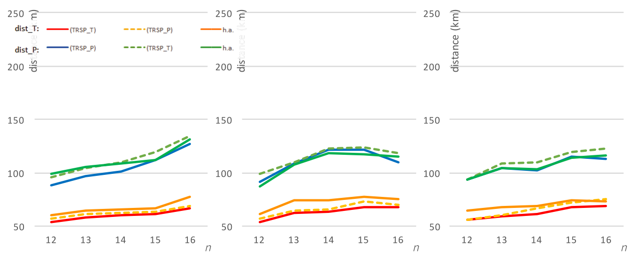


Fig. 5 Comparing dist_T and dist_P for $F = 3$ for Sendai.

Table 5 Results when the number of vehicles was increased.

		$n = 15$	$n = 16$
TRSP_P	dist_T	100.13*	107.61*
	dist_P	209.31*	208.59*
TRSP_P with $k = 7$	dist_T	109.17	113.97*
	dist_P	184.77	193.11*
h.a.	dist_T	107.51	108.01
	dist_P	188.66	195.54

the algorithm was sometimes shorter than that for TRSP_P. In this situation, the relative errors were less than zero. In the instances for Totsuka shown on the left of Fig. 4, dist_P obtained by our heuristic algorithm was comparatively shorter than the values for TRSP_P, especially when $n = 15$ and 16. This was not only caused by the number of vehicles but also because the dist_P of TRSP_P were not optimal solutions. For the instances with $n = 15$ and 16, we resolved TRSP_P for $k = 7$, which was the number obtained by the heuristic algorithm. The result is shown in **Table 5**. The dist_P of the heuristic algorithm and that of TRSP_P were almost the same. From this, our heuristic algorithm could find a good approximate solution even when k was the same as that of TRSP_P's. In comparison, in the right graph for Totsuka when $n = 14$, the gap of dist_P was large, where the relative error was about 17%. This is because the heuristic algorithm used less vehicles than TRSP_P. We now turn to comparing dist_T. Although the average relative error was 9.5%, the relative errors between our heuristic algorithm and obtained from solutions for TRSP_P were about −.5% on average. Hence, we can see that our heuristic algorithm finds solutions close to the optimal solution to TRSP_P. From the results for all of the areas, the same as the results of $F = 2$, we recognized that the relative errors did not depend on n. To summarize the results of the experiments for both cases of $F = 2$ and 3, we conclude that our heuristic algorithm can find good approximate solutions for TRSP_P and can be applied to instances with a larger n.

We next discuss the computational time. **Table 6** shows the computational time of TRSP_P and the heuristic algorithm for the instances of $F = 3$. Note that the computational time of the heuristic algorithm means the time that was taken to perform Algorithm 2 for each problem 15 times. As we mentioned before, TRSP_P needed to take a lot time to solve the problem when n was over 15. TRSP_P could not obtain optimal solutions within the time limit of 3,600 seconds in most cases when $n = 16$. We recognize that whether TRSP_P can be solved within a reasonable amount of time depends on the number of participants n and their location. However, the heuristic algorithm could solve the problems within about 10 seconds even when $n = 16$ for all of the areas. Although the heuristic algorithm took more time as n increased, the rate of increase rate was not that much. From these results, we conclude that our heuristic algorithm should be applied to instances with a larger n in a reasonable amount of computational time.

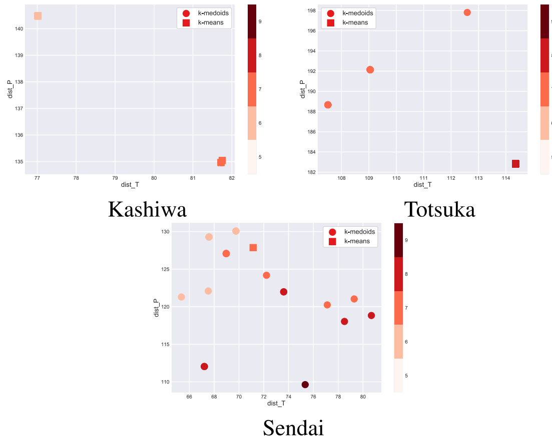


Fig. 6 Solutions obtained with Algorithm 1 employing k-medoids and k-means methods.

