Intelligent Environments 2019
A. Muñoz et al. (Eds.)

doi:10.3233/AISE190040

A Method for Generating Multiple Tour Routes Balancing User Satisfaction and Resource Consumption

YODAI HIRANO [a,b], HIROHIKO SUWA [a,b] and KEIICHI YASUMOTO [a,b]
[a] *Nara Institute Science and Technology, Nara 630-0192, Japan*
[b] *RIKEN, Center for Advanced Intelligence Project, Tokyo 103-0027, Japan*

Abstract. In general, it is a complicated and time-consuming task for a tourist to plan a satisfactory sightseeing tour, because he/she must take into account various factors and constraints (e.g., budget, available time, etc) at the same time. This difficulty comes from the fact that there is a trade-off between the satisfaction/experience obtained by the sightseeing tour and the resource consumed for the tour, hence the optimal solution is not unique. To help decision making, it is desirable to show the tourist a variety of solutions (i.e., tours) considering the trade-off in various ways, but to the best of our knowledge, no existing methods/systems provide such a wide variety of solutions. In this paper, we formulate the sightseeing tour recommendation as a multi-objective optimization problem with money, time and stamina consumption of a tourist and satisfaction degree obtained by the tourist as independent variables. Since this problem is NP-hard, we propose a heuristic algorithm to quickly obtain semi-pareto optimal solutions based on genetic algorithm NSGA-II. We applied the proposed method to planning tours targeting 30 tourist spots in Higashiyama-area of Kyoto, Japan. As a result, our algorithm could output semi-pareto optimal solutions in reasonable time.

Keywords. tour planner, multi-objective GA, route search, decision making

1. Introduction

Thanks to the wide-spread of smart devices with GPS, navigation systems that navigate a user from the current location to the destination location are now widely available. Such a navigation system like Google Maps has a route planning function that helps a user decide his/her satisfactory route in terms of monetary and/or temporal constraints, by showing multiple routes with (possibly combination of) multiple transportation means (trains, buses, taxis, etc).

Most of the existing route planning functions/systems, however, are suffering from its usability that a user needs to specify detailed conditions to get his/her satisfactory route. A more easy-to-use route planning system which takes into account user's context and preference is desired.

In the domain of tourism, there is a route planning system called the tour planner such as NAVITIME TRAVEL[1]. The tour planner suggests a user a tour plan considering

[1] https://travel.navitime.com/ja/

his/her preference, but it requires a user to input a lot of information such as selection of PoIs (Point of Interest), hence takes a lot of time to obtain a plan. To solve this problem, there have been proposed automatic tour planning systems (ATPS) which automatically select PoIs in the specified area and compose/show users a sightseeing tour including moving ways between PoIs.

In existing ATPS, various objective functions are considered. Shiraishi et al.[1] proposed an ATPS which solves a multi-objective optimization problem taking into account a trade-off between monetary cost and satisfaction degree. Wu et al. [2] proposed an ATPS which maximizes satisfaction degree taking into account tourist's remaining stamina as a constraint.

In general, users do not always select the shortest or the cheapest route during sightseeing tour because their main purpose is to get satisfaction/experience through sightseeing activities. Whereas, they do not always maximize the satisfaction since their resource (money, time and stamina) is limited. Hence, for the move between PoIs, we must consider the trade-off between the satisfaction of the user and the time, money and stamina as user's resource. For the selection of PoI, we must also compare multiple PoIs in terms of the trade-off between the satisfaction and the resource consumption. As such, ATPS can be formulated as a multi-objective optimization problem with several independent factors.

In this paper, we formulate the sightseeing tour recommendation as a multi-objective optimization problem with money, time and stamina consumption during a tour and satisfaction degree obtained by the tour as independent variables. Since this problem is NP-hard, we propose a heuristic algorithm to quickly obtain semi-pareto optimal solutions based on genetic algorithm NSGA-II [3]. We applied the proposed method to planning tours targeting 30 tourist spots in Higashiyama-area of Kyoto, Japan. As a result, we confirmed that our algorithm could output semi-pareto optimal solutions in reasonable time that could be used for decision making under trade-off.

The remainder of the paper is structured as follows: Section 2 overviews the existing studies related to our proposal. Section 3 provides the formulation of our target problem. Section 4 describes the proposed algorithm based on GA. Section 5 provides the experimental results to evaluate our method and finally Section 6 concludes the paper.

