ABSTRACT Trip planning services have been developed along with tourism promotion and information technology evolutions, where we must construct trip routes that simultaneously optimize multi-objective functions such as trip expenses and user satisfaction. Moreover, utilization of past-trip records is essential, because similarities to past-trip records well reflect users’ general preferences and tendencies during trip planning. In this paper, we propose a multi-objective trip planning method using ant colony optimization (ACO). By effectively using the pheromones in ACO, we can construct trip routes similar to trip records stored before and the constructed route can reflect users’ general preferences. In addition, we vary ants’ behaviors in ACO corresponding to various objective functions and hence we can obtain multi-objective trip routes naturally. Experimental results demonstrated that our method outperforms the baseline methods in terms of point-of-interest (POI) satisfaction, POI cost, and past-trip similarity. We also conducted a user study, which clearly indicates that our method obtains high scores through various user questionnaires.

INDEX TERMS Multi-objective trip planning problem, ant colony optimization, past-trip records, pheromone updating, point-of-interest (POI).

I. INTRODUCTION
A. TRIP PLANNING PROBLEM
With the promotion of tourism and the evolution of information technologies, trip planning services have been rapidly developed in the past decades [1], [2].

Trip planning studies nowadays aim at constructing routes that optimize almost any user’s requirements such as point-of-interest (POI) satisfaction and POI cost, where POI refers to a point location that a user wants to visit or he/she considers useful. For example, Bao et al. [3] considered POIs’ features including satisfaction, expense, and duration time. On the other hand, besides users’ preferences and trip time, Isoda et al. [4] additionally evaluated a POI through its static features such as geographical positions and business hours. These studies suggest that a POI has multiple features and it is essential to consider all these features as many as possible. In this paper, we focus on maximizing the total POI satisfaction and minimizing the total POI cost, and we call this the first condition of the trip planning problem.

Meanwhile, due to the wide variety of user’s behavioral principles, it is impossible to perfectly satisfy users’ requirements by only considering the first condition. For example, it is difficult to quantify a user’s behavior like a user prefers to visit a relaxing indoor POI after high-intensity outdoor activities. Also, even for quantifiable objective functions such as POI genres and duration time, if the amount of the objective functions becomes large, it is still hard to satisfy all the objective functions simultaneously. One effective way to solve this issue is utilizing past-trip records [5].

Most studies only calculate POI values based on the number of times that the POI has been visited in the past-trip records [4], [6], [7], [8]. Thus, they ignore the relationships and visiting orders between POIs. We have to take into account the past-trip records including POIs themselves and their visiting orders. Some studies verified the positive effect of utilizing past-trip records. Kasahara et al. [5] modeled the users’ behaviors from their past-trip records to predict
the next visiting POI. Compared with the method without considering past-trip records, the accuracy of the method [5] increased by 10.4% and the routes constructed by the method [5] showed higher similarities to past-trip records. Lu et al. [9] solved the trip planning problem by adding scores to a potential route using past-trip records as references. Similarly, the routes constructed by [9] obtained higher user satisfaction and higher diversity (more POI characteristics are included). Thus, these studies’ results suggest that the utilization of past-trip records is helpful to better reflect users’ behavior principles in trip planning. The past-trip records well represent the users’ behavior. By taking this idea into account, not only the POI satisfaction and the POI cost according to the first condition but also the satisfaction of the entire route (e.g., visiting an indoor POI after an outdoor POI) will be taken into account. We call the utilization of the past-trip records the second condition of the trip planning problem.

B. RELATED WORKS

Generally, the trip planning problem is based on the orienteering problem (OP) [10]. Given a POI candidate set and a time limitation, the OP aims at exploring the optimal route between the starting location and the ending location. Souffriau et al. [11] presented the trip design problem from the OP and developed a method that constructs an optimal route by applying modified guided local search [12]. However, the method is unsatisfied with the first condition, since it only takes one objective function. At the same time, it is insufficient to meet the second condition since there is no use of the past-trip records. As is the case of Souffriau’s method, most studies of OPs [13], [14], [15], [16] only considered a single objective function which usually is the total satisfaction of POIs, and it violates the first condition as a single objective function cannot fully reflect users’ requirements.

On the other hand, the multi-objective orienteering problem (MOOP) [17] assumes that each POI has several features and all these features should be considered in route construction. The MOOP is sufficient to fulfill the first condition as it takes into account multi-objective functions. For the route construction part in [17], Pareto ant colony optimization (P-ACO) and Pareto variable neighborhood search (P-VNS) were proposed based on ant colony optimization [18] and variable neighborhood search [19], respectively, to realize the multi-objective optimization. Martin et al. [20] presented the multi-objective artificial bee colony algorithm (MOABC) which is an optimization method inspired by the foraging behavior of honey bees to solve the MOOP. Although Schilde’s and Martin’s methods are sufficient to fulfill the first condition, there are no use of the past-trip records so that they cannot satisfy the second condition.

Bao et al. [3] and Isoda et al. [4] picked up in Section I-A fulfilled the first condition by simultaneously optimizing multiple objectives, but they were insufficient for fulfilling the second condition because there was no direct use of past-trip records in planning. Also, Kasahara et al. [5] and Lu et al. [9] fulfilled the second condition, but they did not satisfy the first condition due to the lack of consideration of multi-objective functions such as minimizing POI costs and maximizing POI satisfaction.

To our best knowledge, there is no method that sufficiently satisfies both the first and second conditions.

C. PROPOSAL AND CONTRIBUTION

To achieve the first condition, well-known multi-objective optimization techniques such as genetic algorithm [21], simulated annealing [22], local search [17], and memetic algorithm [23] can be used. However, because these techniques are based on updating the solutions by applying the operations such as substitutions and insertions on some of the nodes in the initial or intermediate solutions, it is difficult to maintain information on the relationships and the visiting order of the past-trip records in the process of updating.

Now, we discuss utilizing ACO to tackle the first and second conditions. ACO has several ants that construct routes corporately, and it is promising to accomplish the multi-objective optimization by modifying the activity of each ant to handle each objective function. In addition, each ant follows pheromones left by the previous ants, and each ant constructs a route sequentially. Likewise, we consider past users as the previous ants, and their past-trip records as the past pheromones. As pheromones can be added to any piece of complete routes, we can construct a route reflecting past users’ behaviors very effectively. As a result, ACO is expected to fulfill the first and second conditions and construct routes that optimize the multi-objective functions.

In this paper, we propose a multi-objective trip planning method based on ACO utilizing past-trip records. The proposed method optimizes multi-objective functions through modifying ants’ behaviors (the first condition) and reflects the past-trip records by adding pheromones to a path between POIs (the second condition).

We aim to answer the following two research questions (RQs):

RQ1- Does the proposed method outperform the existing trip planning method?

RQ2- Do the objective functions in the proposed method work effectively to meet users’ requests?

To answer these research questions, we conducted experimental evaluations using three real-world datasets. We compared the proposed method to the existing methods [17], [24]. In [24], a multi-objective ant colony algorithm based on the pheromone weight (MOACOPW) is proposed, which is one of the latest multi-objective ACO methods.

The evaluation results demonstrate that our method obtained high user satisfaction over routes and generated more stable and diverse solutions. Thus, we can say YES to the above two research questions.
The contributions of this paper are summarized as follows.  
1) ACO is applied to the MOOP and it constructs routes that fulfill user requirements by varying ant behaviors corresponding to different objective functions.
2) In the route construction process, the proposed method adjusts pheromones based on past-trip records and hence the proposed method can well reflect the past-trip records.
3) We compared the proposed method with the baseline methods by applying them to the datasets that have 150 POIs in Kyoto, Japan, 100 POIs in Sapporo, Japan, and 150 POIs in Tokyo, Japan. The results show that the proposed method outperforms the baseline methods in terms of POI satisfaction, POI cost, and past-trip similarity.

The rest of this paper is organized as follows: Section II defines the multi-objective trip planning problem (MOTPP); Section III proposes an optimizing method of the MOTPP by using ACO; Section IV demonstrates the experimental results compared to the baseline methods; and Section V gives a conclusion and future works.

II. DEFINITION OF MULTI-OBJECTIVE TRIP PLANNING PROBLEM (MOTPP)

In this section, we define the MOTPP. Consider a complete graph \( G = (V, E) \) constructed from a target area where \( V \) is a node set that includes all POIs, a starting location and an ending location, and \( E \) is an edge set that contains edges between any two nodes \( v_i, v_j \in V \). \( G \) is called an area graph. Starting location and ending location are denoted as \( v_{st} \) and \( v_{dst} \in V \). The other nodes are called POI nodes and the set of all POI nodes is denoted as \( V_p \). POI satisfaction \( s_i \), duration time \( d_i \), and POI cost \( c_i \) are given to each POI node \( v_i \in V_p \). Travel time \( t_{ij} \) is given to each edge \( (v_i, v_j) \in E \). We set the past-trip records as \( Tour^{post} = [v_1, \cdots, v_k] \) which represents the visiting orders from \( v_1 \in V \) to \( v_k \in V \).