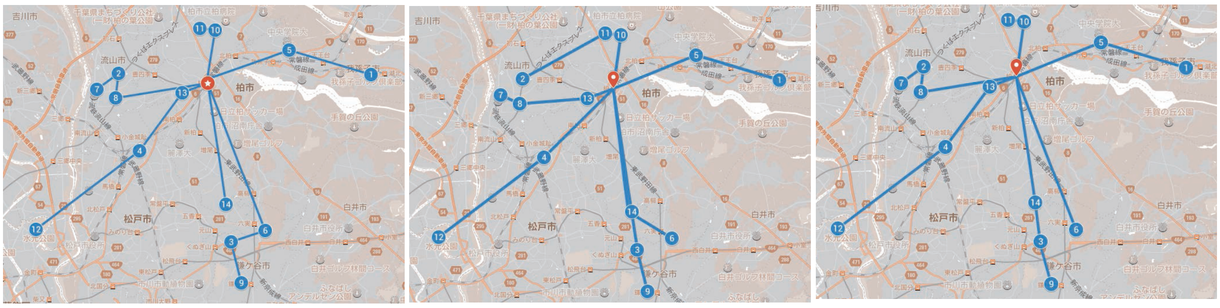
We finally discuss the appropriateness of the k-medoids method that is employed in Algorithm 1. It is said that this method is more robust to outliers because it uses a dissimilarity matrix and not the Euclidean distance of data. The partition, however, depends on the initial medoids used instead of the centroids that are used in the k-means method. **Figure 6** shows the results obtained from 30 different initial medoids for an instance with $n = 25$ and $F = 3$ for each area, where the horizontal and vertical axes stand for dist_P and dist_T, respectively, and the color of each plot denotes the necessary number of vehicles in the corresponding solution. In Fig. 6, we also display the results that were obtained by using Algorithm 2, which adopts the k-means method, where we performed a principal component analysis in order to reduce the dimensions of the similarity matrix S, before the k-means method was applied. For Kashiwa, the results obtained with the k-medoids method overlapped with those obtained with the k-means method. Since we can see that the k-medoids method depends on the initial medoids, our experimental results show that the best solution among the obtained solutions was achieved by performing the algorithm 15 times. By performing the algorithm several times, we expect to obtain a desirable solution balanced between dist_P and dist_T that is strongly associated with the number of vehicles.

5.3 Validity of Tours Minimizing Total Trip Distance of Participants

We now examine the validity of the tours obtained by TRSP_P and the heuristic algorithm by observing cases of $F = 3$. The three maps in **Fig. 7** display the tours determined by TRSP_T, TRSP_P, and our heuristic algorithm, respectively, for the instance with $n = 14$ in the first dataset for Kashiwa. On these maps, we indicated the locations on Google Maps but did not draw the routes along real roads. **Figure 8** and **Fig. 9** also show tours for Totsuka and Sendai, respectively. In all of the areas, almost all of the tours determined by TRSP_P and by the heuristic algorithm were better than that determined by TRSP_T in terms

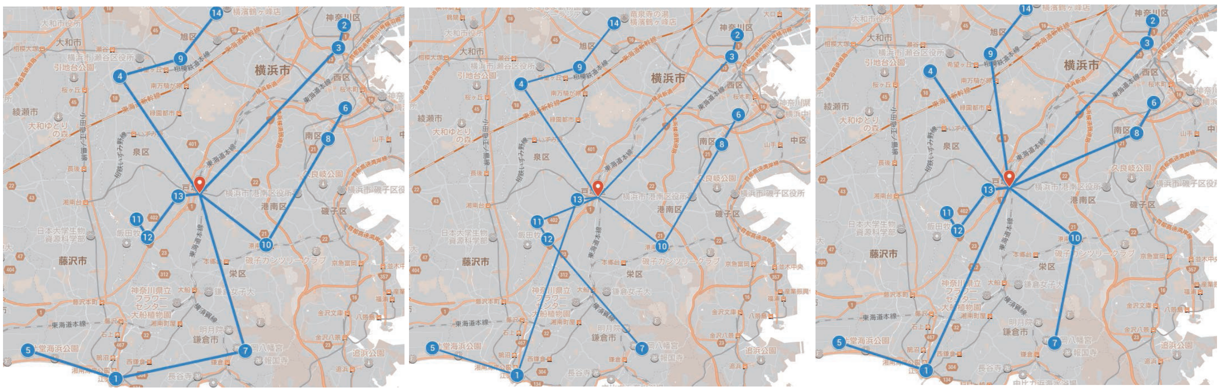
Table 6 Comparing computational time when $F = 3$.

