

Travel Itinerary Recommendations with Must-see Points-of-Interest

Kendall Taylor
RMIT University
School of Science
s9004570@student.rmit.edu.au

Kwan Hui Lim
The University of Melbourne
Computing and Information Systems
kwan.lim@unimelb.edu.au

Jeffrey Chan
RMIT University
School of Science
jeffrey.chan@rmit.edu.au

ABSTRACT

Travelling and touring are popular leisure activities enjoyed by millions of tourists around the world. However, the task of travel itinerary recommendation and planning is tedious and challenging for tourists, who are often unfamiliar with the various Points-of-Interest (POIs) in a city. Apart from identifying popular POIs, the tourist needs to construct a travel itinerary comprising a subset of these POIs, and to order these POIs as a sequence of visits that can be completed within his/her available touring time. For a more realistic itinerary, the tourist also has to account for travelling time between POIs and visiting times at individual POIs. Furthermore, this itinerary should incorporate tourist preferences such as desired starting and ending POIs (e.g., POIs that are near the tourist's hotel) and a subset of must-see POIs (e.g., popular POIs that a tourist must visit). We term this the TOURMUSTSEE problem, which is based on a variant of the Orienteering problem. Following which, we propose the LP+M algorithm for solving the TOURMUSTSEE problem as an Integer Linear Program (ILP). Using a Flickr dataset of POI visits in seven touristic cities, we compare LP+M against various ILP-based baselines, and the results show that LP+M recommends better travel itineraries in terms of POI popularity, total POIs visited, total touring time utilized and must-visit POI(s) inclusion.

CCS CONCEPTS

• **Information systems** → **Personalization; Recommender systems; Location based services; Data mining; Web applications;**

KEYWORDS

Tour Recommendations; Trip Planning; Recommendation Systems

ACM Reference Format:

Kendall Taylor, Kwan Hui Lim, and Jeffrey Chan. 2018. Travel Itinerary Recommendations with Must-see Points-of-Interest. In *WWW '18 Companion: The 2018 Web Conference Companion, April 23–27, 2018, Lyon, France*. ACM, New York, NY, USA, 8 pages. <https://doi.org/10.1145/3184558.3191558>

1 INTRODUCTION

Tour recommendation and itinerary¹ planning are challenging tasks for individual users. One main contributing factor is that the choice of Points-of-Interest (POIs) to visit in a city is often large, and

¹'Tour' and 'itinerary' are used interchangeably in this paper.

This paper is published under the Creative Commons Attribution 4.0 International (CC BY 4.0) license. Authors reserve their rights to disseminate the work on their personal and corporate Web sites with the appropriate attribution.

WWW '18 Companion, April 23–27, 2018, Lyon, France

© 2018 IW3C2 (International World Wide Web Conference Committee), published under Creative Commons CC BY 4.0 License.

ACM ISBN 978-1-4503-5640-4/18/04.

<https://doi.org/10.1145/3184558.3191558>

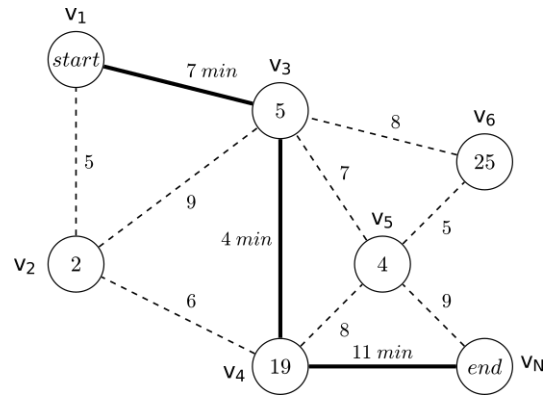


Figure 1: Example graph where vertices represent POIs (with a utility score s_i), and edges the time to travel between them ($t_{i,j}$). Not all edge and travel time values are shown.

varied. In addition, users often have limited time for touring, hence any recommended tour is also constrained by time, mainly for the consideration of travelling times between POIs and adequate POI visit durations. Furthermore, it is desirable that the distance and travel time between POIs should be reduced so as to allow the number of POIs and/or visitation time to be maximised. The desirability to include must-see POIs is often high on the user's agenda as there are often major attractions which are considered essential to visit, (such as the "Buda Castle" in Budapest or Toronto's Royal Ontario Museum).