Based on the above settings, the MOTPP is defined as below. Table 1 lists 14 important notations in this paper.

**Definition 1:** Given an area graph \( G \), a time limitation \( T \), and \( l \) past-trip records, the problem aims to find multiple routes that begin from \( v_{st} \) to \( v_{dst} \) optimizing the objective functions (a) to (d) below.

(a) Maximize total POI satisfaction. Total POI satisfaction \( S^{Tour} \) is given by adding up the satisfaction of each POI node visited in the constructed route \( Tour \), i.e.,

\[
\text{Maximize } S^{Tour} = \sum_{v_i \in Tour} s_i \tag{1}
\]

(b) Minimize total POI cost. Total POI cost \( C^{Tour} \) is given by adding up the cost of each node visited in the constructed route \( Tour \), i.e.,

\[
\text{Minimize } C^{Tour} = \sum_{v_i \in Tour} c_i \tag{2}
\]

(c) Maximize similarity with past-trip records. The similarity \( Sim^{Tour} \) between past-trip records and a constructed route \( Tour \) is computed in Eq. (8) which will be defined in Section III-E3, i.e.,

\[
\text{Maximize } Sim^{Tour} \tag{3}
\]

(d) All of (a) to (c). The objective function (d) is expressed by:

\[
\text{Minimize } wv = \sqrt{(1 - (\bar{S}))^2 + (\bar{C})^2 + (1 - (\bar{Sim}))^2} \tag{4}
\]

where \( \bar{S} \), \( \bar{C} \), and \( \bar{Sim} \) are normalized values with a range of \([0, 1]\) representing total POI satisfaction, total POI cost, and similarity with past-trip records, respectively.

In Definition 1, we set the objective function (d) to obtain a route so that all the objectives (a)–(c) on that route are well balanced for users.

Also, a total travel time given by the sum of POI durations and travel times between nodes included in the constructed route must not exceed the time limitation \( T \).

Note that the MOTPP is a discrete optimization problem, because it deals with discrete variables related to POIs. Thus, this problem belongs to the discrete optimization problems.

**Example 1:** Fig. 1, Fig. 2, and Table 1 show the example of MOTPP. Fig. 1 shows a set of POI nodes and its location. In this figure, we have 10 POI nodes, \( v_a \) to \( v_{j} \). We also have a start location \( v_{st} \) and an end location \( v_{dst} \). A past user visits \( v_d, v_e, \) and \( v_{dst} \) in this order. Table 1 summarizes POI information.

Fig. 2 shows the routes constructed. In this case, we have four routes, a red route, a green route, a blue route, and a yellow route, which correspond to the objective functions (a) to (d), respectively. Route (c) traverses the past user’s route.

III. MULTI-OBJECTIVE TRIP PLANNING BASED ON ACO UTILIZING TRIP RECORDS

In this section, we propose an ACO-based method for solving the MOTPP.

A. STRATEGY

We first introduce the two main strategies of ACO for solving the MOTPP as follows.

| POI node | POI satisfaction | POI cost | Duration time [h] |
|----------|-----------------|----------|------------------|
| \( v_a \) | 1.0             | 2.0      | 1.0              |
| \( v_b \) | 3.0             | 1.0      | 1.0              |
| \( v_c \) | 3.0             | 0.5      | 1.0              |
| \( v_d \) | 4.0             | 1.0      | 1.0              |
| \( v_e \) | 3.0             | 0.5      | 1.0              |
| \( v_f \) | 5.0             | 1.0      | 1.0              |
| \( v_g \) | 4.0             | 1.5      | 1.0              |
| \( v_h \) | 4.0             | 0.5      | 1.0              |
| \( v_i \) | 1.0             | 1.0      | 1.0              |
| \( v_j \) | 2.0             | 2.0      | 1.0              |
For the implementation of (B), it is difficult to find a solution that optimizes all objective functions completely because of the trade-off relationship [25], [26] of several objective functions. To settle the issue, one possible way is to construct Pareto optimal solutions [27]. Also, it is required to find Pareto optimal solutions that consider all objective functions well-balanced without trapping in local solutions. To avoid trapping in local solutions, our method proposes local search and path relinking, and the details of the two procedures are shown in Section III-E1 and Section III-F1, respectively.

B. THE FLOW OF OUR METHOD

Based on the discussion in Section III-A, the flow of our method is shown in Fig. 3 and below.

(Step 1) Initialization
Initialize the pheromones to the edges in the area graph based on the past-trip records.

(Step 2) Route Construction
Set an objective function to each ant. Each ant constructs a route from the starting location \( v_{st} \) to the ending location \( v_{dst} \) by selecting nodes referring to the pheromones and the assigned objective function.

(Step 3) Local Search + Pheromone Updating
Apply local search to a constructed route in Step 2 to locally optimize one of the objective functions (a) to (d). Update the pheromones by local updating and similarity updating. If the number of newly constructed routes is less than \( M \), go to Step 2.

(Step 4) Path Relinking + Pheromone Global Updating
Apply path relinking to add a solution to a set of Pareto optimal solutions. Perform global updating to update the pheromones. Repeatedly perform Step 2 to Step 4 \( N \) times.
E. Saeki et al.: Multi-Objective Trip Planning Based on Ant Colony Optimization Utilizing Trip Records

(Step 5) Route Selection and Construction

Select optimal routes corresponding to the objective functions (a) to (d) and provide them to the user.

We explain Step 1–Step 5 in Section III-C to III-G respectively in detail.

C. INITIALIZATION (STEP 1)

This step initializes the pheromones based on the past-trip records. Let $\text{Tour}_{\text{past}}$ be one of the past-trip records. Let $(v_a, v_b)$ be an edge included in $\text{Tour}_{\text{past}}$, where $v_a, v_b \in V$ are consecutive nodes. Note that, $\text{Tour}_{\text{past}}$ means one of the past-trip records and initial pheromone will be given to the same route based on $\text{Tour}_{\text{past}}$. Then we give the pheromone of $I_{\text{old}}$ to $(v_a, v_b)$. $I_{\text{old}}$ is a parameter that controls the importance of the past-trip records. In our method, $I_{\text{old}}$ is set to be 0.6 based on our preliminary studies where 0.6 turns out to be the best value in the range of (0.0, 1.0). This step enables the ant to construct a route that resembles the past-trip records.

For the example in Fig. 3, with the four nodes named A, B, C, and D, and the past route which follows A→C→B shown in [Step 1], we give the pheromones to the edges in green. Every ant will follow these pheromones in Step 2 and Step 3 and hence we can construct a route similar to past-trip records.

D. ROUTE CONSTRUCTION (STEP 2)

In this step, an ant constructs a route from $v_{st}$ to $v_{dst}$ by moving POI nodes sequentially until the total travel time exceeds the time limitation $T$. Step 2 and Step 3 are repeated $M$ times. Let $\text{Tour}^k$ be the route constructed at $k$-th iteration of Step 2 – Step 3. Let $t_{ij}$ and $d_l$ be the travel time from $v_i \in V$ to $v_j \in V$ and the duration time at $v_l \in V_p$. Then the total travel time, $T_k$, of $\text{Tour}^k$ is given by summing up all the travel times and duration times in $\text{Tour}^k$. Let $s_j$ and $c_j$ be the POI satisfaction and POI cost at $v_j \in V_p$. Likewise, the total POI satisfaction $S_{\text{Tour}^k}$ and total POI cost $C_{\text{Tour}^k}$ of $\text{Tour}^k$ are given by summing up all POI satisfactions and POI costs, respectively.

1) POI SELECTION

Assume that an ant $k$ is currently located at the node $v_j$. Let $J_k(i)$ show a set of unvisited POI nodes. Then we decide the next visiting POI node $v_j \in J_k(i)$ as the next visiting POI node based on the state transition rule below:

1) Ant $k$ will choose $v_j \in J_k(i)$ as the next visiting POI node at a probability of $q_0$, where $v_j$ is the POI node which has the highest transition probability $p_k(i, j)$ among all POI nodes in $J_k(i)$.

2) Ant $k$ will randomly choose $v_j \in J_k(i)$ as the next visiting POI node at a probability of $(1 - q_0)$.

By introducing the probability of $q_0$, the ant $k$ explores various routes that focus on the pheromones and POI features at the same time. $q_0$ is set as 0.4 to have some amount of randomness in route exploration but to follow the state transition rule based on our preliminary studies where 0.4 turns out to be the best value in the range of (0.0, 1.0).
If the total travel time of the route from \( v_j \) to \( v_{dst} \) exceeds the time limitation \( T \) when \( v_j \) is decided as the next POI node, the route immediately ends up with a route from the current node \( v_j \) to \( v_{dst} \) and terminate the route exploration.