	n	12	13	14	15	16	12	13	14	15	16	12	13	14	15	16
Kashiwa	TRSP_P	2.34	15.60	86.83	261.75	3,600.00	2.48	6.82	82.67	246.65	3,600.00	1.69	2.622	50.385	401.879	3,600.00
	h.a.	4.53	5.29	6.24	6.74	6.97	5.10	5.80	7.37	7.50	8.51	5.12	5.02	6.27	6.84	7.25
Totsuka	TRSP_P	7.72	90.62	1,682.02	3,600.00	3,600.00	4.34	6.04	74.84	1,998.96	3,600.00	28.06	19.56	132.31	877.83	3,600.00
	h.a.	8.23	8.16	8.09	8.97	10.19	5.95	6.04	7.32	7.57	7.33	5.91	7.86	9.70	9.66	9.37
Sendai	TRSP_P	1.96	20.39	53.43	634.11	3,600.00	1.46	13.57	125.71	99.46	34.21	2.84	17.63	58.25	406.37	232.37
	h.a.	6.81	8.41	8.67	10.18	9.66	6.12	6.73	6.91	7.48	9.16	9.52	8.75	8.93	10.06	10.17



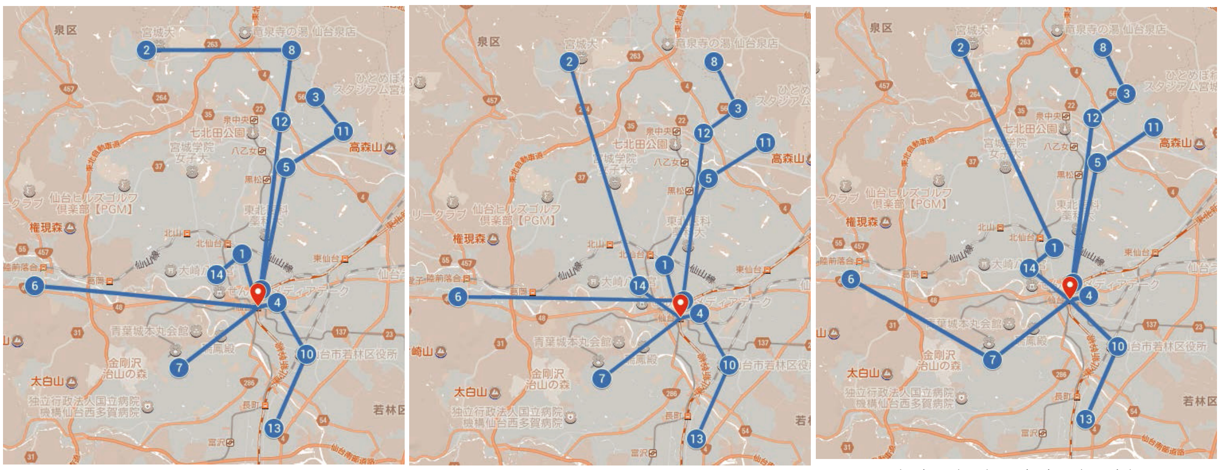
solution for TRSP_T   solution for TRSP_P   solution by heuristic algorithm

Fig. 7 Tours sketched out for Kashiwa.



solution for TRSP_T   solution for TRSP_P   solution by heuristic algorithm

Fig. 8 Tours sketched out for Totsuka.



solution for TRSP_T   solution for TRSP_P   solution by heuristic algorithm

Fig. 9 Tours sketched out for Sendai.

of the tours going straight to the destination without much detouring. However, for Kashiwa, route p_2-p_{11}-p_{10}-p_g, which was determined by TRSP_P was not good for passenger 2. This was because k might not be suitable for solving TRSP_P. As the number of k affects solutions, it is important to decide the appropriate k.