In our work, recommended itineraries consist of a start and end point linked together by a path connecting a selection of POIs, including all or a subset of must-see POIs determined by the user. A POI can only be visited once, and the tour must include at least one POI other than the starting or ending POI. Links connecting POIs are associated with a travel time cost, which when summed, constrains the tour duration to a user-specified time budget. This constraint also ensures that only a subset of available POIs can be visited for a single itinerary. In order to represent user preference for particular types of POIs, each POI has a value representing its utility/score to the user. A good proposed itinerary is therefore one with a maximal sum of POI utility/score value but that can be completed within the allotted time budget. Figure 1 shows an example of this problem.

Our dataset of POIs is obtained from the Yahoo Flickr Dataset [25], and include POIs with high popularity and interest levels, where each POI is allocated a utility (or score) based on these criteria. Average POI visit duration is derived from the same data and the

geospatial distance between POIs is used to represent travel time (conversion based on walking speed).² The total time budget comprises the sum of the time taken to travel between POIs and the sum of average POI visit durations.

In this paper, we make a three-fold contribution to the field of itinerary recommendation, namely:

- (1) We introduce and formulate the TOURMUSTSEE problem, which is based on a variant of the Orienteering problem [26].
- (2) We propose the LP+M algorithm, which solves the TOURMUSTSEE problem as an Integer Linear Program.
- (3) Using a Flickr dataset of seven cities, we evaluate our proposed algorithm and demonstrate its good performance against various baselines.

The rest of the paper is organized as follows: Section 2 discusses related literature in the field of itinerary recommendation; Section 3 provides some background and a formal formulation of our TOURMUSTSEE problem; Section 4 gives an overview of our experimental design and evaluation metrics; Section 5 discusses the main results and findings; and Section 6 summarizes and concludes the paper.

2 RELATED WORK

In this section, we discuss some key works in the related areas of operations research, top-k POI recommendations and itinerary recommendations, before elaborating on the main differences of our research with these earlier works.

Operations Research. Itinerary recommendation has its origins in the traveling salesman problem (TSP) and its many variants typically used in the field of Operations Research. Of particular note are the TSP variants known as the Selective TSP [3], TSP with Profits [9], or the Orienteering Problem [11, 26, 27]. The main objective of these works are to plan for an itinerary comprising a set of POIs with the highest profit for a general tourist, while the Orienteering Problem has additional constraints of a specific starting and ending point. Integer Linear Program (ILP) formulations for this problem are widely used and have been successfully implemented to provide optimal solutions [9, 13]. The focus of these works are on the optimality of their approaches and their evaluation involves synthetic datasets where the optimal solutions are known.

Itinerary Recommendation using Social Media. In recent years, researchers have merged the works in Operations Research with the use of social media for itinerary recommendation [4, 6, 7, 18–20, 22, 23, 28]. Most of these works are formulated as ILPs based on either Orienteering Problem or TSP variants but instead of using synthetic datasets, these works evaluate their approaches using social media, such as Flickr geo-tagged photos that reflect real-life POI-based itineraries for several cities around the world. Similarly, there have been many applications for tour itinerary planning and recommendation [5, 33] that utilizes social media and techniques similar to these discussed works. The focus of such works is on recommending itineraries that are popular and reflect the real-life itineraries taken by these social media users.

Top-k POI Recommendation. The field of top-k POI recommendation is also closely related to tour itinerary recommendation

and we discuss key works in this research area [15, 24, 34, 35]. In these works, the main aim is to recommend a set of k POIs, where these POIs are ranked based on their relevance to a user. Such works typically utilize approaches based on various adaptations of matrix factorization or collaborative filtering algorithm. Similarly, there have been applications [1, 12] developed to recommend a list of individual POIs, which are not structured as an itinerary. The focus of these works are more on the relevance of individual POIs to users, rather than recommending an itinerary of connected POIs with the various user constraints.