2) TRANSITION PROBABILITY

The transition probability \( p_k(i, j) \) in 1) and 2) above is designed as follows. The amount of pheromone at an edge \((v_i, v_j)\) in \( E \) is expressed as \( \tau(i, j) \). It is also set as \( \eta(i, j) = 1/t_{ij} \). The smaller the travel time is, the larger the value of \( \eta(i, j) \) becomes. Moreover, \( \xi(j) \) describes what extent the POI node \( v_j \) costs and \( \xi(j) \) for \( v_j \in V_p \) is defined as:

\[
\xi(j) = 1 - \frac{c_j}{\max_{u \in V_p} c_u}
\]

where \( s_j \) is POI satisfaction of \( v_j \), \( \alpha \) shows the weight of the pheromone on the edge \((i, j)\), \( \beta \) shows the weight of the time between the nodes \( v_i \) and \( v_j \), \( \gamma \) shows the weight of the POI satisfaction on the POI node \( v_j \). Lastly, \( \mu \) shows the weight of the POI cost of the POI node \( v_j \). All the objective functions (a) to (d) must relate to the parameter \( \alpha \) since the pheromones on the edge well reflect the past-trip records as well as the solution quality searched before. Also, \( \beta \) should be set to large since the time limitation must be satisfied. \( \gamma \) and \( \mu \) should be set to large depending on the objective functions used. The transition probability \( p_k(i, j) \) affects the selection of next POI node. It leads each ant to choose a better POI node for the objective functions used. The example of our selection of the next POI node by the state transition rule is shown below.

\[
p_k(i, j) = \frac{[\tau(i, j)]^\alpha [\eta(i, j)]^\beta [s_j]^\gamma [\xi(j)]^\mu}{\sum_{u \in J_k(i)} [\tau(i, u)]^\alpha [\eta(i, u)]^\beta [s_u]^\gamma [\xi(u)]^\mu}
\]

where \( s_j \) is POI satisfaction of \( v_j \), \( \alpha \) shows the weight of the pheromone on the edge \((i, j)\), \( \beta \) shows the weight of the time between the nodes \( v_i \) and \( v_j \), \( \gamma \) shows the weight of the POI satisfaction on the POI node \( v_j \). Lastly, \( \mu \) shows the weight of the POI cost of the POI node \( v_j \). All the objective functions (a) to (d) must relate to the parameter \( \alpha \) since the pheromones on the edge well reflect the past-trip records as well as the solution quality searched before. Also, \( \beta \) should be set to large since the time limitation must be satisfied. \( \gamma \) and \( \mu \) should be set to large depending on the objective functions used. The transition probability \( p_k(i, j) \) affects the selection of next POI node. It leads each ant to choose a better POI node for the objective functions (a) to (d) depending on the parameter setting. These four parameters are mainly determined based on [17] and [28]. Further, we have also conducted several preliminary experiments and finally determined the parameter values in Table 3.

In our method, we determine the four parameters in Eq. (6) depending on the iteration count. Let us consider the \( k \)-th iteration of Step 2–Step 3 in Fig. 4 and let \( k' = k \mod 5 \). The settings of the four parameters are summarized in Table 3. For example, for \( k' = 0 \), the four parameters are set as \( \alpha = 0.5, \beta = 1.0, \gamma = 0, \) and \( \mu = 1.0 \) corresponding to the objective functions (a) which aims at minimizing the total POI cost.

The example of our selection of the next POI node by the state transition rule is shown below.

\[1\] In our state transition rule, the parameters for the objective function (c) only consider the pheromones (\( \alpha = 1.0 \) and \( \beta = \gamma = \mu = 0 \)) without considering the similarity discussed in Section III-E3. This is because the pheromones on the edges already reflect the similarities to the past-trip records.

Meanwhile, the objective function (d) has the ants twice as many as the objective function (a), (b), or (c) has. Based on our preliminary studies, these settings of ant number guarantee the solution quality.

### TABLE 3. Parameters for the state transition rule for an ant \( k(k' = k \mod 5) \).

| \( k' \) | Objective | \( \alpha \) | \( \beta \) | \( \gamma \) | \( \mu \) |
|-------|-----------|---------|---------|---------|---------|
| 0     | (a)       | 0.5     | 1.0     | 0.0     | 1.0     |
| 1     | (b)       | 0.5     | 1.0     | 1.0     | 0.0     |
| 2     | (c)       | 1.0     | 0.0     | 0.0     | 0.0     |
| 3, 4  | (d)       | 1.0     | 1.0     | 1.0     | 1.0     |

**FIGURE 4. Example of a route selection by \( p_k(i, j) \).**

*Example 2:* With POI nodes, POI satisfactions and POI costs in Table 4, we suppose that an ant \( k \) starts at \( v_{st} \) in Fig. 4. If the aim of the ant \( k \) is the objective function (a), the parameters are set as \( \alpha = 0.5, \beta = 1.0, \gamma = 0, \) and \( \mu = 1.0 \) according to Table 3 and \( p_k(i, j) \) of each POI node is listed in the fourth column in Table 4. \( v_c \) should be selected at a probability of \( q_0 \) as the next POI node as \( p_k(i, j) \) is the highest among all \( p_k(i, j) \) values, and then we construct a sub-route \( v_{st} \to v_c \) shown in red in Fig. 4. In the same way, if the aim is the objective function (b), the parameters are set as \( \alpha = 0.5, \beta = 1.0, \gamma = 1.0, \) and \( \mu = 0 \) according to Table 4 and \( p_k(i, j) \) of each POI node is listed in the fifth column in Table 4. Likewise, \( v_b \) should be selected at a probability of \( q_0 \) as the next POI node and then we construct a sub-route \( v_{st} \to v_b \) shown in blue in Fig. 4. POI selection in the cases of the objective functions (c) and (d) will be conducted similarly.

**E. LOCAL SEARCH + PHEROMONE UPDATING (STEP 3)**

In Step 3, **local search** and two pheromone updatings, **local updating and similarity updating**, are performed.
1) LOCAL SEARCH

Local search tries to improve the route constructed by an ant through swapping POI nodes in the route.

Assume that $Tour^k$ is the route constructed by an ant $k$ in Step 2 aiming at the objective function (a). Let $v_i$ be a POI node in $Tour^k$. Let $v_j \in J_k(i)$ be one of the unvisited POI nodes. Then we swap $v_i$ and $v_j$ and see how much the total POI satisfaction is improved without exceeding the time limitation $T$.

Let $v'_j \in J_k(i)$ be the unvisited POI node which gives the maximum total POI satisfaction improvement after swapping $v_i$ and $v'_j$. Then we actually swap $v_i$ and $v'_j$. Note that, if any POI node in $J_k(i)$ does not improve total POI satisfaction without exceeding the time limitation, the POI node $v_j$ remains in $Tour^k$. The above process is repeatedly performed for every POI node in $Tour^k$.

The same process is applied to the routes aiming at the objective functions (b) and (c). But the local search process is not applied to the routes aiming at the objective function (d) as to save the order of POI nodes constructed by an ant considering the pheromones in the route.

2) PHEROMONE UPDATING 1: LOCAL UPDATING

After the route from $v_{st}$ to $v_{dst}$ is constructed by an ant $k$ in Step 2 and modified in local search of Step 3, the pheromones along the route is adjusted through local updating. The pheromones of the edge $(v_i, v_j)$ of two nodes $v_i$ and $v_j$ in the route are updated as:

$$\tau(i, j) \leftarrow (1 - \rho_1)\tau(i, j) + \rho_1\Delta\tau(i, j)$$

where $\rho_1$ is a parameter that controls the amount of pheromone evaporation and it is set as $\rho_1 = 0.01$ in our method based on [17] since $\rho_1$ must be small enough to gradually reflect the trajectory of an ant $k$. $\Delta\tau(i, j)$ is a constant that indicates the amount of pheromone which is added to edge $(v_i, v_j)$ by an ant $k$. In our method, the constant is $1/(nT)$ referring to the setting in [28], where $n$ is the number of the nodes.

Local updating makes other ants harder to follow the route of an ant $k$ to prompt them to explore various routes during Step 2 and Step 3 by evaporating the amount of pheromone deposit by an ant $k$.

