DEIM2022 D24-5

A Multi-agent Reinforcement Learning Approach Towards Congestion-aware Route Recommendation for Tourists

Wei-Kun KONG†,††, Shu-Yuan ZHENG††, Minh-Le NGUYEN†, and Qiang MA††

† School of Information Science, Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
†† Department of Social Informatics, Kyoto University,
Yoshida-Honmachi, Sakyo-ku, Kyoto 606-8501, Japan
E-mail: †{kong.diison,nguyenml}@jaist.ac.jp, ††caryzheng@db.soc.i.kyoto-u.ac.jp, †††qiang@i.kyoto-u.ac.jp

Abstract Route recommendation is an essential part of e-tourism, significantly promoting the booming of the tourism industry. Many models have been proposed to formalize the route recommendation problem and achieve promising results. Most of the existing models take only one tourist into account. They may raise the selfish issue, which causes the congestion problem of the optimal solution path when multiple users use the system simultaneously. This work defines a multi-user route recommendation problem and proposes a multi-agent reinforcement learning approach with a dynamic reward mechanism to tackle it. According to the preliminary experiment results, our method can significantly improve total tourists' rewards.

Key words Congestion-aware Route Recommendation, Multi-agent Reinforcement Learning

1 Introduction

Nowadays, tourism becomes not only an important choice for personal relaxation but also an essential industry component of the regional economy. According to the statistics from the World Tourism Organization (UNWTO) [1], the direct economic contribution of tourism is estimated at US$ 1.9 trillion in 2021 well below the pre-pandemic value of US$ 3.5 trillion. Even with the impact of the epidemic, the number of foreign tourists to Japan in 2020 still reached 3.9 billion, see Figure 1 [2]. The increasing tourists not only brings huge economic benefits but also brings congestion problems for the spots and the cities where they are located.

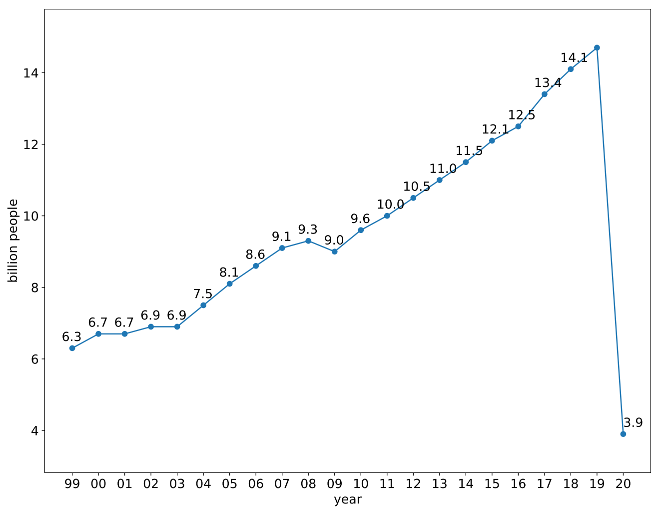

Figure 1 Trend in the number of international tourists

Many touring route recommendation models has been proposed [3–5], aiming to plan the optimized route for a single tourist and they can really achieve promising performance for single tourist's request. There is a problem that may put the tourists into a prisoner's dilemma [6]: The touring route recommendation model recommends the optimized route to every single tourist based on the static information. If there are enough tourists who have the same request, and they get the same recommended route from the touring route recommendation model, it will inevitably bring great pressure to both of the spots and the traffic, and make tourists themselves stacked into congestion.

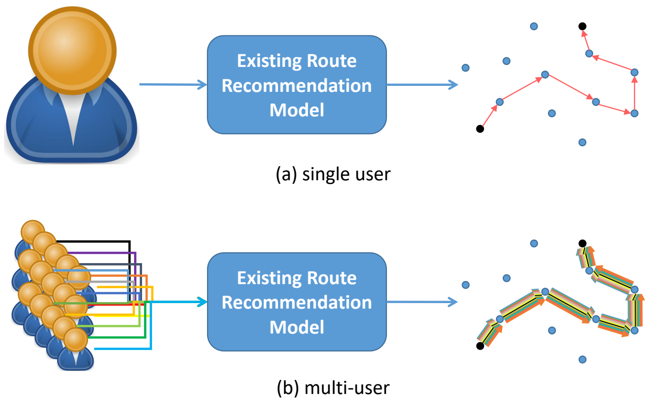

Figure 2 The existing models cannot handle the multi-user situation.