2. Related Work

Most of the existing tour navigation systems recommend PoIs with high average review scores and/or according to the user's preference [4,5,6]. These studies focus on the improvement of user satisfaction and recommend a single PoI using a kind of filtering systems, but they do not consider the whole tour planning including routes and other factors related to tourism.

Some studies support the whole tour planning by connecting multiple PoIs. Kurata et al. [7,8] created a web-based interactive tour planning service called CT-Planner. This service plans and recommends a tour route while analyzing the user's preference.

There are some studies that add other factors related to tourism such as money, time and stamina as constraints [9,10,11,12]. For example, Wu et al. [2] considered the stamina as a constraint. However, these existing studies focus only on the satisfaction of users as the main factor and do not consider the trade-off between the satisfaction and other factors such as money, time, and stamina.

There have been proposed some methods that consider the trade-off with other factors. Shiraishi et al. [1] proposed a method which recommends tour routes considering the trade-off between satisfaction and time. Tamashiro et al. [13] defined multi-objective optimal routing problems for sightseeing by extending the optimal routing problem and considered a trade-off between the value of the tour and the required money. These studies, however, consider only a single factor conflicting the satisfaction and multiple conflicting factors are not considered.

Users take into account not only satisfaction/experience by the tour but also a balance between it and other factors such as money, time and stamina during the tour. Therefore tour planners must consider/compare multiple PoIs in terms of the trade-off between the satisfaction and the resource consumption.

In order to help users make decision for a good sightseeing tour, we model this problem as a multi-objective optimization method with money, time and stamina consumption of a tourist and satisfaction degree obtained as independent variables. PSO (Particle Swarm Optimization) and MOGA (Multi-Objective Genetic Algorithm) are popular methods for solving multi-objective optimization problems. Some algorithms for MOGA have been proposed so far [14,3]. Among them NSGA-II is one of the most popular algorithms which can derive diverse solutions.

The final goal of this work is to present multiple diverse tours to users. For this purpose, the diversity of solutions is very important. We consider three of factors: money, time and stamina as resources retained by users, and present diverse solutions to users by solving the tour search problem as a multi-objective optimization problem taking into account trade-off between these resources and satisfaction.

3. Diverse Tour Routes Search Problem

3.1. Problem

We assume that user resources consumed by sightseeing activities and movements between PoIs are (1) stamina, (2) time and (3) money. Moreover, to simplify the problem, the tourist's purpose of the sightseeing tour is only obtaining satisfaction by the sightseeing.

Our target problem is a multi-objective optimization problem to derive tour plans that have higher satisfaction with smaller consumption of resources consisting of stamina, time and money.

The values of four variables corresponding to stamina, time, money and satisfaction vary depending on the initial values assigned to these variables and increase/decrease of them at each PoI and each move included in the tour plan under consideration.

3.2. Problem Formulation

Let $\boldsymbol{X} = \{x_0, ..., x_n\}$ denote the set of tourist spots (PoIs). Let $\boldsymbol{r_n} = (m_n, t_n, s_n)^T$ denote the vector of the values of the remaining resources after enjoying sightseeing at each of n spots. Here, m_i, t_i and s_i denote the remaining amounts of money, time and stamina, respectively after enjoying i-th visiting spot in $\boldsymbol{X}$. Let $\boldsymbol{r_0} = (m_0, t_0, s_0)^T$ denote the vector of initial values assigned to resource variables, where m_0, t_0 and s_0 are initial values and correspond to the user's budget, available time for the tour and the initial stamina, respectively. We assume that the user will set these values manually. $\boldsymbol{r_n}$ can be represented by Eq. (1).

$$\boldsymbol{r_n} = \boldsymbol{r_0} - \sum_{i=0}^{n-1} [\boldsymbol{CR}(x_i) + \boldsymbol{moveCR}(x_i, x_{i+1})] - \boldsymbol{CR}(x_n) \tag{1}$$

where

$$\boldsymbol{r_n} = (m_n, t_n, s_n)^T$$
$$\boldsymbol{CR}(x) = (CM_x, CT_x, CS_x)^T$$
$$\boldsymbol{moveCR}(x, x') = (moveCM_{x,x'}, moveCT_{x,x'}, moveCS_{x,x'})^T$$

Here, $\boldsymbol{CR}(x)$ denotes the resource consumption at spot x and $\boldsymbol{moveCR}(x, x')$ denotes the resource consumption while moving from spot x to x'. CM_x, CT_x and CS_x denote the consumption of money, time and stamina for enjoying sightseeing at spot x, respectively. $moveCM_{x,x'}$, $moveCT_{x,x'}$ and $moveCS_{x,x'}$ denote the consumption of money, time and stamina for moving from spot x to x', respectively. We assume that these are constant values given in advance.