Itinerary Recommendation with Mandatory POIs. The use of a set of POIs that must be visited is a further extension to the itinerary recommendation problem and has been explicitly addressed in only a few papers. Gendreau et al. [10], presented a branch-and-cut algorithm formulated as an ILP and tested against several heuristics. *Compulsory vertices* were incorporated into the model comprising a subset of total vertices which must be visited for a successful tour. Conclusions focused on the success of solving problems with a larger number of vertices rather than the impact of compulsory vertices.

Must-see (or "specified nodes") were incorporated in the the model used in an early paper by Laporte et al. [14]. This work used a branch-and-bound algorithm to find optimal tours where specified nodes were to be visited *exactly once*. Again the focus of the work was not the effect of compulsory nodes but the computational effort required for sub-tour elimination. Later work by Li et al. [16] presents an approach for the constrained orienteering model with compulsory nodes, however the formulation of the problem is based on network flow theory and differs markedly from that used here.

Main Differences with Our Work. While these earlier works offer interesting insights into the itinerary recommendation problem, our work differs from them in the following ways: (i) the works on *Operations Research* focus on the optimality of their solutions, typically utilizes synthetic datasets and do not consider must-visit POIs, while our work uses real-life POI itineraries mined from a Flickr geo-tagged photo dataset and consider must-visit POIs; (ii) the works on *Itinerary Recommendation using Social Media* utilizes a real-life dataset but does not consider must-visit POIs in their problem formulation, whereas our work considers this additional constraint; (iii) the works on *Top-k POI Recommendation* focus on recommending a ranked list of individual POIs, whereas our work requires the recommendation of relevant POIs, and additionally the planning of an itinerary through these POIs with a specific starting and ending POI that can be completed within a certain time; and (iv) the works on *Itinerary Recommendation with Mandatory POIs*, like works on *Operations Research*, is focused on solution optimality and uses synthetic datasets where the profit are randomly generated, while our work uses a real-life dataset where the POI profit reflect the popularity of POIs based on their number of visits. In addition, our formulation of the must-visit POIs allows for the flexibility to recommend itineraries that include a subset of must-visit POIs, instead of having to include all POIs like the earlier works.

3 BACKGROUND AND PROBLEM

In this section, we present some basic definitions used in our work and formulate the TOURMUSTSEE problem.

²Although we used walking speed and distance to represent travelling time, this representation can be easily extended to account for different transport modes, e.g., driving, cycling, train.

3.1 Problem definition

Our proposed TOURMUSTSEE problem of recommending an itinerary with must-see POIs is NP-hard [10, 16]. The complexity of TOURMUSTSEE is exponential as the set of POIs grow and the problem becomes intractable using a “brute-force” approach. This problem is similar to the *Travelling Salesman Problem with Profits* and the *Orienteering Problem*, where the objective is to maximise a total score (accumulated from visiting POIs), while keeping travel time between POIs under a fixed budget. The recommended itinerary begins at a specified starting POI and terminates at another.

This problem can be represented as a graph $G = \langle V, E \rangle$ where $N = |V|$ and $V = \{v_1, \dots, v_N\}$ the set of vertices (or POIs), and E is the set of connecting edges. Each edge connecting vertex i to j has a travel time cost and is known for all $v \in V$ and can be represented as $t_{i,j}$. The total time budget for the tour is T_{max} which limits the number of vertices in a tour. Each vertex also has a non-negative known score S_i for all $v_i \in V$ where $i \in \{1, \dots, N\}$.

We define an itinerary as a path between distinct starting and ending POIs, containing at least one other POI, where POIs are visited only once and sub-tours are excluded. Let P be a set of POIs with p_1 as the starting POI and p_N the destination POI, an itinerary I is a sequence of connected POIs such that $I = \{p_1, \dots, p_N\}$. We introduce a set M of *must-see* POIs where $M = \{m_1, \dots, m_L\}$ where $L \leq N$, ideally L is much smaller than N . An itinerary with must-see POIs I_M can subsequently be presented as a sequence $I_M = \{p_1, \dots, m_1, \dots, m_L, \dots, p_N\}$ where $p_1, p_N \notin \{m_1, \dots, m_L\}$.