Since the existing tasks [7–13] are designed for single tourist, it is logical that they did not introduce a dynamic reward function based on congestion. According to [14], the congestion of spots in this period has a significant negative impact on the number of tourists in the next period, which may be due to the decline in tourist experience caused by the congestion. There is an illustration describing the problem shown in Figure 2.

For tourists, each of them is selfish and the only thing they concern is how to find a route that meets their own conditions and can maximize the gains for themselves. On the other hand, for local governments, they hope to maximize the sum of all tourists' revenue to promote the development of local tourism. Under normal circumstances, the aforementioned two targets are not contradictory, but when many travelers cannot communicate with each other when they making decisions and only consider their own gains, it may cause a situation where both tourists and local government suffer: though both tourists and local government hope to get high touring gains, and the recommendation model also recommends routes for tourists based on their requests, but actually both tourists and local government can only get poor gains because of congestion.

In this paper, we introduced a novel congestion-aware touring route recommendation task. In most of the existing route recommendation tasks' settings, there is only single tourist and no capacity limitations for spots, which can not meet the real sightseeing circumstances. The congestion-aware touring route recommendation task aims to fill the aforementioned gap. Several tourists start at their start location, Each of them must visit the start spot first and the end spot last, and try to visit as many spots as possible within a prescribed time budget. The spots are limited in capacity and congestion of the spot affects the reward of visiting the spot. The aim of the congestion-aware touring route recommendation problem is to maximize the total reward of all tourists. We denote the **C**ongestion-aware **T**ouring **R**oute **R**ecommendation problem by CTRR.

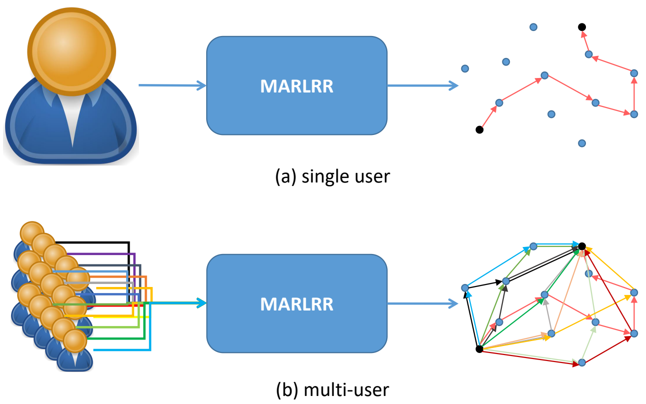

Figure 3 MARLRR can recommend various paths for many users, avoiding congestion.

As the first step to congestion-aware touring route recommendation, we propose MARLRR model, a **M**ulti-**A**gent **R**einforcement **L**earning method for touring **R**oute **R**ecommendation. We evaluate the proposed model using the total static reward and the total dynamic reward of all the tourists. Our model improves the total dynamic reward by over 50%, at the cost of less than 5% loss of on total static reward. Figure 3 illustrates the effectiveness of the proposed model in avoiding crowding.

We proposed a non-linear reward-congestion function according to a tourism economic model, which is able to reflect the function between tourist gains and possible spot congestion well when tourists visiting attractions.

We combined simulated environments with real-world statistics during training and testing, which significantly improves the applicability of our proposed model.

2 Related Works

The orienteering problem(OP) [7] is defined as follows: Given n nodes in the Euclidean plane each with a score $s(i) \geqq 0$ [note that $s(1) = s(n) = 0$], find a route of maximum score through these nodes beginning at 1 and ending at n of length (or duration) no greater than the time budget T_{max}.

The team orienteering problem(TOP) [9] extends the single-competitor version of OP to a multi-competitor version. A team consisting of several competitors starts at the same point. Each member of the team tries to visit as many control points as possible within a prescribed time limit and then ends at the finish point. Once a team member visits a point and gets the reward, other team members can not be awarded anymore for visiting this point. Each member of a team has to select a subset of control points to visit so that there is minimal overlap in the points visited by each member of the team, the time limit is not violated, and the total team score is maximized. But TOP is not like the ordinary multi-agent task. The relationship between a competitor and other competitors in TOP is more like the relationship between the competitor and its avatars. There is only cooperation and no competition between the competitors. They work together to complete the work of the single competitor in the OP.