We assume that satisfaction denoted by c is determined by the tour route, the stay time at each spot and the environmental condition at the spot and route. Then, we represent c by Eq. (2).

$$c(\boldsymbol{X}) = \sum_{i=0}^{n-1} [SAT(x_i) + moveSAT(moveCT(x_i), x_i, x_{i+1})] + SAT(x_n) \tag{2}$$

Here, $SAT(x)$ denotes satisfaction obtained at spot x, and $moveSAT(moveCT(x), x, x')$ denotes satisfaction obtained while moving from spot x to x'.

We assume that the satisfaction obtained is always a positive value. The objective of the problem is to maximize both remaining resources and satisfaction. Then, the objective function is represented by Eq. (3).

$$maximize\ m_n(\boldsymbol{X}), t_n(\boldsymbol{X}), s_n(\boldsymbol{X}), c(\boldsymbol{X}) \tag{3}$$

4. GA-based Algorithm to Derive Diverse Tour Routes

The problem in Sect. 3 is an NP hard problem since it implies the multi-objective knapsack problem (known as an NP-hard problem) as a special case, so we propose a heuristic algorithm to solve it in practical time. In this section, first, we describe the coding of solutions operated in our algorithm, then describe genetic operators including mutation and crossover used in the proposed algorithm.

4.1. Coding of solution candidates

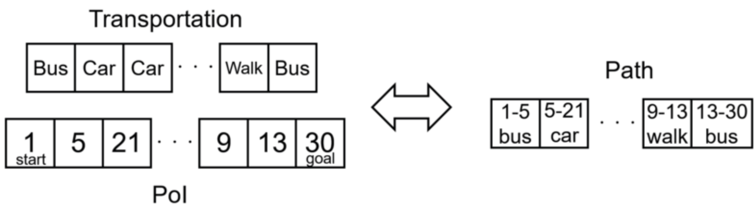

Figure 1. Representations of path data (right) and the corresponding Transportation type/ PoI data (left)

The solution candidates or simply solutions (i.e., chromosomes) used in our algorithm are path data shown in the right of Fig. 1 that contains a series of moves (i.e., transportation types) between two consecutive PoIs. These path data can be converted to PoI data and transportation type data shown in the left of Fig. 1. Genetic operations like Mutation and Crossover are applied to the PoI data and the transportation type data after converting from the path data.

4.2. Detailed Algorithm

The proposed algorithm consists of the following 6 steps and iterates Steps 2 to 6 for specified times (generations).

1. **Initialization**: First set the number of generations (iterations) to T, set the number of individuals (solutions) in initial population to N, and initialize the current generation number $t = 0$ and searching population $Q_0 = \emptyset$. Next, create initial population P_0 with randomly generated path data.
2. **Non-Dominated Sort**: Generate new population $R_t = P_t \cup Q_t$, and execute Non-Dominated sort for R_t and classify all elements of R_t by their rank i (i.e., the number of elements which dominate the element under consideration). To decide the rank of an element, all elements in the set are compared in terms of stamina, time, money and satisfaction which are calculated for each element (path) with equations (1) and (2). Then, the classified elements of R_t are added to $F_i\,(i = 0,..,n)$ according to their rank i.
3. **Crowding Sort**: Generate the next generation population P_{t+1} by adding $F_0, F_1, F_2, ...$ in this order while satisfying the condition $|P_{t+1}| \leq N$. In addition, if $|P_{t+1}| + |F_i| > N$, apply Crowding sort to F_i to add $N - |P_{t+1}|$ better (i.e., higher diversity) solutions in F_i to P_{t+1} . When the generation number t satisfies the condition $t + 1 = T$, the algorithm is terminated.
4. **Crowding Tournament**: Apply Crowding tournament based on stamina, time, money and satisfaction to solutions in P_{t+1}. It is applied to randomly selected $N/2$ pairs of 1-to-1 tournament to add $N/2$ better solutions to searching population Q_{t+1}.
5. **Crossover**: Randomly choose $N/4$ pairs of solutions from Q_{t+1} and apply Crossover to them (Fig. 4).