3.2 Formulation

By applying a linear programming approach to the problem we can specify the generation of an itinerary *without* must-see POIs as the optimization of an objective function and adherence to various constraints. We first describe the objective function of maximizing the utility/score of a recommended itinerary to a user (Equation 1), which is formally represented as:

$$\text{Max} \sum_{i=2}^{N-1} \sum_{j=2}^N S_i x_{i,j} \quad (1)$$

where $x_{i,j} = \begin{cases} 1, & \text{if vertex } i \text{ is visited followed by } j \\ 0, & \text{otherwise} \end{cases}$

S_i represents the score for including p_i in a tour, where $i = 1, \dots, N$. The score is a known non-negative value that represents the popularity score of a POI based on frequency-based visit counts, which is derived from Flickr geo-tagged photos taken in close proximity to that POI [20].

The existing model I (that excludes must-see POIs), is subject to the following constraints:

- (1) A tour itinerary must begin at the chosen *start* POI and terminate at the chosen *end* POI (Equation 2).

$$\sum_{j=2}^N x_{1,j} = \sum_{i=1}^{N-1} x_{i,N} = 1 \quad (2)$$

This constraint allows the tourist to select starting and ending POIs based on their preferences, such as POIs that are conveniently located near their accomodation.

- (2) No POI can be visited more than once and all nodes in the path must be connected (Equation 3).

$$\sum_{i=1}^{N-1} x_{i,k} = \sum_{j=2}^N x_{k,j} \leq 1 \quad \forall k = 2, \dots, N-1 \quad (3)$$

As our TOURMUSTSEE problem is targetted at tourists, this constraint ensures that we do not recommend a tour itinerary that comprises multiple visits to the same POI or a tour itinerary that is not connected.

- (3) At least one additional POI (other than the start and ending points) must be included (Equation 4).

$$x_{i,j} = 0, \text{ where } i = p_1 \text{ and } j = p_N \quad (4)$$

This constraint ensures that our algorithm does not recommend a tour itinerary that is overly simplistic in the form of a direct path from the starting POI to ending POI, which could occur if the time budget is too small.

- (4) Total tour duration D cannot exceed the specified budget B (Equation 5).

$$\sum_{i=1}^{N-1} \sum_{j=2}^N D_{i,j} x_{i,j} \leq B, \quad \text{where } D_{i,j} = \text{TravelTime}(i,j) + \text{VisitDuration}(j) \quad (5)$$

$\text{TravelTime}(i,j)$ is the time taken to walk from p_i to p_j at a speed of 4kph.³ This is a known non-negative value providing a time approximation based on the geospatial distance between p_i and p_j in minutes. $\text{VistDuration}(j)$ is the average visit time (in minutes), identified for each $p \in P$, derived from the real-life itineraries of all users in the Flickr geo-tagged photos dataset. The budget B represents the maximum time allowed for the tour to finish and cannot be exceeded.

- (5) Sub-tours are to be eliminated (Equations 6 and 7).

$$2 \leq p_i \leq N, \quad \forall i = 2, \dots, N \quad (6)$$

$$p_i - p_j + 1 \leq (N-1)(i - x_{i,j}), \quad \forall i, j = 2, \dots, N \quad (7)$$

A sub-tour is a separate loop comprising POIs that do not include the starting and ending POIs, thus leading to a solution that comprises two disjoint itineraries. Thus the elimination of such sub-tours is important in the Orienteering problem, TSP and similarly our TOURMUSTSEE problem. However, eliminating sub-tours before optimizing the model is exponential in complexity. A common method used to remove sub-tours while avoiding excessive computation cost is the MTZ [21] approach, which we utilize in this work. This method examines an initially generated tour t for sub-tours and adds a constraint to the model for each sub-tour found. The model is then re-evaluated with the new constraints producing a new tour t' which is again evaluated for sub-tours.