The vehicle routing problem(VRP) [11]: Let $G = (V, A)$ be a directed graph where $V = \{0, \ldots, n\}$ is the vertex set and $A = \{(i, j) : i, j \in V, i \neq j\}$ is the arc set. Vertex 0 represents the depot whereas the remaining vertices correspond to customers. A fleet of m identical vehicles of capacity Q is based at the depot. The fleet size is given a priori or is a decision variable. Each customer i has a nonnegative demand q_i.

The diverse profit variants of the classic OP change the united fixed profits of spots to different values. The orienteering problem with variable profits(OPVP) [15]: The underlying assumption is that the collection of scores at a particular node require either a number of discrete passes or a continuous amount of time to be spent at that node. The collected score on node i depends on an associated collection parameter $\alpha_i \in [0, 1]$. Both proposed discrete and continuous models [15] are formulated as a linear integer programming model and a non-linear integer programming model, respectively. It is shown that the discrete model can be solved for instances with up to 200 nodes within 2 hours of computational time. On the other hand, the continuous model requires more computation time, for instances with 75 nodes.

The team orienteering problem with decreasing profits(DP-TOP) [16] extends TOP into version with decreasing profits. The profit of each node is a decreasing function of time. Due to the complexity of the problem, the Column Generation approach (CG) is introduced to reformulate and calculate the lower and upper bounds of the initial DP-TOP integer programming model. Evolutionary Local Search (ELS) is also proposed to solve the problem. TOP benchmark instances [9] are modified by adding the variable profits to the nodes. Almost all instances can be solved optimally by CG with the cost of computational time, while the ELS is less competitive in terms of the quality of solutions.

Multi-agent Orienteering Problem (MOP) is a multi-agent planning problem where individual agents are self-interested and will interact with each other when they arrive at the same nodes simultaneously. [17] studies the MOP with Time-dependent Capacity Constraints (MOPTCC). Due to the capacity constraint, each node can only receive a limited number of agents at the same time. If more agents are present, all agents will have to wait due to some extra queueing time. Therefore, the main focus is to identify a Nash equilibrium where individual agents cannot improve their current utilities by deviation. The problem is formulated as an integer programming model and a game-theoretic formulation. They propose two solution approaches: a centralized approach with Integer Linear Programming (ILP) that computes the exact global solution and a variant of the Sampled Fictitious Play (SFP) algorithm [18] that efficiently identifies equilibrium solutions. However, the first approach does not scale well and can only solve very small instances. The computational experiments show the ability of finding the equilibrium solutions in randomly generated instances.

The selfish OP (SeOP) [19] models the problem of crowd congestion at certain venues as a variant of the OP, whose main issue is how to provide route guidance to multiple selfish users (with budget constraints) moving through the venue simultaneously. SeOP combines OP with Selfish Routing (SR), which is a game between selfish agents looking for minimum latency paths from source to destination along edges of a network available to all agents. Thus, SeOP is a variant of MOP, where agents have selfish interests and individual budget constraints. As with Selfish Routing, Nash Equilibrium as the solution concept in solving SeOP is employed. [19] proposed DIRECT, an incremental and iterative master-slave decomposition approach to compute an approximate equilibrium solution. Similar to existing flow based approaches, DIRECT is scale-invariant in the number of agents. A theoretical discussion of the approximation quality and experimental results clearly show that the non-pairwise formulation achieves the same solution quality as the pairwise one using a fraction of the number of constraints and the master-slave decomposition achieves solutions with an adjustable approximation gap using a fraction of the full path set.

There is no existing model considers the impact to tourists' experience caused by the congestion of spots and assumes that the profits of tourists visiting spots are related to the degree of congestion of spots, which can better conform to reality. Our proposed congestion-aware touring route recommendation task fills this gap.