It is unfavorable that the fare of each participant increases when other participants join ride sharing. Therefore, we determined how the fares change. On the basis of the tours obtained by our heuristic algorithm, we checked how the fares change by increasing n in a range from 12 to 25 for the first 12 participants in the 3 datasets that were used for the experiments with $F = 3$. The

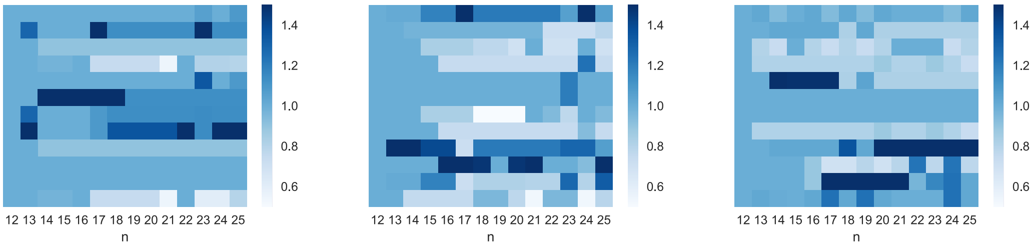


Fig. 10 Shift in paid fares for first 12 participants in Kashiwa.

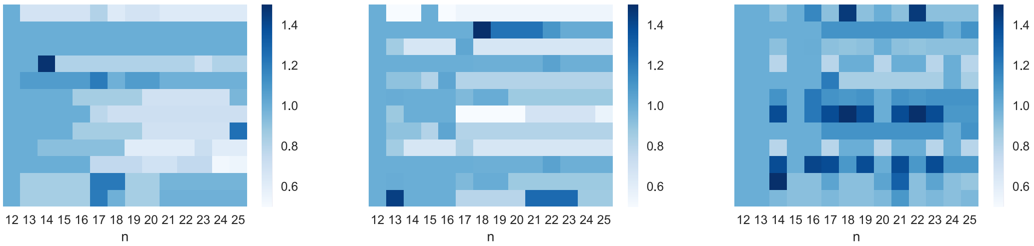


Fig. 11 Shift in paid fares for first 12 participants in Totsuka.

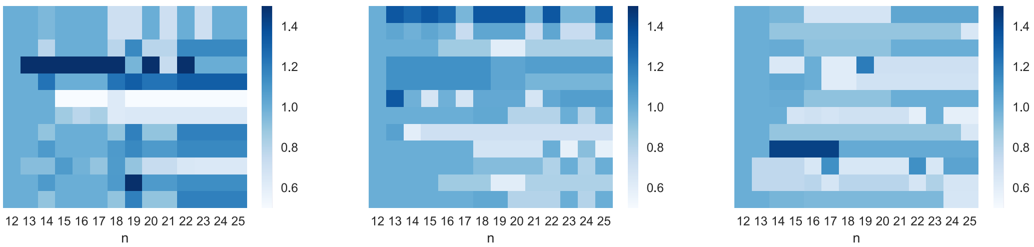


Fig. 12 Shift in paid fares for first 12 participants in Sendai.

fares of participants were determined by using two methods for distributing fares; one was a proportional distribution method that distributes fares for each participant on the basis of the ratio of the fares that they pay when they ride alone, and the other is a DEA game [21]. Because the tendency in the fare distribution of each method was almost the same, we show the results for the proportional distribution method. The results are shown in heat maps in **Fig. 10**, **Fig. 11**, and **Fig. 12**, where each row corresponds to a participant included in the first dataset with $n = 12$ and each column stands for n. They display the increase in the ratio of the fare of each participant by normalizing the value of each paid fare to 1 when $n = 12$. The range of the indicator was set from 0.5 to 1.5 because almost all of the increasing ratios were contained in this range. The fare of each participant seemed to be reduced or changed little when n increased. Thus, in terms of fares, the obtained tours seem to be reasonable. However, the fares of some of the participants increased greatly. Such a participant would be changed to riding alone from sharing with someones after a new participant joined ride-sharing. If the participants are dissatisfied with a large increase in fare due to there being additional members joining in on ride-sharing, another way to distribute fare is needed.

6. Conclusion

We dealt with the taxi ride-sharing problem for event trips and introduced an objective function that minimizes the total trip distance of all passengers for the problem. We formulated this problem by MILP. However, the computational time was long when the number of passengers was large. Thus, we proposed exact algorithms for the problem under the restriction that the ride capacity was 2 and developed a heuristic algorithm that could find good approximate solutions for TRSP_P. We verified that our heuristic algorithm found appropriate solutions for the ride-sharing problem in numerical experiments. In addition, we also discussed solutions obtained by using our heuristic algorithm in terms of tours and fare distribution.