³We used a walking speed of 4kph in this paper but this travelling speed can be generalized to other transport modes, e.g., faster speeds for cars. Also, we can use a time-dependent travel speed to represent different traffic conditions at different times, e.g., slower travelling during peak hours.

The iteration continues until a solution is found without sub-tours.

3.3 Must-see POI(s) constraint

In addition to the constraints above, we introduce a constraint that ensures a set of must-see POIs are visited in a successful solution. The set of must-see POIs M must be visited such that:

$$\sum_{i=1}^{N-1} x_{i,m} = \sum_{j=2}^N x_{m,j} = 1, \forall m \in M = \{m_1, \dots, m_L\} \text{ where } L = |M| \quad (8)$$

For each $m \in M$, $m \in P$ and $L \leq N$ where $N = |P|$. This constraint expresses that a valid tour must include a connection to, and from each node m_i in the set M . This constraint reflects the real-life consideration of tourists to include must-see POIs, which are often major attractions that are considered essential to visit, (such as the ‘‘Buda Castle’’ in Budapest or Toronto’s Royal Ontario Museum). This constraint can also be easily adapted to include a subset of the must-visit POIs, instead of all POIs like earlier works, by changing $L = |M|$ to $L \leq |M|$ in Equation 8.

4 EXPERIMENTAL DESIGN

In this section, we describe our experimental design, which includes a description of our dataset, baseline algorithms, evaluation methodology and metrics.

Table 1: POI data sets

Set ID	City	Connections	POIs
1	Budapest	1332	39
2	Edinburgh	650	26
3	Toronto	812	30
4	Vienna	756	29
5	Glasgow	600	29
6	Perth	462	25
7	Osaka	552	28

4.1 Data sets

Our experimental evaluation was performed using sets of POIs from seven cities, (see Table 1). These sets are derived from the Yahoo Flickr Creative Commons 100 Million Dataset (YFCC100M) which contains nearly 100 million photos, 69 million with annotations, and 48 million with geo-tags [25].

POIs were selected based on a list of popular POIs for each city from Wikipedia [29–32]. The Flickr geo-tagged photos are then mapped to each POI based on their proximity, thus allowing us to determine the popularity of each POI and the average visit duration at each POI. For more details on this dataset, please refer to Lim et al. [20]. The distance between POIs are geospatial measurements in meters, which we then convert to a travelling cost in terms of time in minutes (based on a walking speed of 4 kph). Table 2 provides a sample of the POI set for the city of Edinburgh.

Table 2: Example POI data set

Origin POI	Destination POI	Distance (meters)	Score/ Profit	Visit duration (minutes)
1	2	2236.79	83	25.32
1	3	795.75	859	47.6
1	4	237.16	895	8.67
1	5	1631.57	552	230.63

4.2 Proposed algorithm and baseline

The baseline for our evaluations is the tour T_{opt} , the optimal path found by implementing the above problem *without* the set M , of must-see POIs. If M is non-empty then the inclusion of an element m_1 in the resulting tour acts as a constraint on potential solutions. It is therefore expected that including must-see POIs will result in a reduction in overall tour score compared with T_{opt} .

Apart from our proposed LP+M algorithm, we also evaluate other baseline algorithms that are based on variations of ILP:

- (1) **LP.** Generates an itinerary *without* must-see POIs. This is the existing ILP model that generates optimal itineraries subject to the constraints outlined in section 3.2. above, and is used to generate the tour T_{opt} . These itineraries comprise a path from a starting to an ending POI where the collected POI score is maximised and the total tour time is limited by a given budget. Must-see POIs may be visited in the path however their inclusion is the product of an optimal tour rather than explicitly visiting them.
- (2) **LP+M (our proposed model).** Generates an itinerary *with* must-see POIs. This model builds on the LP model using the same objective function and constraints but adds a new constraint ensuring that must-see POIs are included in all solutions. This function models a typical itinerary situation where the user wants to visit several POIs *as a part of* a larger more general tour.
- (3) **MaxM.** Based on the approach used by Laporte et al.[14] that generates an itinerary with must-see POIs by allocating a large profit value to each must-see POIs. Combined with the LP model outlined above, successful solutions contain the must-see POIs as their inflated profits ensure their selection. This approach provides a reliable means of incorporating must-see POIs into an ILP model.
- (4) **GreedyM.** Generates an itinerary for the must-see POIs first, then generates an itinerary for the remaining POIs, and joins them together. This is a simple yet practical approach to building itineraries based on visiting compulsory POIs. As must-see POIs are the focus of the tour, a feasible option is to visit those POIs first and then use any remaining time budget to visit some of the remaining POIs. Both parts of the itinerary can be optimised using the LP approach and combined together.

The algorithms used for this work were implemented using the R statistical programming language, and the lp_solve library. The lp_solve package is a Mixed Integer Linear Programming (MILP) solver based on the simplex method, and branch and bound algorithm [2].

4.3 Experimental Setup

A total of 9,100 tours were generated to evaluate the effect of must-see POIs on itinerary planning and optimality. For each of the seven cities in the dataset, 300 random starting and ending POIs were generated and five sets of random must-see POIs were selected. The must-see POI sets include cardinalities from one to four POIs and are selected independently. Each set of must-see POIs is visited in its entirety.

For the LP algorithm, 700 tours are generated and the number of visited must-see POIs in each tour is recorded. For the LP+M, MaxM and GreedyM algorithms, 2,800 itineraries are generated for each, with 2,100 tours created for each of the four sets of must-see POIs. The cities used for this work contain varying numbers of POIs dispersed with urban areas of differing size and density. In order to provide tour time budgets which reflect these variations, each city’s budget was calculated using the formulation below:

$$[\mu(P_{travel}) + \mu(P_{visit})] \cdot q, \text{ where } q = L + 2 \quad (9)$$

The time budget for each city is the sum of the median travel time $\mu(P_{travel})$ in minutes and visit durations, $\mu(P_{visit})$ for each POI in the city.⁴ The result is multiplied by q which represents the minimum number of POIs required to visit the largest set of must-see POIs. In our case the largest set comprises four POIs, and with the inclusion of the start and end POIs $q = 6$.

Table 3: Number of LP tours (out of 100) which visited all or at least one must-see POIs. Higher values are better.

City	Visited all # must-see POIs				Visited at least one # must-see POIs			
	1	2	3	4	1	2	3	4
Budapest	26	10	2	0	26	49	62	65
Edinburgh	39	14	13	0	39	61	85	90
Toronto	31	12	6	0	31	57	77	85
Vienna	38	15	4	3	38	57	70	87
Glasgow	34	15	8	4	34	66	72	83
Perth	44	11	10	2	44	63	81	84
Osaka	41	19	11	4	41	63	81	84

4.4 Evaluation Metrics

To evaluate the performance of our algorithm against the various baselines, we utilize the following evaluation metrics:

- (1) **Inclusion of Must-see POIs.** Whether a recommended itinerary includes the set of must-see POIs.
- (2) **Tour Profit.** The total profit from all POIs in a recommended itinerary.
- (3) **POIs Visited.** The number of unique POIs in a recommended itinerary.
- (4) **Utilized Budget.** The total budget utilized for a recommended itinerary, i.e., the total cost needed to travel between POIs and visit durations at POIs.

⁴This definition of time budget realistically considers both travelling time and POI visiting durations, and the same time budget is used for our proposed algorithm and all baselines to ensure the same experimental setting for a fair evaluation. Although we can use other definitions of time budget, the general result trends will be the same as the same time budget is used for all algorithms.

Table 4: Number of successful tours (out of 100) which included must-see POIs. Higher values are better and the best performance among LP+M, GreedyM and GreedyM are in bold/blue.