3 Methodology

3.1 Preliminary

We give the definition of the problem of touring route recommendation with dynamic reward as Definition 1 and the meanings of the symbols in Definition 1 is shown in Table 1:

Definition 1. Given a set of N spots:

$$\mathcal{S} = \{s_1, \cdots, s_N\}, \text{ where } s_n := \{loc_n,\ tt_n,\ c_n,\ num_n,\ r_n^{max}, r_n^{min}\}.$$

A set of M agents:

$$\mathcal{A} = \{a_1, \cdots, a_M\}, \text{ where } a_m := \{\mathcal{T}_m,\ loc_m^{start},\ ss_m \in S,\ se_m \in S\}$$

The commuting time cost of agent a_m between spot s_i and spot s_j is defined as:

$$t_c^{ijm} = t_c(loc_i, loc_j) = \frac{\sqrt{\left(loc_i^x - loc_j^x\right)^2 + \left(loc_i^y - loc_j^y\right)^2}}{v_m}$$

A tourist a_m visits a spot s_n to get a reward r_n^m, which varies based on the congestion of s_n. Every agent plans a route to visit and get the rewards. The aim of this task is to maximize the total reward $\mathcal{R}$ within time budget constraints:

$$\max \mathcal{R} = \sum_m \sum_n r_n^m \cdot \mathbf{1}_{\mathcal{S}^m}(s_n^m), \tag{1}$$

$$\text{s.t. } \mathbf{1}_{\mathcal{S}^m}(s_n^m) := \begin{cases} 1 & \text{if } s_n^m \in \mathcal{S}^m , \\ 0 & \text{if } s_n^m \notin \mathcal{S}^m . \end{cases} \tag{2}$$

$$\mathcal{S}^m := (s_0^m, ..., s_k^m),\ s_i^m \in \mathcal{S} \tag{3}$$

$$s_0^m \equiv ss_m,\ s_k^m \equiv se_m \tag{4}$$

$$\{s_n^m\} \bigcap \complement_{\mathcal{S}^m} \{s_n^m\} = \varnothing \tag{5}$$

$$r_n^m = g(num_n, c_m, r_n^{min}, r_n^{max}) \tag{6}$$

$$t_c\left(loc_n^{start}, loc_0\right) + \sum_k t_c\left(loc_k, loc_{k+1}\right) + \sum_k tt_k \leq \mathcal{T}^m, s_k, s_{k+1} \in \mathcal{S}^m \tag{7}$$

The equation Eqn. 1 ~ Eqn. 7 works as follows:

- Eqn.1 is the objective function, which is to maximize the total reward of all tourists.
- Eqn.2 indicates whether the tourist a_m has visited the spot s_n.
- Eqn.3 to ensure that the tourist visits a subset of $\mathcal{S}$.
- Eqn.4 to ensure that the tourist visits ss_m at the first and visits se_m at the last.
- Eqn.5 ensures that for each agent a_m, each spot can be visited at most once.

Table 1 The meanings of the symbols in Definition 1.

Symbol	Meaning
loc_n	the location of spot s_n.
tt_n	the time that the tourist needs to spend on touring the spot s_n.
c_n	the capacity of spot s_n.
num_n	the number of tourists who is touring the spot s_n.
r_n^{max}	the maximum reward for touring the spot s_n.
r_n^{min}	the minimum reward for touring the spot s_n.
$\mathcal{T}_m$	the time budget of the tourist a_m.
loc_m^{start}	the location of the tourist a_m when he starts touring.
ss_m	the spot the tourist a_m must visit at the first.
se_m	the spot the tourist a_m must visit at the last.

- Eqn.6 defines the general form of the reward of touring a spot s_n, which is based on the number of tourists, the capacity, the maximum reward and the minimum reward of spot s_n.
- Eqn.7 ensures that the total time cost of commuting and touring spots should not exceed the time budget of the tourist a_m.

3.2 Model Architecture

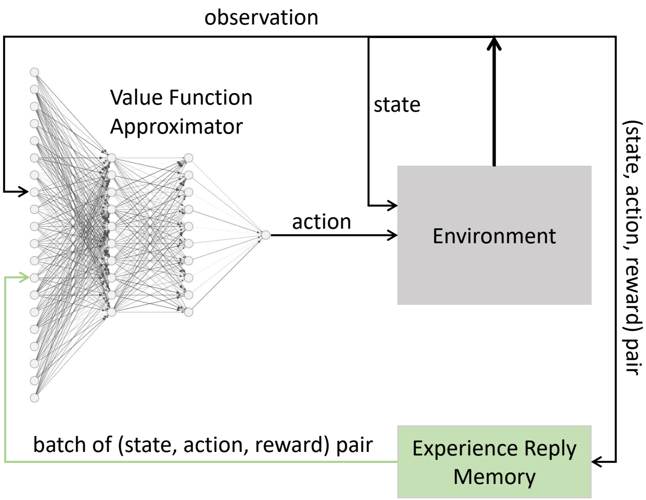

Figure 4 The architecture of MARLRR.