6. **Mutation**: Apply Decrement Mutation to randomly chosen $N/20$ solutions (Fig. 3), and apply Increment Mutation to other $N/20$ solutions (Fig. 2). Then, go to step 2 after incrementing t. $N/20$ was chosen based on the default mutation rate 0.1 used in NSGA-II.

4.3. Mutation

In our algorithm, we designed two Mutation algorithms: one is to randomly insert one PoI and the other is to randomly remove one PoI in a solution so that diverse solutions are kept in terms of the number of PoIs. One of the algorithms is selected to use at the probability of 0.5.

Increment Mutation: As shown in Fig. 2, first a PoI (PoI 7 in the fuigure) is randomly chosen from the set of all PoIs except for the ones already in the solution and a transportation type leaving from the added PoI (Car in the figure) is randomly chosen. Next, the insert position in a solution is randomly chosen, and both the chosen PoI and transportation type are inserted.

Decrement Mutation: This algorithm is applied only to a solution with length more than two. If the length is more than two, as shown in Fig. 3, the point to remove PoI is randomly chosen (PoI 21 was chosen in the figure) and the PoI and the transportation type leaving from the PoI (Car in the figure) are removed.

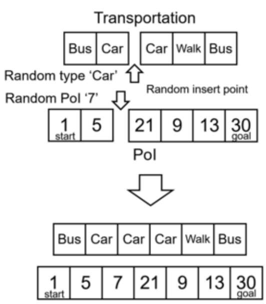

Figure 2. Increment Mutation

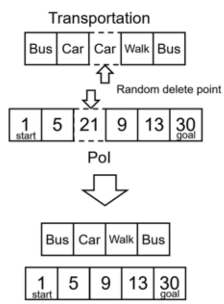

Figure 3. Decrement Mutation

4.4. Crossover

We employ single-point crossover where a single cut point in each of two parents is randomly selected and the left (right) part of the parent 1 and the right (left) part of the parent 2 are swapped to make new offspring, as shown in Fig. 4. However, simply using the single-point crossover for our algorithm, it is likely to generate lethal (invalid) solutions which include the same PoI multiple times. Hence, before concatenating the divided parts of the parent solutions, we try to reduce generation rate of such lethal solutions by randomly replacing either of the PoIs included in the left part of a parent and right part of another parent (middle of Fig. 4). For convenience, when we divide a parent solution into two parts, we remove the incoming transportation type in the right part and add a randomly selected transportation type when concatenating the parts as shown in Fig. 4.

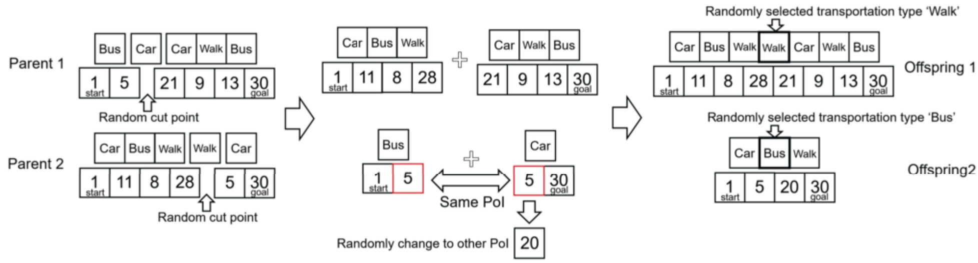

Figure 4. Single-Point Crossover

5. Evaluation Experiment

5.1. Experimental Environment

Table 1. Initial Resources

	Generation	Time(s)	Money(yen)	Stamina
low	500	15000	7500	1000
medium	500	30000	15000	2000
high	500	60000	30000	4000

We have implemented our algorithm in Python3 and executed the implemented algorithm on a PC with Intel Core i7-8550U 1.8 GHz, 16 GB RAM and Windows 10 Home OS. Further, we executed our algorithm for three different initial resource assignments shown in Table 1, to investigate the correlations between the derived solutions and the initial resources. In this experiment, we derived the information on the resource consumption on money and time for each path (solution) by using Google Map API. Since it is difficult to know actual stamina consumption and satisfaction got on a path, we set the imaginary values empirically. Modeling of stamina consumption and satisfaction acquisition (proposed in [15]) is our future work.