City	LP+M # must-see POIs				MaxM # must-see POIs				GreedyM # must-see POIs			
	1	2	3	4	1	2	3	4	1	2	3	4
Budapest	79	58	39	21	74	53	37	19	77	59	39	17
Edinburgh	82	75	53	38	82	69	50	36	82	75	53	32
Toronto	77	43	27	4	76	40	26	4	75	42	20	4
Vienna	80	50	28	21	79	50	30	20	81	45	29	16
Glasgow	77	57	28	18	74	53	25	15	76	52	26	13
Perth	82	57	47	20	82	55	46	20	82	54	44	18
Osaka	88	82	66	49	87	76	65	45	91	80	62	39

Table 5: Average number of POIs visited including failed tours by algorithm, must-see set size and city. Higher values are better and the best performance among LP+M, GreedyM and GreedyM are in bold/blue.

City	LP	LP+M # must-see POIs				MaxM # must-see POIs				GreedyM # must-see POIs			
		1	2	3	4	1	2	3	4	1	2	3	4
Budapest	10.7	7.6	5.8	4.0	2.0	7.4	5.5	3.9	1.9	7.1	5.0	3.3	1.2
Edinburgh	10.2	8.6	7.9	5.4	4.0	8.6	7.4	5.2	3.8	8.2	7.1	4.8	2.8
Toronto	9.6	6.5	3.7	2.4	0.4	6.4	3.6	2.4	0.4	6.0	3.2	1.6	0.3
Vienna	10.0	7.9	5.1	2.7	1.9	7.9	5.1	3.0	1.9	7.4	4.1	2.3	1.3
Glasgow	10.2	8.2	6.1	3.1	2.0	7.9	5.8	2.9	1.8	7.6	4.9	2.5	1.3
Osaka	10.4	9.3	8.4	6.7	5.0	9.3	8.0	6.6	4.7	8.9	6.8	5.0	2.9
Perth	9.0	7.6	4.9	4.2	1.9	7.6	4.9	4.2	1.9	7.4	4.4	3.7	1.5

Metric 1 allows us to determine if the recommended tour itineraries fulfilled the constraint of visiting the entire set of must-see POIs (or at least one of the must-see POIs). Metrics 2 and 3 measure the total profit and number of visited POI in a recommended tour, which reflect the perceived utility value to a tourist, i.e., a tour with higher profit and more POIs is better. Metric 4 examines the total time budget used for a tour itinerary and a preferred tour is one that utilizes more time for visiting POI compared to travelling between POIs. These evaluation metrics and related variants are also commonly used in similar itinerary recommendation works [4, 6–8, 17].