Algorithm 1: Overview of the proposed model

Data: tourist statistics of spots

Result: a well-trained route recommendation model

```
initialize env, qdn, replay_buffer;
observation, touristId = env.reset();
while i in range(episode) do
    while not done do
        action = dqn(observation);
        observation, touristId, reward, done = env.step(touristId,
          action);
        replay_buffer.add(observation, action, reward);
    end
end
qdn.train(replay_buffer);
```

In this section, we describe our proposed model and describe its use in programming a sub-optimal route for tourists given the statistics of spots. Our model can be roughly divided into two components: the value function approximator and the environment. The first module is a 3-layer feed-forward neural network, which learns how to generate an action according to the observation. The second module is a simulator of the real touring environment, which contains spots, tourists, constraints, and the rules of how the tourists get rewards. Figure 4 and Algorithm 1 show the overall architecture of our model. We learn the parameters of the value function approximator using the in-game reward feedback.

$$
\text{reward} = max\left(r, r_n^{min}\right), \tag{8}
$$
$$
r = \begin{cases} r_n^{max} \cdot \cos\left(\frac{num_n}{2c_n}\pi\right), & 0 \leqq num_n < 2c_n \\ r_n^{min}, & 2c_n \leqq num_n \end{cases}
$$

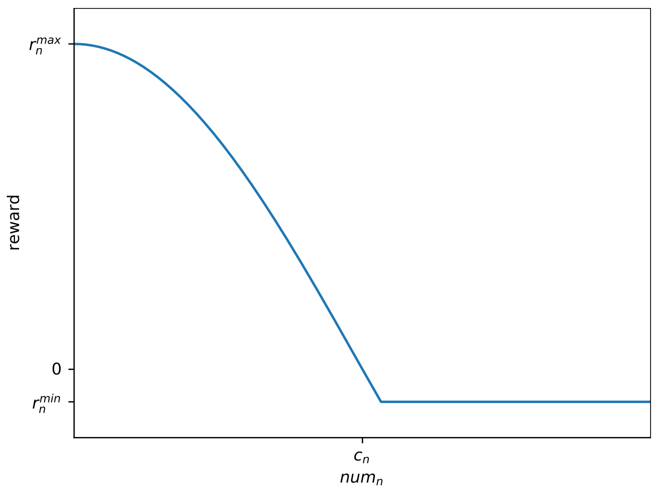

Figure 5 An illustration of Eqn.8.

The environment contains information on 72 spots in Kyoto, a randomly initialized group of tourists, and incentive strategies for tourists visiting the spots. We adopt Eqn.8 as the basic reward function, which goes like Figure 5. The basic assumption is that the more the number of tourists in the spot, the greater the expectation of congestion in the spot, and the rate at which the expectation of congestion will increase also increases as the number of tourists increases. The reward policy of our environment has been shown in Algorithm 2.

3.3 Real-time Statistics of the External Environment

The real-time statistics information of the external environment has been taken from [20], including geographic location, the number of tourists in spots at different time periods, and the score of spots at different time periods. The scores of spots are calculated from the average scores of photos uploaded by Flickr users, and the number of people at different spots in different time periods is counted by the number of Flickr users who posted photos at the

Algorithm 2: Reward Policy

Data: tourist statistics of spots

Result: a list of rewards of every spot

```
initialize rewardList;
Input: tList, seSet, tId, action, p, q
reward=max(cos(Num_j/(2C_j) π) R_j^max, R_j^min);
if action==wait then
    reward=0
else
    if tList[tId].route==null then
        if action==tList[tId].startSpot then
            if reward > 0 then
                reward = p*reward
            end
        else
            reward = -| p*reward |
        end
    else
        if action in tList[tId].route or action==tList[tId].startSpot
          then
            reward=-| p*reward |
        else
            if action==tList[tId].endSpot then
                if reward > 0 then
                    reward = p*reward
                end
            else
                if action in seSet then
                    if reward > 0 then
                        reward = q*reward
                    end
                end
            end
        end
    end
end
```

where line 2 is $\text{reward}=max\left(\cos\left(\frac{Num_j}{2C_j}\pi\right)R_j^{max}, R_j^{min}\right)$;

spots. In some time periods, the number of photos posted by Flickr users in many spots is zero, so we exclude the data for these time periods. For the case where the number of people in the spot is 0, we adopt Laplace smoothing: add the minimum non-zero tourist number of people to all spots' tourist numbers. We update the spot's statistics after the tourists have performed a certain number of actions. Since we only obtained the available data in 7 time periods, we update the spot data cyclically.