In this experiment, we targeted 30 PoIs in Higashiyama-area of Kyoto, Japan as shown in Fig. 11, and we determined the start and the goal points in advance. Moreover, because of the short distance between PoIs, we used car (taxi), bus and walking as types of transportation.

5.2. Relationship between Initial Resource and Solutions

Fig. 5 shows the scattered plot of initial solutions generated at random (at 0-th Generation). These solutions are paths consisting of 10 randomly selected PoIs including the start and goal points. Furthermore, Fig. 6, Fig. 7 and Fig. 8 plot 100 solutions that our algorithm calculated at 500-th generation with low, medium and high initial resources. Table 2 shows a part of the derived solutions, and values represent consumed/remaining resources and satisfaction value. This result supports that our proposed algorithm can derive diverse solutions considering the trade-offs between resources and satisfaction values.

Moreover Figs. 6–8 indicate relationships between money, time, stamina and satisfaction. Specifically, in Fig. 6, solutions are dense in the left-bottom area, because solutions are limited by low initial resources in this case. On the other hand, medium initial resource case (Fig. 7), solutions are more distributed in wider area than the low initial resource case (Fig. 6). Moreover, we found that the solutions calculated with high initial resources (Fig. 8) could fully search the solution space, because there were no solutions with the remaining resource equal to 0. The figures also suggest that there are positive correlations between resources, and negative correlations between resources and satisfactions.