As we described in Sections 5.2 and 5.3, the number of vehicles k affects the solutions of TRSP_P. Thus, it is an important future work to determine the appropriate k. It seems to be also valuable to incorporate fares into the model of TRSP_P.

Acknowledgments This research is supported in part by a MEXT Grant-in-Aid for Scientific Research (B) No. 16H03118.

References

[1] Agatz, N., Erera, A., Savelsbergh, M. and Wang, X.: Optimization for dynamic ride-sharing: A review, *European Journal of Operational Research*, Vol.223, No.2, pp.295–303 (2012).

[2] Agatz, N.A., Erera, A.L., Savelsbergh, M.W. and Wang, X.: Dynamic ride-sharing: A simulation study in metro Atlanta, *Transportation Research Part B: Methodological*, Vol.45, No.9, pp.1450–1464 (2011).

[3] Asghari, M., Deng, D., Shahabi, C., Demiryurek, U. and Li, Y.: Price-aware real-time ride-sharing at scale: An auction-based approach, *Proc. 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pp.1–10 (2016).

[4] Ben-Smida, H.E., Krichen, S., Chicano, F. and Alba, E.: Mixed Integer Linear Programming Formulation for the Taxi Sharing Problem, *Smart Cities: First International Conference, Smart-CT 2016, Málaga, Spain, June 15-17, 2016, Proceedings*, Alba, E., Chicano, F. and Luque, G. (Eds.), Springer International Publishing, Cham, pp.106–117 (2016).

[5] Bicocchi, N. and Mamei, M.: Investigating ride sharing opportunities through mobility data analysis, *Pervasive and Mobile Computing*, Vol.14, No.Supplement C, pp.83–94 (2014).