4 Experiment and Result

4.1 Experiment Setting

We use the first set of external environment statistics to initialize 72 scenic spots, where the coordinate of the spots is the latitudes and longitudes, the r^{max} of the spot is its rating at this time, r^{min} is set to -1, and the capacity of the spot is set to 2 times of the number of people in the spot at this time. The time budget for each tourist is set to 3 seconds, and its speed is between 5 unit and 10 unit. The start spot and end spot are two randomly generated different spots. The randomly generated starting locations of all tourists satisfy the normal distribution whose average is the average of the horizontal and vertical coordinates of all scenic spots and variance is 1.

We also trained a baseline model MARLRR$_{static}$, which shares the same settings with our proposed model MARLRR except for the observation and reward function of the baseline is based on static reward during training. On evaluation, we record both total static reward and dynamic reward of all the tourists.

4.2 Evaluation Metrics

We adopt total reward(TR), total static reward(TSR) and Gini coefficient(gini) to measure the performance of the proposed model, which is shown in Eqn. 9, Eqn. 10 and Eqn. 11, respectively.

$$\text{TR} = \sum_m \sum_n r_n^m \cdot \mathbf{1}_{S^m}(s_n^m), \tag{9}$$

$$\text{TSR} = \sum_m \sum_n r_n^{max} \cdot \mathbf{1}_{S^m}(s_n^m), \tag{10}$$

$$\text{gini} = \frac{\sum_{k=1}^{M}\sum_{l=1}^{M} |r_k - r_l|}{2\sum_{k=1}^{M}\sum_{l=1}^{M} r_l} \tag{11}$$

where M is the number of tourists,

and r is the reward of a certain tourist.

4.3 Result

Since the tourists are randomly generated, we ran our proposed model and baseline ten times to eliminate the uncertainty introduced by random. The results are shown in Table 2. The statistics of dynamic reward per tourist of MARLRR and MARLRR$_{static}$ is shown in Figure 6. We can find that MARLRR is able to not only get better total rewards for all tourists, but get more balanced reward for every tourist.

The statistics of static dynamic reward per tourist of MARLRR and MARLRR$_{static}$ is shown in Figure 7. Although our model achieves a lower total static reward, this is because of the congestion of the spots. If we force a tourist to visit more spots regardless of the crowd, we can indeed improve the total static reward, but this is meaningless, because it will inevitably reduce the total dynamic return.

The statistics of the number of visits to the spots are shown in Figure 8. We can find that MARLRR is able to significantly reduce the crowding of popular spots and disperse tourists to different attractions, better balancing the benefits of attractions.

The routes recommended by MARLRR are shown in Figure 9. Compared to MARLRR$_{static}$, we can find that MARLRR is able to re commend more diverse routes to tourists.

5 Conclusion

In this paper, we introduced a novel congestion-aware touring

Table 2 Results of MARLRR and MARLRR$_{static}$

	MARLRR		MARLRR$_{static}$		MARLRR		MARLRR$_{static}$	
	TR*	gini	TR*	gini	TSR**	gini	TSR**	gini
1	5149.0	0.148	3553.6	0.291	11416.5	0.098	12009.7	0.082
2	5090.1	0.153	3402.9	0.279	11586.1	0.089	12054.1	0.083
3	5342.1	0.149	3351.7	0.287	11474.0	0.088	12200.4	0.082
4	5258.2	0.155	3574.5	0.250	11314.8	0.107	12159.4	0.085
5	5178.7	0.158	3412.1	0.278	11564.3	0.094	12125.8	0.089
6	5513.7	0.160	3239.9	0.271	11290.8	0.109	11904.3	0.094
7	5018.8	0.173	3514.5	0.275	11506.8	0.095	11919.9	0.086
8	5357.3	0.141	3312.7	0.286	11841.4	0.082	12298.1	0.082
9	5480.1	0.164	3293.9	0.290	11644.2	0.091	11759.1	0.101
10	5141.0	0.169	3707.8	0.253	11229.5	0.111	12151.8	0.074
avg	**5252.9**	**0.157**	3436.3	0.276	11486.8	0.096	**12058.2**	**0.085**