3) PHEROMONE UPDATING 2: SIMILARITY UPDATING

After local updating, an additional pheromone is attached to the route based on the similarity to the past-trip records. It helps the ant $k + 1$ or further ants to build a route that considers the past-trip records.

a: INTRODUCTION OF SIMILARITY

Let $Tour^k$ be the route constructed in Step 2 and Step 3 by an ant $k$ and $T_k$ be its travel time. By adopting the concept of similarity calculation between strings, we regard nodes in $Tour^k$ as a string $str_k$ such as $str_k = “abc”$ if we have $Tour^k = \{a, b, c\}$. $l_k$ means the length of the string $str_k$. Likewise, $str_o = “ab”$ is a string corresponding to the past-trip record $Tour^{past} = \{a, b\}$, and its length is represented as $l_o$. Assume that $lsd(str_k, str_o)$ is the levenshtein distance [29] and $lcs(str_k, str_o)$ is the length of the longest common subsequence [30], between two strings. The similarity $Sim(Tour^k, Tour^{past})$ between two routes is defined as:

$$Sim(str_k, str_o) = \left[\rho_0 \left(1 - \frac{lsd(str_k, str_o)}{\max(l_k, l_o)}\right) + (1 - \rho_0)\frac{lcs(str_k, str_o)}{\min(l_n, l_o)}\right] \frac{T_k}{T}$$

(8)

where $\rho_0 = l_k/(l_k + l_o)$. $\rho_0$ is a parameter to add balance weights between $lsd(str_k, str_o)$ and $lcs(str_k, str_o)$ regardless of the length gap between $Tour^k$ and $Tour^{past}$. Even if the length of the past-trip record is quite short, $\rho_0$ prevents $Sim(Tour^k, Tour^{past})$ from becoming small.

As Shown in Eq. (8), the similarity depends on $lsd(str_k, str_o)$ and $lcs(str_k, str_o)$. If the two routes have the same and similar sequence, the similarity becomes larger since $lsd(str_k, str_o)$ becomes larger. Moreover, if $Tour^k$ shares POI nodes with $Tour^{past}$, the similarity also becomes larger since $lcs(str_k, str_o)$ becomes larger.

Example 3: Consider $Tour^k = \{a, b, c, d, e\}$ and $Tour^{past} = \{b, c, e\}$, and let $T_k = T$ for simplification. In this case, $lsd(str_k, str_o)$ becomes two and $lcs(str_k, str_o)$ becomes three. As a result, the similarity $Sim(Tour^k, Tour^{past})$ becomes 0.75. Moreover, $Sim(Tour^k, Tour^{past})$ becomes 1.00 if $Tour^k$ perfectly matches $Tour^{past}$ and $Sim(Tour^k, Tour^{past})$ becomes 0 if they have no shared nodes.
b: SIMILARITY UPDATING

The average value of similarities between a constructed route and all past-trip records is represented as \( \text{Sim}_k \) and \( \text{Sim}_k \) is called similarity of \( \text{Tour}^k \). Pheromone whose amount is \( \text{Sim}_k \cdot \rho_{old} \) is attached to all edges in \( \text{Tour}^k \). We set \( \rho_{old} = 3.0 \) based on our preliminary studies where 3.0 turns out to be the best value in the range of \((0.0, 10.0)\), since we reflect the similarity largely and then we set \( \rho_{old} \) to be a relatively large value.

If the route of an ant \( k \) has a high similarity to past-trip records, similarity updating makes other ants easy to follow the route of an ant \( k \) to construct a route also similar to past-trip records during Step 2 and Step 3.

4) UPDATE PARETO OPTIMAL SOLUTIONS

At the end of Step 3, Pareto optimal solutions are updated by comparing total POI satisfaction and total POI cost among each solution \( \text{Tour}^i \in S^{PO} \), where \( S^{PO} \) is the set of all Pareto optimal solutions constructed before.

Let \( S^{Tour^k} \) and \( C^{Tour^k} \) be the total POI satisfaction and total POI cost of \( \text{Tour}^k \) just constructed by an ant \( k \) in Step 2 and Step 3. In the same way, \( S^{Tour^l} \) and \( C^{Tour^l} \) represent those of one Pareto optimal solution \( \text{Tour}^l \). If there exists a Pareto optimal solution \( \text{Tour}^l \) that meets either \( S^{Tour^k} > S^{Tour^l} \) or \( C^{Tour^k} < C^{Tour^l} \), \( \text{Tour}^k \) is added to \( S^{PO} \). After that, for any two solutions \( \text{Tour}^l \) and \( \text{Tour}^k \), if \( S^{Tour^l} \geq S^{Tour^k} \) and \( C^{Tour^l} \leq C^{Tour^k} \), \( \text{Tour}^l \) is deleted from \( S^{PO} \). Hence, \( S^{PO} \) includes solutions, each of which has larger total POI satisfaction or smaller total POI cost compared to others.

In updating Pareto optimal solutions, we do not take into account the similarity of \( \text{Tour}^l \). This is because the similarity defined in Section III-E3 must relate to user preferences but does not directly reflect it. On the other hand, \( S^{Tour^k} \) and \( C^{Tour^k} \) directly reflect total POI satisfaction and total POI cost. In our trip planning strategy, we give the priority to total POI satisfaction and total POI cost and construct the Pareto optimal solutions as above.

F. PATH RELINKING + PHEROMONE UPDATING (STEP 4)

In Step 4, path relinking and global updating are performed.

1) PATH RELINKING

By repeating Step 3 and Step 4, several Pareto optimal solutions are constructed. The goal of path relinking is to optimize multi-objective functions in good balance by partially merging these Pareto optimal routes [31]. Path relinking is composed of node insertion and node deletion.

α: NODE INSERTION

Let \( \text{Tour}^{init} \) and \( \text{Tour}^{guide} \) be two routes constructed before (see Fig. 5). Let \( V_{gi} \) be the set of POI nodes that are included in \( \text{Tour}^{guide} \) but not in \( \text{Tour}^{init} \), \( V_{ig} \) is the set of POI nodes that are included in \( \text{Tour}^{init} \) but not in \( \text{Tour}^{guide} \). We try to insert every POI node in \( V_{gi} \) one by one into the best position into \( \text{Tour}^{init} \). Assume that we insert the POI node \( v_i \) in \( V_{gi} \) into an edge \((v_k, v_{k+1})\) in \( \text{Tour}^{init} \) and that the total travel time is increased by \( \delta t(i, k) \). When \( \delta t(i, k) \) is minimized, the edge \((v_k, v_{k+1})\) is the best insert position for \( v_i \). In Fig. 5(a), let us pick up \( v_5 \) in \( V_{gi} \). The best position for \( v_5 \) can be the edge \((v_1, v_2)\). \( \delta t_{min} \) is defined by the minimum \( \delta t(i, k) \) value for inserting every POI node \( v_i \in V_{gi} \) into its best insert position \((v_k, v_{k+1})\).

Then we find out the POI node \( v_w \) in \( V_{gi} \) maximizing Eq. (9) below:

\[
b(w) = \frac{[\tau(k, w)]^\alpha [\eta(k, w)]^\beta [\delta w]^\gamma [\xi(w)]^\mu}{\delta t_{min}}
\]  

where the edge \((v_k, v_{k+1})\) is the best position for inserting \( v_w \) and \( \alpha, \beta, \gamma, \) and \( \mu \) are the parameters of the objective function (d) in Table 3. If the total travel time does not exceed the time limitation \( T \), we actually insert the POI node \( v_w \) into \( \text{Tour}^{init} \).
at its best insert position. In Fig. 5(b), if $v_5$ in $V_{gi}$ maximizes Eq. (9), then $v_5$ is actually inserted into $Tour_{init}$ at the edge $(v_1, v_2)$.

$b(w)$ becomes large if $\delta t_{min}$ is small and/or the numerator in Eq. (9) is large. If $\delta t_{min}$ is small, we can insert the POI node $v_w \in V_{gi}$ into $Tour_{init}$ without much detouring. If the numerator in Eq. (9) is large, it means that the POI node $v_w \in V_{gi}$ is preferable in terms of the objective function (d). Then we pick up the POI node $v_w \in V_{gi}$ maximizing Eq. (9).

We try the above process until all the POI nodes in $V_{gi}$ are inserted into $Tour_{init}$.

b: NODE DELETION

Due to the time limitation $T$, it may be the case where no POI nodes can be inserted in the node insertion step described above.

In this case, we delete a POI node in $Tour_{init}$. We find out the POI node $v_x$ in $V_{gi}$ minimizing

$$r(x) = \frac{\left[ \tau(i, x)^\alpha \rho^i(i, x)^\beta \phi_s[i]^\gamma \tau(x)^\mu \right]}{d_x}$$

(10)

where $v_i$ is the preceding POI node for $v_x$ in $Tour_{init}$ and $\alpha$, $\beta$, $\gamma$, and $\mu$ are the parameters of the objective function (d) in Table 3. Then, we actually delete the POI node $v_x$ from $Tour_{init}$.