[6] Chan, N.D. and Shaheen, S.A.: Ridesharing in North America: Past,

Present, and Future, *Transport Reviews*, Vol.32, No.1, pp.93–112 (2012).
[7] Fu, X., Huang, J., Lu, H., Xu, J. and Li, Y.: Top-k Taxi Recommendation in Realtime Social-Aware Ridesharing Services, *Advances in Spatial and Temporal Databases: 15th International Symposium, SSTD 2017, Arlington, VA, USA, August 21–23, 2017, Proceedings*, Gertz, M., Renz, M., Zhou, X., Hoel, E., Ku, W.-S., Voisard, A., Zhang, C., Chen, H., Tang, L., Huang, Y., Lu, C.-T. and Ravada, S. (Eds.), Springer International Publishing, Cham, pp.221–241 (2017).
[8] Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X. and Koenig, S.: Ridesharing: The state-of-the-art and future directions, *Transportation Research Part B: Methodological*, Vol.57, No.Supplement C, pp.28–46 (2013).
[9] Gillett, B.E. and Miller, L.R.: A Heuristic Algorithm for the Vehicle-Dispatch Problem, *Operations Research*, Vol.22, No.2, pp.340–349 (1974).
[10] Hosni, H., Naoum-Sawaya, J. and Artail, H.: The shared-taxi problem: Formulation and solution methods, *Transportation Research Part B: Methodological*, Vol.70, No.Supplement C, pp.303–318 (2014).
[11] Huang, Y., Bastani, F., Jin, R. and Wang, X.S.: Large Scale Real-time Ridesharing with Service Guarantee on Road Networks, *Proc. VLDB Endow.*, Vol.7, No.14, pp.2017–2028 (2014).
[12] Lee, A. and Savelsbergh, M.: Dynamic ridesharing: Is there a role for dedicated drivers?, *Transportation Research Part B: Methodological*, Vol.81, No.Part 2, pp.483–497 (2015).
[13] Li, B., Krushinsky, D., Reijers, H.A. and Woensel, T.V.: The Share-a-Ride Problem: People and parcels sharing taxis, *European Journal of Operational Research*, Vol.238, No.1, pp.31–40 (2014).
[14] Li, X. and Quadrifoglio, L.: Feeder Transit Services: Choosing between Fixed and Demand Responsive Policy, *Transportation Research Part C: Emerging Technologies*, Vol.18, No.5, pp.770–780 (2010).
[15] Lin, J., Sasidharan, S., Ma, S. and Wolfson, O.: A Model of Multimodal Ridesharing and Its Analysis, *2016 17th IEEE International Conference on Mobile Data Management* (*MDM*), Vol.1, pp.164–173 (2016).
[16] Lin, Y., Li, W., Qiu, F. and Xu, H.: Research on Optimization of Vehicle Routing Problem for Ride-sharing Taxi, *Procedia - Social and Behavioral Sciences*, Vol.43, No.Supplement C, pp.494–502 (2012).
[17] Ma, S., Zheng, Y. and Wolfson, O.: T-share: A large-scale dynamic taxi ridesharing service, *2013 IEEE 29th International Conference on Data Engineering* (*ICDE*), pp.410–421 (2013).
[18] Ma, S. and Wolfson, O.: Analysis and Evaluation of the Slugging Form of Ridesharing, *Proc. 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, *SIGSPATIAL '13*, pp.64–73, ACM (2013).
[19] Ma, X., Ding, J., Wang, W., Hua, X. and Peng, Y.: A taxipooling system with equity consideration, *14th International Conference on Computers in Urban Planning and Urban Management*, pp.1–18 (2015).
[20] Massobrio, R., Fagúndez, G. and Nesmachnow, S.: A parallel micro evolutionary algorithm for taxi sharing optimization, *VIII ALIO/EURO Workshop on Applied Combinatorial Optimization* (2014).
[21] Minami, S., Horikawa, K., Sato, K., Watanabe, E., Yoshida, T., Yano, M. and Shigeno, M.: Designing for charge distribution of ride-share services by customers being bound for the same event, *Proc. 6th Annual Meeting on Serviceology*, PO-17 (2018) (in Japanese).
[22] Molenbruch, Y., Braekers, K. and Caris, A.: Typology and literature review for dial-a-ride problems, *Annals of Operations Research*, Vol.259, No.1, pp.295–325 (2017).
[23] Qian, X., Zhang, W., Ukkusuri, S.V. and Yang, C.: Optimal assignment and incentive design in the taxi group ride problem, *Transportation Research Part B: Methodological*, Vol.103, No.Supplement C, pp.208–226 (2017).
[24] Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S.H. and Ratti, C.: Quantifying the benefits of vehicle pooling with shareability networks, *Proc. National Academy of Sciences*, Vol.111, No.37, pp.13290–13294 (2014).
[25] Santos, D.O. and Xavier, E.C.: Taxi and Ride Sharing: A Dynamic Dial-a-Ride Problem with Money as an Incentive, *Expert Systems with Applications*, Vol.42, No.19, pp.6728–6737 (2015).
[26] Schreieck, M., Safetli, H., Siddiqui, S.A. and Pfl, C.: A Matching Algorithm for Dynamic Ridesharing, *Transportation Research Procedia*, Vol.19, pp.272–285 (2016).
[27] Stiglic, M., Agatz, N., Savelsbergh, M. and Gradisar, M.: Enhancing urban mobility: Integrating ride-sharing and public transit, *Computers & Operations Research*, Vol.90, No.Supplement C, pp.12–21 (2018).
[28] Tao, C.C. and Chen, C.Y.: Heuristic Algorithms for the Dynamic Taxipooling Problem Based on Intelligent Transportation System Technologies, *4th International Conference on Fuzzy Systems and Knowledge Discovery* (*FSKD 2007*), Vol.3, pp.590–595 (2007).

Taketo Yoshida is a graduate student in University of Tsukuba. He received his Bachelor's degree in Policy and Planning Sciences from University of Tsukuba. His research interest is service science and optimization modeling.

Masaki Yano received his Bachelor's degree in Policy and Planning Sciences from University of Tsukuba in 2018. He received the award of the president of University of Tsukuba. His research interest is deep learning and optimization algorithms.

Kenichiro Horikawa is currently engages in the research on generic technology development for solution of social problems with DENSO CORPORATION.

Keita Sato is currently engages in the research on generic technology development for solution of social problems with DENSO CORPORATION.

Shota Minami received his Master's degree in Service Engineering from University of Tsukuba in 2019. His research interest is game theory, data envelopment analysis, and mobility services.

Maiko Shigeno received her Ph.D. in science from Tokyo Institute of Technology in 1996. She is currently a professor at University of Tsukuba. Her specialized research areas are operations research and mathematical optimization.