* Total Reward

** Total Static Reward

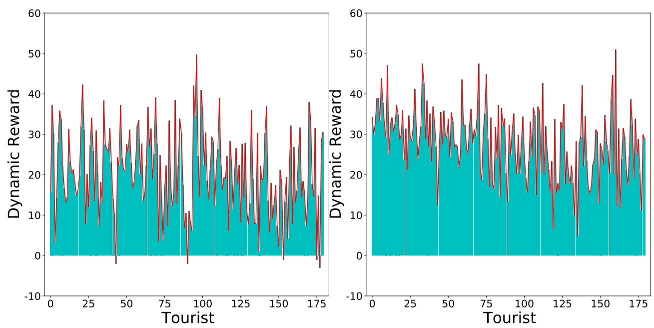

Figure 6 The Dynamic Reward (The left is the MARLRR$_{static}$, the right is MARLRR).

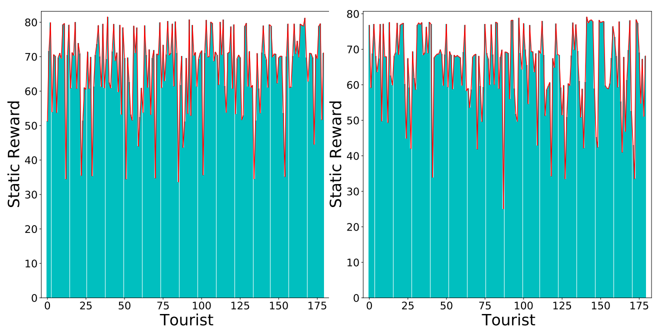

Figure 7 The Static Reward (The left is MARLRR$_{static}$, the right is MARLRR).

route recommendation task. We proposed an multi-agent reinforcement learning approach for solving the congestion-aware touring route recommendation task. We proposed a reward-congestion function according to a tourism economic model. We combined simulated environment with real-world statistics during training and testing, which improves the applicability of our proposed model. According to our experiment results, the proposed model can significantly improve the total dynamic rewards for all tourists, reduce the stress of popular spots, balance the number of visits to different spots and bring tourists more diverse recommendations.

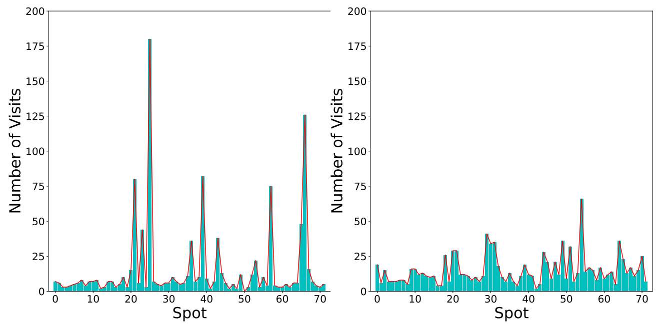

Figure 8 The Statistics of Visits of Spots (The left is MARLRR$_{static}$, the right is MARLRR).

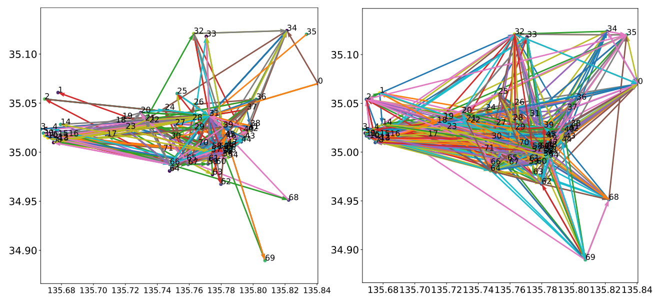

Figure 9 The Visualization of Tourist Trajectories (The left is MARLRR$_{static}$, the right is MARLRR).

Acknowledgments

This work was supported by JST SPRING, Grant Number JPMJSP2102. This work is partly supported by MIC SCOPE (201607008).

References

[1] “UNWTO world tourism barometer,” November 2021.