$r(x)$ becomes small if $d_x$ is large and/or the numerator in Eq. (10) is small. If $d_x$ is large, the POI node $v_x$ takes a long time to look around. If the numerator in Eq. (10) is small, it means that the POI node $v_x$ is not preferable in terms of the objective function (d). Then we delete the POI node $v_x$ from $Tour_{init}$.

c: PATH RELINKING PROCEDURE

The path relinking procedure is summarized as Algorithm 1. By applying this procedure, the constructed routes are mixed up and we expect that the good routes are constructed.

2) PHEROMONE GLOBAL UPDATING

After performing path relinking, we update the amount of pheromone for all Pareto optimal solutions in $S^{PO}$ and this adjustment is called global updating. In the updating, $\Delta \tau(k)$ is a pheromone adding parameter that adjusts the amount of pheromone for $Tour^k$ in $S^{PO}$.

For a solution $Tour^k \in S^{PO}$ if $S_{Tour^k}$ of $Tour^k$ is the highest or second highest among all solutions in $S^{PO}$, i.e., the solution is good enough in terms of the objective function (a), then $\Delta \tau(k)$ is calculated in Eq. (11).

$$\Delta \tau(k) = \rho_p \cdot \frac{S_{Tour^k}}{C_{Tour^k} \cdot T_k}$$

(13)

where $T_k$ shows the travel time of $Tour^k$.

In Eq. (11) to Eq. (13), $\rho_p$, $\rho_c$, and $p_p$ are parameters that adjust the amount of pheromone of the route where $\rho_p = 0.002$, $\rho_c = 0.0008$, and $p_p = 0.02$ as this combination obtained the best performance based on our preliminary studies.

Based on the above settings, the pheromone $\tau(i, j)$ of an edge $(v_i, v_j)$ in $Tour^k \in S^{PO}$ is updated as:

$$\tau(i, j) \leftarrow (1 - \rho_p)\tau(i, j) + \Delta \tau(k)$$

(14)

where $\rho_p$ is a parameter that controls the evaporation amount of pheromone and $\rho_p = 0.1$ in our method based on our preliminary studies since $\rho_p$ must also be small enough to gradually reflect the trajectories in the Pareto optimal solutions.

Global updating makes subsequent ants easy to follow the good routes in the iteration of Step 2 to Step 4.

G. ROUTE SELECTION & CONSTRUCTION (STEP 5)

In Step 5, several routes corresponding to the objective functions (a) to (d) are provided to users. The selection is conducted in $S^{PO}$ after $N$-th iteration of Step 2–Step 4. For
TABLE 5. Dataset details.

| Target area | #POIs | #past-trip records | $v_{st}$ | $v_{dst}$ |
|-------------|-------|---------------------|----------|-----------|
| Kyoto       | 150   | 315                 | Kyoto Sta. | Renka no Yu |
| Sapporo     | 100   | 146                 | Sapporo Sta. | Sapporo Grand Hotel |
| Tokyo       | 150   | 154                 | Tokyo Sta. | Tokyo Dome Hotel |

the objective functions (a) and (b), a route with the highest total POI satisfaction and a route with the lowest total POI cost in all Pareto optimal solutions are selected, respectively. For the objective function (c), a route that has the highest similarity $\text{Sim}_{k}$ is selected in all Pareto optimal solutions.

For the objective function (d), a route with a minimum value of $w_{v}$ of Eq. (4) is selected in all Pareto optimal solutions, as described in Definition 1. $w_{v}$ analyzes how the three objective functions (a) to (c) are balanced in each route.

Totally, we output four routes corresponding to the objective functions (a) to (d) in Step 5.

IV. EXPERIMENTAL EVALUATIONS

In this section, we evaluate our proposed method with real map data.

A. EXPERIMENT METHOD

1) DATASETS

The aim of this study is to use the past-trip records for trip planning to reflect the users’ requests. Therefore, we have to use real-world datasets which have a sufficient amount of past-trip records for experiments to evaluate our method. The datasets used for evaluation are shown in Table 5. The target areas of the datasets are Kyoto with 150 POIs, Sapporo with 100 POIs, and Tokyo with 150 POIs. These areas are expected to have a sufficient amount of past-trip records because they are popular travel destinations in Japan.

To make a practical route according to the travel site [32], a train station that is frequently visited is set as the starting location and the most popular hotel in the area is set as the ending location in these datasets. POIs in the datasets include a wide variety of categories such as restaurants, parks, museums, and temples. The time limitation is set as ten hours for a one-day trip. POI satisfaction, POI cost, duration time, and the past-trip records are obtained from [32].

POI satisfaction is an average score of rating from real users which ranges from 1.0 as minimum and 5.0 as maximum. POI cost represents the users’ average food cost at the restaurant if POI is a restaurant, otherwise POI cost represents the entrance fee which ranges from 0 yen to a maximum of 22,000 yen. Note that, if POI cost of a restaurant is not available in [32], average POI cost on a gourmet site [33] is used as the restaurant’s POI cost. Furthermore, for POIs of shopping malls and markets, the POI cost is set as 2,456 yen utilizing the average cost for general dietary at travel in [32].

Duration time represents the average time users spend at a POI. The past-trip records are a real one-day trip footprint at the target area posted to [32]. Past-trip records include 315 routes in Kyoto where the average number of POIs visited in each record is 4.96, 146 routes in Sapporo where the average number of POIs visited in each record is 3.15, and 154 routes in Tokyo where the average number of POIs visited in each record is 3.14.

Travel time is the driving time between any two POIs at the speed of 30 km/h which is near the average driving speed on general roads in Japan [34].

2) COMPARISON METHOD

To confirm the effectiveness of our method, we compare our method with two baseline methods, P-VNS [17] and MOACOPW [24] which deal with multi-objective functions and satisfies the first condition in Section I. Note that, there is no method that sufficiently satisfies the first and second conditions as far as we know.

Baseline method 1 (P-VNS): P-VNS [17] is one of the local search methods that replaces nodes in the initial solution with other nodes to explore neighborhoods. To deal with the trade-off relations between different objective functions, P-VNS constructs several Pareto optimal solutions and varies weights of objective functions. In our experiments for P-VNS, we only use total POI satisfaction and total POI cost as its objective functions. Note that, it is difficult to incorporate a mechanism for considering the visiting orders and relationships of past-trip records in the objective function (c) into the P-VNS since P-VNS is based on the local search composed of node replacement and node insertion. At the same time, P-VNS adopts an inferior solution with a constant probability to avoid local solutions at every update. Since P-VNS stores Pareto optimal solutions in it, we select the four solutions as its outputs corresponding to the objective functions (a) to (d), respectively, as in Section III-G. Furthermore, in order to evaluate the distribution of the entire solutions including intermediate solutions, all constructed Pareto optimal solutions are used for evaluations in Figs. 6 and 7 and the discussion in Section IV-D.

We implemented P-VNS in Python3 on the server computer (CPU: Intel Xeon Gold 6130 2.10GHz, Memory: 768GB). P-VNS was conducted ten times and...
the average execution time of the method is shown in Table 10.

**Baseline method 2 (MOACOPW):** MOACOPW [24] is one of the latest multi-objective ACO methods based on decomposition framework [35] which has the advantages of low time complexity and good convergence performance. In MOACOPW, the multi-objective problem is decomposed into several sub-problems, and the ant colony is divided into several sub-groups to cover all objectives. We add the time element and limit time to MOACOPW because it is originally designed for the multi-objective traveling salesman problem (TSP). We also replace the objectives for the multi-objective TSP with the objectives (a) to (d) in the MOTPP. The parameters in MOACOPW are the same as [24]. Since the MOACOPW stores Pareto optimal solutions in it, we select the four solutions as its outputs corresponding to the objective functions (a) to (d), respectively. Furthermore, in order to evaluate the distribution of the entire solutions, all constructed Pareto optimal solutions are used for evaluations in Fig. 6, Fig. 7, and Fig. 8 and the discussion in Section IV-D.
We implemented MOACOPW in Python3 on the server computer (CPU: Intel Xeon Gold 6130 2.10GHz, Memory: 768GB), and it was conducted ten times and the average execution time of the method is shown in Table 10.

**Ours:** Parameters in our method are summarised in Table 6. We set the iteration times $N$ as 50, and $M$ as 20 which is the number of ants for one outer iteration. In our method, the objective functions (a) to (d) are considered by each ant in route construction. Although the output of our method is four solutions corresponding to the objective functions (a) to (d), not only the four solutions, but all constructed Pareto optimal solutions are used for evaluation together in Figs. 6 and 7 and the discussion in Section IV-D.

We implemented our method in Python3 on the server computer (CPU: Intel Xeon Gold 6130 2.10GHz, Memory: 768GB). Our method was conducted ten times and the average execution time of the method is shown in Table 10.