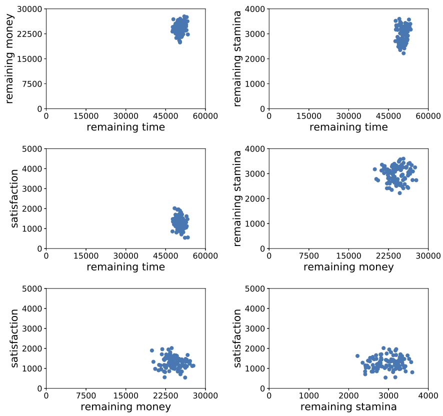

Figure 5. Random Solutions

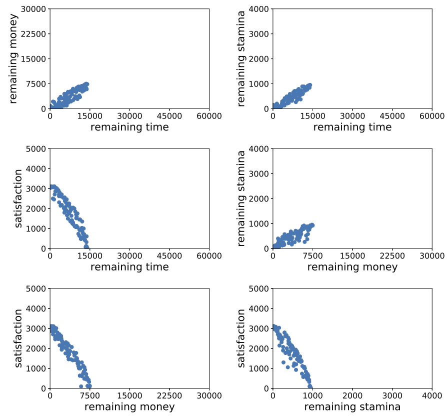

Figure 6. Solutions with low initial resources

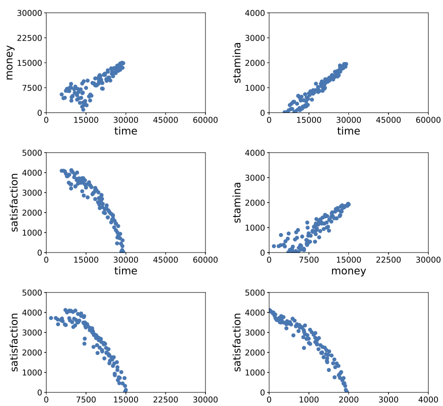

Figure 7. Solutions with medium initial resources

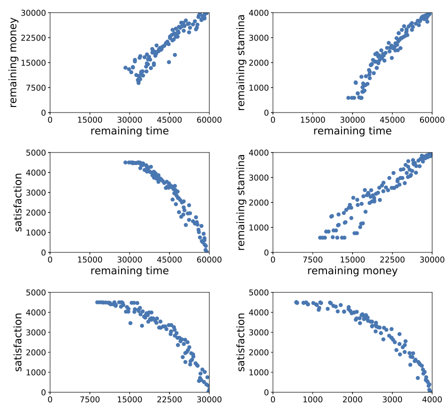

Figure 8. Solutions with high initial resources

Table 2. Example of solutions with low initial resources

POI and Transportation	**consumed (remaining) time (sec)**	**consumed (remaining) money (yen)**	**consumed (remaining) stamina**	**satisfaction**
[0, 12, 14, 10, 8, 1, 26, 2, 23, 27] [Walk, Bus, Bus, Walk, Bus, Bus, Walk, Bus, Car]	6591 (8409)	4040 (3460)	366(634)	1859
[0, 21, 10, 23, 19, 27] [Walk, Bus, Walk, Car, Bus]	11108 (3892)	6600 (900)	580 (420)	926
[0, 12, 10, 27] [Walk, Walk, Bus]	13076(1924)	7270 (230)	910 (90)	423

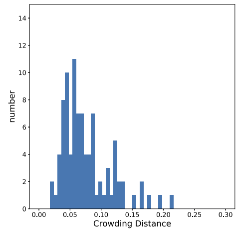

Figure 9. Crowding Distance (Random Solution)

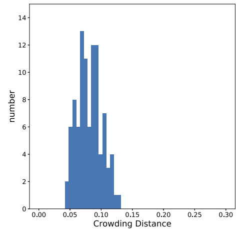

Figure 10. Crowding Distance (High initial resource)

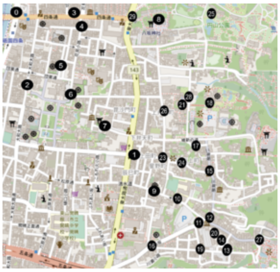

Figure 11. PoIs of Higashiyama-area

5.3. Diversity of Solutions

We used crowding distance as an indicator to evaluate diversity of solutions. The crowding distance $distance_i$ is defined as Eq. (4).

$$distance_i = \sum_{m=1}^{M} (E_m(i+1) - E_m(i-1))/(E_m(0) - E_m(n)) \quad i \in \{2, \ldots, n-1\} \tag{4}$$

In Eq. (4), n is the total number of solutions, and E_m, $m = [money, time, stamina, satisfaction]$ is sorted evaluation values in ascending order. The boundary solutions are defined as $distance_1 = distance_n = \infty$. The crowding distance is calculated as Manhattan Distance between the neighboring solutions, and crowding distances are equal in all neighboring pairs of solutions (except $distance_1$ and $distance_n$), if the distribution of solutions are completely uniform. When we compare the distribution of the crowding distances of randomly calculated solutions shown in Fig. 9 and those at 500-th generation with high initial resources shown in Fig. 10, we see that crowding distances of randomly calculated solutions have smaller crowding distances than that of high initial resource case. This result supports that solutions calculated with high initial resources have high diversity.

5.4. Computation Time

Table 3 shows the computation time in one generation for three different initial resources cases. When we use 500 generations, the total computation time will be 1700 to 2900 seconds. This time may look very long, but we still believe that it is feasible when planning a satisfactory tour, by reducing the number of generations and so on. From the table, we see that our algorithm takes more computation time in one generation in the case of more initial resources assigned. This is because the probability of lethal solution generation is higher with low initial resources, because the lethal solutions are ignored at the crossover and mutation steps, then computation time decreases.

6. Conclusion

In this paper, we proposed the NSGA-II based Multi-Objective Genetic Algorithm to search the semi-pareto optimal solutions of the tour route search problem considering

Table 3. Computation Time for One Generation (sec)

	N-D Sort	Crowding Sort	Tournament	Crossover	Mutation	Sum
low	0.158 ± 0.023	0.199 ± 0.027	0.023 ± 0.004	2.683 ± 0.282	0.346 ± 0.056	3.409 ± 0.350
middle	0.210 ± 0.041	0.261 ± 0.049	0.023 ± 0.004	3.596 ± 0.625	0.454 ± 0.101	4.544 ± 0.773
high	0.266 ± 0.082	0.325 ± 0.089	0.023 ± 0.005	4.610 ± 1.281	0.579 ± 0.179	5.803 ± 1.576

four conflicting factors: money, time, stamina resources and satisfaction degree. We evaluated the solutions calculated by our algorithm targeting the popular tourism area with 30 PoIs in Kyoto, Japan. As a result, we found that there are positive correlation between remaining resources and negative correlations between remaining resource and satisfaction. In addition, the solutions using high initial resources are more diverse and more uniformly distributed than random solutions. Moreover, the algorithm using high initial resources can search wider solution space but takes longer computation time. Our future work includes reduction of calculation time (especially for crossover operation), support of more transportation types as well as wider tourism areas with more PoIs and modeling of stamina consumption and satisfaction acquisition.