[2] J. T. Agency, “White paper on tourism in japan,2021 (summary),” tech. rep., Japan Ministry of Land, Infrastructure, Transport and Tourism, 2-1-3 Kasumigaseki, Chiyoda-ku, Tokyo 100-8918, 2021.

[3] D. Gavalas, C. Konstantopoulos, K. Mastakas, and G. Pantziou, “A survey on algorithmic approaches for solving tourist trip design problems,” *Journal of Heuristics*, vol. 20, pp. 291–328, Mar. 2014.

[4] S. N. Kumar and R. Panneerselvam, “A survey on the vehicle routing problem and its variants,” *Intelligent Information Management*, vol. 04, no. 03, pp. 66–74, 2012.

[5] A. Gunawan, H. C. Lau, and P. Vansteenwegen, “Orienteering problem: A survey of recent variants, solution approaches and applications,” *European Journal of Operational Research*, vol. 255, no. 2, pp. 315–332, 2016.

[6] A. Rapoport, A. M. Chammah, and C. J. Orwant, *Prisoner’s dilemma: A study in conflict and cooperation*, vol. 165. University of Michigan press, 1965.

[7] B. L. Golden, L. Levy, and R. Vohra, “The orienteering problem,” *Naval Research Logistics (NRL)*, vol. 34, no. 3, pp. 307–318, 1987.

[8] N. Christofides, “The vehicle routing problem,” *Revue française d’automatique, informatique, recherche opérationnelle. Recherche opérationnelle*, vol. 10, no. V1, pp. 55–70, 1976.

[9] I.-M. Chao, B. L. Golden, and E. A. Wasil, “The team orienteering problem,” *European Journal of Operational Research*, vol. 88, no. 3, pp. 464–474, 1996.

[10] T. İlhan, S. M. R. Iravani, and M. S. Daskin, “The orienteering problem with stochastic profits,” *IIE Transactions*, vol. 40, no. 4, pp. 406–

421, 2008.

[11] C. Verbeeck, K. Sörensen, E.-H. Aghezzaf, and P. Vansteenwegen, "A fast solution method for the time-dependent orienteering problem," *European Journal of Operational Research*, vol. 236, no. 2, pp. 419–432, 2014.

[12] Z. W. Geem, C.-L. Tseng, and Y. Park, "Harmony search for generalized orienteering problem: Best touring in china," in *Advances in Natural Computation* (L. Wang, K. Chen, and Y. S. Ong, eds.), (Berlin, Heidelberg), pp. 741–750, Springer Berlin Heidelberg, 2005.

[13] C. Archetti, D. Feillet, A. Hertz, and M. G. Speranza, "The capacitated team orienteering and profitable tour problems," *Journal of the Operational Research Society*, vol. 60, no. 6, pp. 831–842, 2009.

[14] I. P. Albaladejo and M. González-Martínez, "Congestion affecting the dynamic of tourism demand: Evidence from the most popular destinations in spain," *Current Issues in Tourism*, vol. 22, no. 13, pp. 1638–1652, 2019.

[15] G. Erdoğan and G. Laporte, "The orienteering problem with variable profits," *Networks*, vol. 61, no. 2, pp. 104–116, 2013.

[16] H. Murat Afsar and N. Labadie, "Team orienteering problem with decreasing profits," *Electronic Notes in Discrete Mathematics*, vol. 41, pp. 285–293, 2013.

[17] C. Chen, S.-F. Cheng, and H. C. Lau, "Multi-agent orienteering problem with time-dependent capacity constraints," *Web Intelligence and Agent Systems: An International Journal*, vol. 12, no. 4, pp. 347–358, 2014.

[18] T. J. Lambert, M. A. Epelman, and R. L. Smith, "A fictitious play approach to large-scale optimization," *Operations Research*, vol. 53, no. 3, pp. 477–489, 2005.

[19] P. Varakantham, H. Mostafa, N. Fu, and H. C. Lau, "Direct: A scalable approach for route guidance in selfish orienteering problems," in *AAMAS*, 2015.

[20] J. Xu, J. Sun, T. Li, and Q. Ma, "Kyoto sightseeing map 2.0 for user-experience oriented tourism," in *2021 IEEE 4th International Conference on Multimedia Information Processing and Retrieval (MIPR)*, pp. 239–242, 2021.