**B. TRIP PLANNING RESULT**

The result for our method, P-VNS, and MOACOPW of Kyoto is shown in Table 7, the result of Sapporo is shown in Table 8, and the result of Tokyo is shown in Table 9. In these tables, our method, P-VNS, and MOACOPW are executed ten times and the averaged data are shown. In the table, “POI satisfaction (a)”, “POI cost (b)”, and “Similarity (c)” correspond to the objective functions (a) to (c) respectively. Those values with “(d)” mean the values selected for the objective function (d). “Travel time” shows the average travel time of all Pareto solutions and “#POIs” represents the average number of POIs in one route. “#solutions” represents the number of Pareto solutions. The distribution of solutions for Kyoto, Sapporo, and Tokyo by our method, P-VNS, and MOACOPW are shown in Fig. 6, Fig. 7, and Fig. 8.

1) **OBJECTIVE FUNCTIONS (A) AND (B)**

Focusing on the “POI satisfaction (a)”, our method constructs more satisfying routes than P-VNS and MOACOPW. Comparing the standard deviations of the “POI satisfaction (a)”, our method has much smaller values than P-VNS and MOACOPW, and this indicates that our method constructs stabilized and cohesive routes corresponding to the objective function (a).

On the other hand, “POI cost (b)” routes based on the objective function (b) show that all of them have the same cost of 0 yen in our method and P-VNS, while MOACOPW obtains an average cost of 2 to 5 thousand yen. Analyzing the 0 yen routes implies that those routes include many popular temples, shrines, and parks with no admission fee. Thus, our method explores better routes than P-VNS and MOACOPW for the first condition.

2) **OBJECTIVE FUNCTION (C)**

For the objective function (c), the values of “Similarity (c)” in our method is $((0.065/0.057) − 1) × 100% ≈ 14\%$ (Kyoto), $((0.110/0.077) − 1) × 100% ≈ 43\%$ (Sapporo), and $((0.067/0.048) − 1) × 100% ≈ 40\%$ (Tokyo) higher than those of P-VNS, and 7\% (Kyoto), 16\% (Sapporo), and 56\% (Tokyo) higher than those of MOACOPW, and this means that our method achieves the second condition better than P-VNS and MOACOPW.

3) **OBJECTIVE FUNCTION (D)**

We compare the objective function (d) values in the tables. In Kyoto, the route our method selects is 7\% higher and 2\% higher in POI satisfaction, 239\% higher and 21\% lower in POI cost, and 4\% higher and 2\% lower in similarity compared with P-VNS and MOACOPW, respectively. In Sapporo, the route by our method is 9\% higher and 6\% higher in POI satisfaction, 56\% lower and 64\% lower in POI cost, and 40\% lower and 64 lower in POI cost, and 40\%
higher and 27% higher in similarity compared with P-VNS and MOACOPW, respectively. In Tokyo, the route by our method is 2% higher and 3% higher in POI satisfaction, 26% higher and 60% lower in POI cost, and 23% higher and 55% higher in similarity compared with P-VNS and MOACOPW, respectively.

As the aim of the objective function (d) is to find a route considering all the objectives, it is more difficult to evaluate the constructed routes than other objectives due to the existence of functions with trade-off relations. Thus, the standard deviation and the distribution of solutions are used to evaluate the stability and goodness of entire solutions. As Table 7, Table 8, and Table 9 show that it can be seen that all standard deviations of our method are lower than those of P-VNS, and this indicates that our method stably explores a route that has multi-objective functions (a) to (c). For comparison with MOACOPW, our method obtains smaller standard deviations than MOACOPW in almost all the terms, expect the standard deviations of “POI cost (d)” in Sapporo and “Similarity (d)” in Tokyo. In general, our method performs more stably than MOACOPW.

“#solutions” is also higher in our method than P-VNS which leads to a high probability of selecting a better-balanced route because of the large number of options for the objective function (d). MOACOPW obtains higher “#solutions” than our method in Sapporo and Tokyo while the solutions are degraded in the three terms of the objective function (d). On the other hand, how to construct more solutions without degradation is still a further work.
Fig. 6, Fig. 7, and Fig. 8 show the distributions of the solutions tendencies. Our method obtains more well-balanced solutions compared with P-VNS and MOACOPW (see Section IV-D in the detailed discussions using three indicators).

The example routes for the objective function (d) by our method, P-VNS, and MOACOPW are shown in Fig. 9 and Table 11.

Focusing on “POI satisfaction”, “POI cost”, and “Similarity” in Table 11, the route our method selects is 1% higher and 2% higher in POI satisfaction, 33% lower and 68% lower in POI cost, and 65% higher and 65% higher in similarity compared with P-VNS and MOACOPW, respectively. These values suggest that our method accomplishes the first condition.

To examine the achievement of the second condition, and the validity of the similarity, frequently appeared POIs and sequences in the past-trip records in Kyoto are obtained and listed in Table 12 and Table 13. Past-trip records appeared in our method, P-VNS, and MOACOPW are shown in brown route and purple node in Fig. 9. We summarize the number of past-trip records that appeared in our method, P-VNS and MOACOPW in Table 14, and it indicates that our method contains more past-trip records than P-VNS and MOACOPW. Since the similarity in Table 14 of our method is 60% higher and 60% higher than that of P-VNS and MOACOPW, respectively, it is reasonable that our method includes more past-trip records. Therefore, P-VNS and MOACOPW are insufficient to meet the second condition, meanwhile, our method calculates the similarity and uses the past-trip records
as initial pheromones to meet the second condition and satisfies the user’s various requests.

4) TRAVEL TIME
As Table 7, Table 8, and Table 9 indicate, all the three methods do not exceed the time limitation in all datasets. Especially, “Travel time” of our method is closer to the time limitations than P-VNS and MOACOPW in Kyoto and Sapporo. Our method’s routes include more POIs than P-VNS and MOACOPW according to the results of “#POIs” in Table 7, Table 8, and Table 9. Thus, it is reasonable that “Travel time” of our method is slightly longer than that of the other two methods.

C. ADDITIONAL EVALUATION
In the experiments in Section IV-B, we set the starting location and the ending location separately, and the results show the effectiveness of our method in the case where the starting location and the ending location are different. This shows one example of the settings of the starting and ending locations.

Meanwhile, as users are allowed to freely set the starting and ending locations, our method is supposed to flexibly deal with other settings of the starting and ending locations. For example, some users tend to have the starting location and the ending location at the same position. Thus, to further confirm the effectiveness of our method, we additionally conduct experiments on the case where the starting location and the ending location are the same. The additional experiment was conducted using the Kyoto dataset, and the process was executed completely the same as the process described in Section IV-A.

The results of our method are shown in Table 15 and an example of the route for the objective function (d) is shown in Fig. 12. According to the results of our method in Table 7 and Table 15, our method obtains the same or better values than the result in Table 7 from the terms “POI satisfaction (a)” to “#solutions”. Thus, in the case where the starting location and the ending location are the same, our method still can construct routes that meet users’ requests.

Also, the route in Fig. 12 is realistic without any redundant and overlapped sub-route, because all POIs in the route are naturally and consecutively connected to their next POIs with no turning back.

D. CONVERGENCE EVALUATION
Evaluation of solutions with multi-objective functions is difficult due to the trade-off relations between objective functions. Therefore, not only the individual solution, but also the convergence, diversity, and distribution breadth of the solutions should be evaluated. We use the following three indicators: Generational Distance (GD) [36] to evaluate the convergence; Inverted Generational Distance (IGD) [37] to evaluate the convergence and the diversity; and Maximum
The definitions of three indicators are shown in below.

**Generational Distance (GD):** GD is an evaluation indicator that evaluates the convergence of solutions by calculating the distance between a set of Pareto optimal solutions and a set of approximate ones [36]. GD is represented as:

$$GD = \sqrt{\frac{\sum_{i=1}^{n} d(PF_i, OP)^2}{n}}$$  \hspace{1cm} (15)

where $n$ is the number of approximate solutions, $PF_i$ is $i$-th approximate solution, and $OP$ is a set of Pareto optimal solutions. $d(PF_i, OP)$ means Euclidean distance between $i$-th approximate solution and the nearest Pareto optimal solution. The smaller GD indicates, the closer the approximate solution is to the optimal solution, and the higher the accuracy of the approximate solutions is.

**Inverted Generational Distance (IGD):** IGD evaluates the convergence and diversity of solutions in a similar way to GD. IGD is calculated as:

$$IGD = \frac{1}{n} \sum_{i=1}^{n} d(OP_i, PF)$$

$$d(OP_i, PF) = \min_{p \in PF} \| OP_i - p \|$$  \hspace{1cm} (16)

where $n$ is the number of Pareto optimal solutions, $OP_i$ is $i$-th Pareto optimal solution and $PF$ is a set of approximate solutions. $d(OP_i, PF)$ means Euclidean distance between $i$-th Pareto optimal solution and the nearest approximate solution. When calculating IGD, we calculate the distance to all Pareto optimal solution to evaluate the diversity of approximate solutions. Similar to GD, the smaller IGD is, the closer the approximate solution is to the optimal solution, and the wider diversity of the solution is.