References

[1] Takayuki Shiraishi, Munenobu Nagata, Naoki Shibata, Yoshihiro Murata, Keiichi Yasumoto, and Minoru Ito. A personal navigation system with functions to compose tour schedules based on multiple conflicting criteria. *IPSJ Digital Courier*, 1:528–536, 2005.

[2] Wu Bing, Sun Weihua, Murata Yoshihiro, Yasumoto Keiichi, and Ito Minoru. Stamina-aware shightseeing tour scheduling method. *Information Processing Society of Japan*, 54(3):1141–1152, 2013.

[3] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2):182–197, April 2002.

[4] Zhiwen Yu, Yun Feng, Huang Xu, and Xingshe Zhou. Recommending travel packages based on mobile crowdsourced data. *IEEE communications magazine*, 52(8):56–62, 2014.

[5] Roberto Interdonato, Salvatore Romeo, Andrea Tagarelli, and George Karypis. A versatile graph-based approach to package recommendation. In *Tools with Artificial Intelligence (ICTAI), 2013 IEEE 25th International Conference on*, pages 857–864. IEEE, 2013.

[6] Xinhuan Chen, Yong Zhang, Pengfei Ma, Chao Li, and Chunxiao Xing. A package generation and recommendation framework based on travelogues. In *Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual*, volume 2, pages 692–701. IEEE, 2015.

[7] Yohei Kurata and Tatsunori Hara. CT-Planner4: Toward a More User-Friendly Interactive Day-Tour Planner. In Zheng Xiang and Iis Tussyadiah, editors, *Information and Communication Technologies in Tourism 2014*, pages 73–86. Springer International Publishing, 2013.

[8] Yohei Kurata, Yasutaka Shinagawa, and Tatsunori Hara. CT-Planner5: a Computer-Aided Tour Planning Service Which Profits Both Tourists and Destinations. In *ResearchGate*, September 2015.

[9] Joel P. Lucas, Nuno Luz, Mara N. Moreno, Ricardo Anacleto, Ana Almeida Figueiredo, and Constantino Martins. A hybrid recommendation approach for a tourism system. *Expert Systems with Applications*, 40(9):3532–3550, July 2013.

[10] L. Sebastia, I. Garcia, E. Onaindia, and C. Guzman. e-Tourism: A Tourist Recommendation and Planning Application. In *2008 20th IEEE International Conference on Tools with Artificial Intelligence*, volume 2, pages 89–96, November 2008.

[11] Francesco Ricci, Quang Nhat Nguyen, Olga Averjanova, Francesco Ricci, Quang Nhat Nguyen, and Olga Averjanova. Exploiting a Map-Based Interface in Conversational Recommender Systems for Mobile Travelers. January 2001.

[12] Pieter Vansteenwegen, Wouter Souffriau, Greet Vanden Berghe, and Dirk Van Oudheusden. The City Trip Planner: An expert system for tourists. *Expert Systems with Applications*, 38(6):6540–6546, 2011.

[13] Azusa Tamashiro, Shinji Aguni, and Morikazu Nakamura. Multi-Objective Optimal Routing Problem for Sightseeing and An MOGA for the Problem. *IEICE Technical Report*, 106, January 2007.

[14] David W. Corne, Nick R. Jerram, Joshua D. Knowles, and Martin J. Oates. PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization. In *Proceedings of the 3rd Annual Conference on Genetic and Evolutionary Computation*, GECCO'01, pages 283–290, San Francisco, CA, USA, 2001. Morgan Kaufmann Publishers Inc.

[15] Yuki Matsuda, Dmitrii Fedotov, Yuta Takahashi, Yutaka Arakawa, Keiichi Yasumoto, and Wolfgang Minker. Emotour: Estimating emotion and satisfaction of users based on behavioral cues and audiovisual data. *Sensors*, 18(11), 2018.