**Maximum Spread (MS):** MS is an evaluation indicator that evaluates the spread of solution distribution by comparing the maximum value with the minimum one in each objective function of approximate solutions. MS is calculated as:

$$MS = \sqrt{\frac{\sum_{i=1}^{k} \max \{ \| a_i - b_i \| : a, b \in PF \}}{k}}$$  \hspace{1cm} (17)

where $k$ is the number of the objective functions, $a_i$ and $b_i$ is a value corresponding to $i$-th objective function in approximate solutions $a$ and $b$. The distribution spread is evaluated by calculating the distance between the maximum and minimum value of each objective function. The larger MS indicates the more spread and various solutions are constructed.

The result of GD, IGD, and MS of our method, P-VNS, and MOACOPW is shown in Table 16. To evaluate POI satisfaction, POI cost, and similarity, all values are normalized with a maximum of 1 and a minimum of 0. In this evaluation, we use all the Pareto optimal solutions stored in our method, P-VNS, and MOACOPW as depicted in Fig. 6, Fig. 7, and Fig. 8.

1) **GENERATIONAL DISTANCE (GD)**

Focusing on GD in Table 16, the GD values of our method are 92% lower and 88% lower in Kyoto, 41% lower and 37% lower in Sapporo, and 56% lower and 25% higher in Tokyo than those of P-VNS and MOACOPW, respectively. This result indicates that the convergence and the quality are higher in our method for the three datasets except for Tokyo with MOACOPW. As MOACOPW has more solutions in Tokyo that may cover optimal solutions in Eq. (15), and it leads a higher GD value than our method. Therefore, how to construct more solutions is a future work.

2) **INVERTED GENERATIONAL DISTANCE (IGD)**

IGD of our method is lower than those of P-VNS and MOACOPW, around 96% and 98% in Kyoto, 41% and
### Table 16. Average values of three indicators (standard deviation).

| Method    | Area   | GD       | IGD      | MS       |
|-----------|--------|----------|----------|----------|
| Our method| Kyoto  | 0.0167 (0.0008) | 0.0133 (0.0042) | 1.6733 (0.0347) |
|           | Sapporo| 0.3632 (0.1155) | 0.3779 (0.1840) | 1.5359 (0.1467) |
|           | Tokyo  | 0.1739 (0.0398) | 0.0939 (0.0511) | 1.6899 (0.0382) |
| P-VNS     | Kyoto  | 0.2083 (0.1620) | 0.3455 (0.3777) | 1.7401 (0.3202) |
|           | Sapporo| 0.6147 (0.1016) | 1.2494 (0.2844) | 1.1300 (0.3539) |
|           | Tokyo  | 0.3916 (0.1120) | 0.6325 (0.2984) | 1.3546 (0.2069) |
| MOACOPW   | Kyoto  | 0.1416 (0.0143) | 0.7623 (0.0439) | 1.4057 (0.0865) |
|           | Sapporo| 0.5772 (0.0954) | 2.5477 (0.2622) | 1.7104 (0.1992) |
|           | Tokyo  | 0.1386 (0.0238) | 0.7400 (0.0596) | 0.9221 (0.1684) |

85% in Sapporo, and 85% and 87% in Tokyo, respectively. The standard deviation of our method is lower than that of P-VNS and MOACOPW, especially in Kyoto. This result indicates that our method constructs more diverse solutions than P-VNS and MOACOPW and the tendency is remarkable in Kyoto.

3) **MAXIMUM SPREAD (MS)**

The average MS value of P-VNS is higher than that of our method in Kyoto, but this relation is inverse in Sapporo and Tokyo. However, in Kyoto, the standard deviation of our method is around 1/10 of the standard deviation of P-VNS. Generally, since P-VNS does not stably construct a route with higher MS for the Kyoto dataset, and the standard deviation of our method is also lower than that of P-VNS in Sapporo and Tokyo.

Also, the value of MOACOPW is higher than that of our method in Sapporo, but this relation is inverse in Kyoto and Tokyo. It may be the reason that MOACOPW constructs solutions that have a wider range of similarity which include a large number of low similarities as shown in Fig. 7. It suggests that the solution distribution is not so good in MOACOPW.

This result shows that our method constructs more diverse and stable solutions with wider distributions in three datasets.

In conclusion, it is shown that our method constructs solutions that have better distributions and simultaneously meet the first and second conditions in Section I compared to P-VNS and MOACOPW.

### E. USER STUDY

In this subsection, we evaluate the effectiveness of the similarity function explained in Section III-E3 and the validity of a route through user evaluations. The purpose of the user study is that we want to see how much every user overall satisfies the route if the similarity of the constructed route to the past-tour records becomes high, and hence we want to see if the similarity introduced in Section III-E3 is reasonable or not.

1) **EXPERIMENTAL METHOD**

For the three target areas in Section IV-A1, we conducted an evaluation experiment by providing routes of the objective function (d) constructed by our method and P-VNS to 20 participants who are familiar with this area. In this experiment, we especially focus on similarity between the objective function (d) route and the past-trip records. We did not compare our method with MOACOPW, because the performances of MOACOPW and P-VNS are similar, and P-VNS has five better values than MOACOPW, either for the results of the objective function (d) in Table 7–Table 9 or the three indicators in Table 16.

The routes for evaluation are shown in Fig. 9, Fig. 10, and Fig. 11. Questions (1), (2), and (3) in Fig. 13 were presented to the participants.

2) **THE RESULT**

The answers are summarized in Fig. 14, Fig. 15, and Fig. 16. For Question (1), our method obtains the interquartile ranges of $\geq 4$ points from all participants, while P-VNS only obtains all the interquartile ranges of $\leq 4$ points. Moreover, the average and median of our method are all over 4.5 points.

For Question (2), our method has an equal or a smaller interquartile range than P-VNS.
For Question (3), our method obtains the rates $\geq 3$ points from all participants in Kyoto area, while P-VNS only obtains all rates $\leq 3$ points. In Sapporo area, our method obtains a smaller interquartile range while P-VNS obtains higher interquartile range. Also, our method obtains higher average scores than P-VNS in Tokyo area.

Our method obtains an equal or smaller interquartile range and a higher average score in both Questions (1) and (2). Therefore, our method constructs more attractive and suitable routes for users. In addition, our method considers a similarity function to reflect the past-trip records on the routes, and the routes of our method have a higher value in similarity than those of the baseline method according to Tables 7, 8, and 9. This result indicates that the similarity function is effective for reflecting the past-trip records and valid to construct routes reflecting users’ preferences.

Moreover, based on the results of Question (3), it is reasonable that users show higher satisfaction in the route by our method as we consider all the objective functions (a) to (c) into the objective function (d). In other words, the objective function (d) is effective to meet users’ multiple requests.

F. ANSWERS TO RESEARCH QUESTIONS

In this subsection, we answer the two research questions, RQ1 and RQ2, shown in Section I based on the evaluation results in Section IV-B to Section IV-D.

RQ1: Does the proposed method outperform the existing trip planning method?

We can say YES to RQ1 because it is indicated that the proposed method accomplishes the first and second conditions for trip planning and constructs better routes.
than P-VNS and MOACOPW based on the results in Section IV-B. Also, the proposed method outperforms P-VNS and MOACOPW because it constructed more diverse and stable solutions with wider distributions than P-VNS and MOACOPW based on the results in Section IV-D.

RQ2: Do the objective functions in the proposed method work effectively to meet users’ requests?

We can say YES to RQ2 because based on the results of Q1–Q3 in Section IV-E, it is shown that the routes constructed by our method satisfy users’ requests better than routes by P-VNS.

To sum up, the results in Section IV-B and Section IV-E confirm the effectiveness of utilization of past-trip records in the objective function (c) which can better satisfy users’ multiple requests at the same time. Then, based on the results in Section IV-D, by utilizing past-trip records for the objective function (d), more diverse and stable solutions can be generated.

V. CONCLUSION

In this paper, we proposed a trip planning method utilizing ACO. This method utilizes ACO and it plans one-day trips with considerations of the past-trip records and multi-objective optimization to meet users’ multiple requests. Our proposed method is compared to P-VNS and MOACOPW, which are the existing effective multi-objective optimization methods. The evaluations on real-world datasets show that our method constructs routes with better values for all the objectives and these routes are more similar to the past-trip records than P-VNS and MOACOPW. Then, we evaluated our method and P-VNS and MOACOPW through three indicators, and the results show that our method constructs more diverse and stable solutions. Moreover, as a result of the user study, our method obtains higher scores and it indicates that using past-trip records helps meet users’ requests better.

Currently, our method can only deal with one-day trip planning. In the future, we will extend the proposed method with additional considerations on hotel selection so that it can be applied to multi-day trip planning. Also, we will further consider time window constraints due to the conditions such as traffic regulations and shortened business hours of POIs.

REFERENCES

[1] AVA Intelligence, AVA Travel. Accessed: May 28, 2022. [Online]. Available: https://travel.ava-intel.com/

[2] Machidukuri Platform, [Route Optimization and Automatic Creation of Sightseeing Plans by AI] Ra-to Saito (in Japanese). Accessed: May 28, 2022. [Online]. Available: https://www.machi-pla.com/concierge

[3] S. Bao, M. Tawada, S. Tanaka, and N. Togawa, “Multi-day travel planning using Ising machines for real-world applications,” in Proc. IEEE Intell. Transp. Syst. Conf. (ITSC), Sep. 2021, pp. 3704–3709.

[4] S. Isoda, M. Hidaka, Y. Matsuda, H. Suwa, and K. Yasumoto, “An on-site planning method in tour navigation for enabling visit to must-see spots at better times (in Japanese),” in Proc. Multimedia, Distrib., Cooperat., Mobile Symp. (DICOMO), 2020, pp. 1487–1498.

[5] H. Kasahara, K. Tamura, M. Iiyama, and M. Mukunoki, “Tourist behavior model construction based on tracks of tourists using regional environmental factors and application (in Japanese),” J. Inf. Process. Soc. Jpn., vol. 57, no. 5, pp. 1411–1420, 2016.

[6] L. Lina, E. K. Noussaima, A. Belaid, and E. Yousef, “Mining tourism as an orienteering problem: A real-world data case,” in Proc. 5th Int. Conf. Logistics Oper. Manag. (GOL), Oct. 2020, pp. 1–5.

[7] Z. Friggstad, S. Gollapudi, K. Kolliads, T. Sarlos, C. Swamy, and A. Tomkins, “Orienteering algorithms for generating travel itineraries,” in Proc. 11th ACM Int. Conf. Web Search Data Mining, Feb. 2018, pp. 180–188.

[8] Y.-L. Hseue and H.-M. Huang, “Personalized itinerary recommendation with time constraints using GPS datasets,” Knowl. Inf. Syst., vol. 60, no. 1, pp. 523–544, Jul. 2019.

[9] X. Wu, C. Wang, J.-M. Yang, Y. Pang, and L. Zhang, “Photo2Trip: Generating travel routes from geo-tagged photos for trip planning,” in Proc. Int. Conf. Multimedia, 2010, pp. 143–152.

[10] B. L. Golden, L. Levy, and R. Vohra, “The orienteering problem,” Nav. Res. Logistics, vol. 34, no. 3, pp. 307–318, 1987.

[11] W. Souffriau, P. Vansteenwegen, J. Vertommen, G. V. Bergh, and D. Van Oudheusden, “A personalized tourist trip design algorithm for mobile tourist guides,” Appl. Intell., vol. 22, no. 10, pp. 964–985, 2008.

[12] C. Voudouris and E. Tsang, “Guided local search and its application to the traveling salesman problem,” Eur. J. Oper. Res., vol. 113, no. 2, pp. 469–499, Mar. 1999.

[13] K. Ostrowski, J. Karbowska-Chilinska, J. Koscelew, and P. Zabielski, “Evolution-inspired local improvement algorithm solving orienteering problem,” Ann. Oper. Res., vol. 253, no. 1, pp. 519–543, Jun. 2017.
I. K.ara, P. B. Ciacic, and T. Derya, “New formulations for the orienteeing problem,” *Proc. Econ. Finance*, vol. 39, pp. 849–854, Jan. 2016.

V. Campos, R. Marti, J. Sánchez-Oro, and A. Duarte, “GRASP with path relinking for the orienteeing problem,” *J. Oper. Res. Soc.*, vol. 65, no. 12, pp. 1800–1813, Dec. 2014.

A. Elzein and G. A. Di Caro, “A clustering metaheuristic for large orienteeing problems,” *PloS ONE*, vol. 17, no. 7, Jul. 2022. Art. no. e0271751.

M. Schilde, K. F. Doerner, R. F. Hartl, and G. Kiechle, “Metaheuristics for the Bi-objective orienteeing problem,” *Swarm Intell.*, vol. 3, no. 3, pp. 179–201, Sep. 2009.

M. Dorigo, V. Maniezzo, and A. Colorni, “Ant system: Optimization by a colony of cooperating agents,” *IEEE Trans. Syst., Man, Cybern., B*, vol. 26, no. 1, pp. 29–41, Feb. 1996.

N. Mladenovic and P. Hansen, “Variable neighborhood search,” *Comput. Oper. Res.*, vol. 24, no. 11, pp. 1097–1100, Nov. 1997.

R. Martín-Moreno and M. A. Vega-Rodríguez, “Multi-objective artificial bee colony algorithm applied to the Bi-objective orienteeing problem,” *Knowl.-Based Syst.*, vol. 154, pp. 93–101, Aug. 2018.

W. Deng, X. Zhang, Y. Zhou, Y. Liu, X. Zhou, H. Chen, and H. Zhao, “An enhanced fast non-dominated solution sorting genetic algorithm for multi-objective problems,” *Inf. Sci.*, vol. 585, pp. 441–453, Mar. 2022.

P. Nunes, A. Mozara, J. P. Santos, and A. Compeño, “A simulated annealing algorithm to solve the multi-objective bike routing problem,” in *Proc. Int. Symp. Comput. Sci. Intel. Contr. (ISCSCI)*, Nov. 2021, pp. 39–45.

Y. Mei, F. D. Salim, and X. Li, “Efficient meta-heuristics for the multi-objective time-dependent orienteeing problem,” *Eu. J. Oper. Res.*, vol. 254, no. 2, pp. 443–457, Oct. 2016.

L. Yang, X. Jia, and G. Liu, “Multi-objective ant colony algorithm based on pheromone weight,” in *Proc. 16th Int. Conf. Comput. Intell. Secur. (CIS)*, Nov. 2020, pp. 49–53.

K. Miettinen, F. Ruiz, and A. P. Wierzbicki, “Introduction to multi-objective optimization: Interactive approaches,” in *Multiobjective Optimization*. Berlin, Germany: Springer, 2008, pp. 27–57.

C. M. Fonseca and P. J. Fleming, “An overview of evolutionary algorithms in multiobjective optimization,” *Evol. Comput.*, vol. 3, no. 1, pp. 1–16, 1995.

J. G. Lin, “Multiple-objective problems: Pareto-optimal solutions by method of proper equality constraints,” *IEEE Trans. Autom. Control*, vol. AC-21, no. 5, pp. 641–650, Oct. 1976.

M. Dorigo and L. M. Gambardella, “Ant colony system: A cooperative learning approach to the traveling salesman problem,” *IEEE Trans. Evol. Comput.*, vol. 1, no. 1, pp. 53–66, Aug. 1997.

V. I. Levenshtein, “Binary codes capable of correcting deletions, insertions, and reversals,” *Sov. Phys.-Dokl.*, vol. 10, no. 8, pp. 707–710, 1966.

J. W. Hunt and T. G. Szymanski, “A fast algorithm for computing longest common subsequences,” *Commun. ACM*, vol. 20, no. 5, pp. 350–353, 1977.

R. Montemanni and L. M. Gambardella, “An ant colony system for team orienteeing problems with time windows,” *Found. Comput. Decis. Sci.*, vol. 34, no. 4, pp. 287–306, 2009.

Recruit Co. *Jalan*. Accessed: Jun. 3, 2022. [Online]. Available: https://www.jalan.net/)

Kakaku.com Inc. *Tabegu*. Accessed: Jun. 3, 2022. [Online]. Available: https://tabegu.com/

Heisei 27 Nendo Zenkoku Douro Gairou Koutuu Jyosei Tyosa Ippan Koutuuryo Tyosa Syunkeihibo (in Japanese). Accessed: Jun. 16, 2022. [Online]. Available: https://www.mlit.go.jp/road/census/h27/data/pdf/syunkei605.pdf

I. Ke, Q. Zhang, and R. Battiti, “MOEA/D-ACO: A multiobjective evolutionary algorithm using decomposition and ant colony,” *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1845–1859, Dec. 2013.

D. A. Van Veldhuizen and G. B. Lamont, “Evolutionary computation and convergence to a Pareto front,” in *Proc. Late Breaking Papers At Genetic Program. Conf.*, 1998, pp. 221–228.

E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. D. Fonseca, “Performance assessment of multiobjective optimizers: An analysis and review,” *IEEE Trans. Evol. Comput.*, vol. 7, no. 2, pp. 117–132, Apr. 2003.

E. Zitzler, *Evolutionary Algorithms for Multiobjective Optimization: Methods and Applications*. Herszenroth, Germany: Shaker Verlag, 1999.