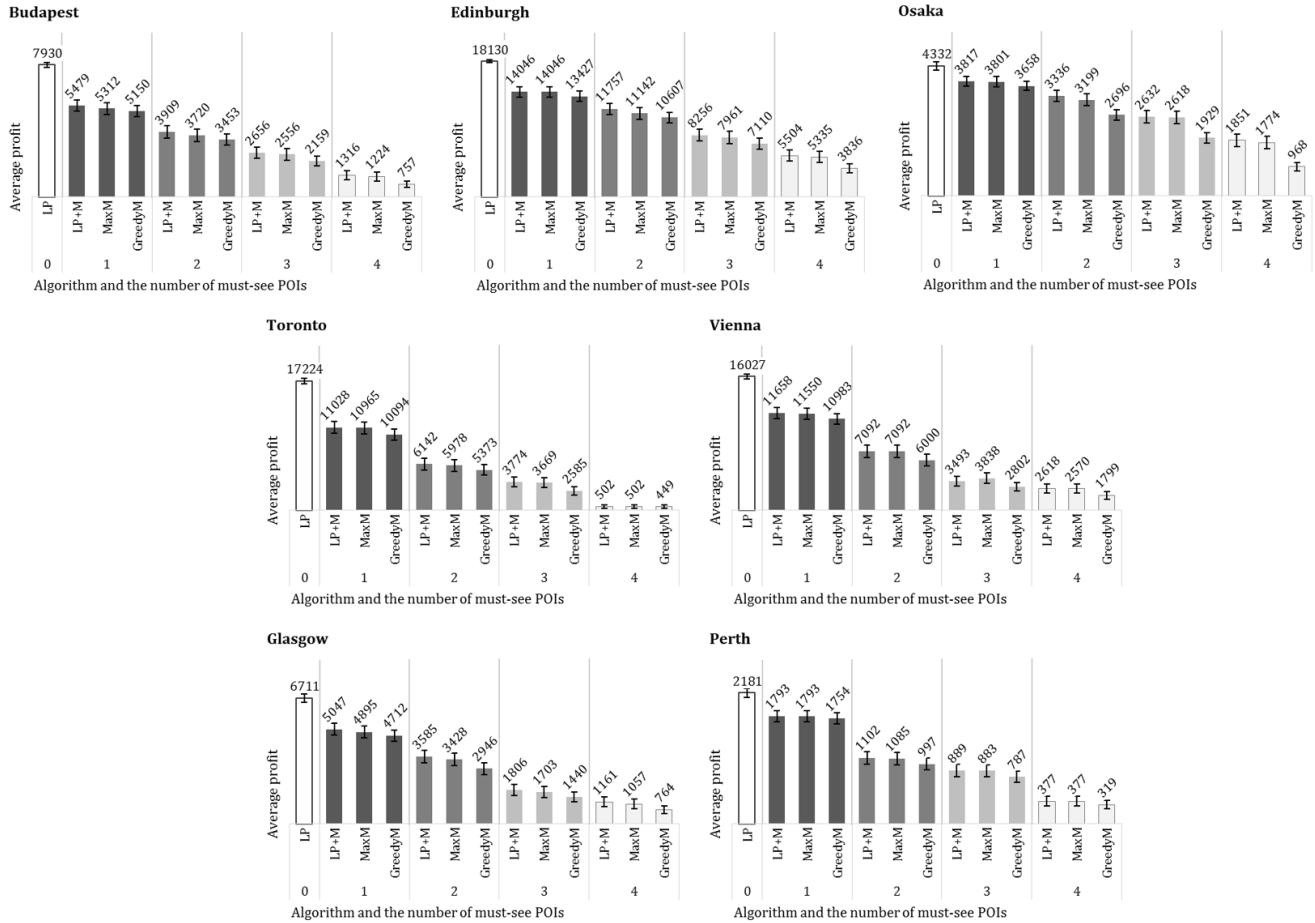
5 EXPERIMENTAL RESULTS

In this section, we present and discuss the experimental results in terms of inclusion of must-see POIs, total tour profit, total POIs visited and utilized budget.

5.1 Inclusion of Must-see POIs

The LP algorithm without must-see POI constraints provides the optimal tour T_{opt} and is the basis for comparisons with the other methods. The LP algorithm does not make any guarantees about visiting of any of the must-see POIs in a generated tour, and in 37% of cases did not visit a single must-see POI. Table 3 presents the number of tours generated by the LP method which successfully visited all POIs in the must-see sets. The incidence of such tours declined rapidly as the must-see set size increased, however the

Figure 2: Average tour profit by number of must-see POIs to visit. Higher values are better.



number of tours visiting at least one must-see POI increases as more options become available.

While nearly all tours generated by the LP method were successful, the number of successful itineraries produced when must-see POIs were included was below perfect. As must-see POIs are included as a constraint on the model, it is not surprising that neither LP+M, MaxM nor GreedyM could find solutions for all tours combinations.

Table 4 presents the number of successful tours found with LP+M, MaxM and GreedyM, along with the sizes of the must-see POI set. The success rate of each method is nearly identical and clearly shows the effect of increases in the number of must-see POIs to be visited. Interestingly, Toronto experienced a lower success rate for nearly all set sizes compared with the other cities even though it had the second highest number of POIs to select from.

5.2 Profit of Recommended Tours

The inclusion of the constraint to visit all must-see POIs reduces average tour profit, and is illustrated in Figure 2 where average profit values are lower than the optimum for the LP+M, MaxM and GreedyM algorithms. Figure 2 presents average profits and standard

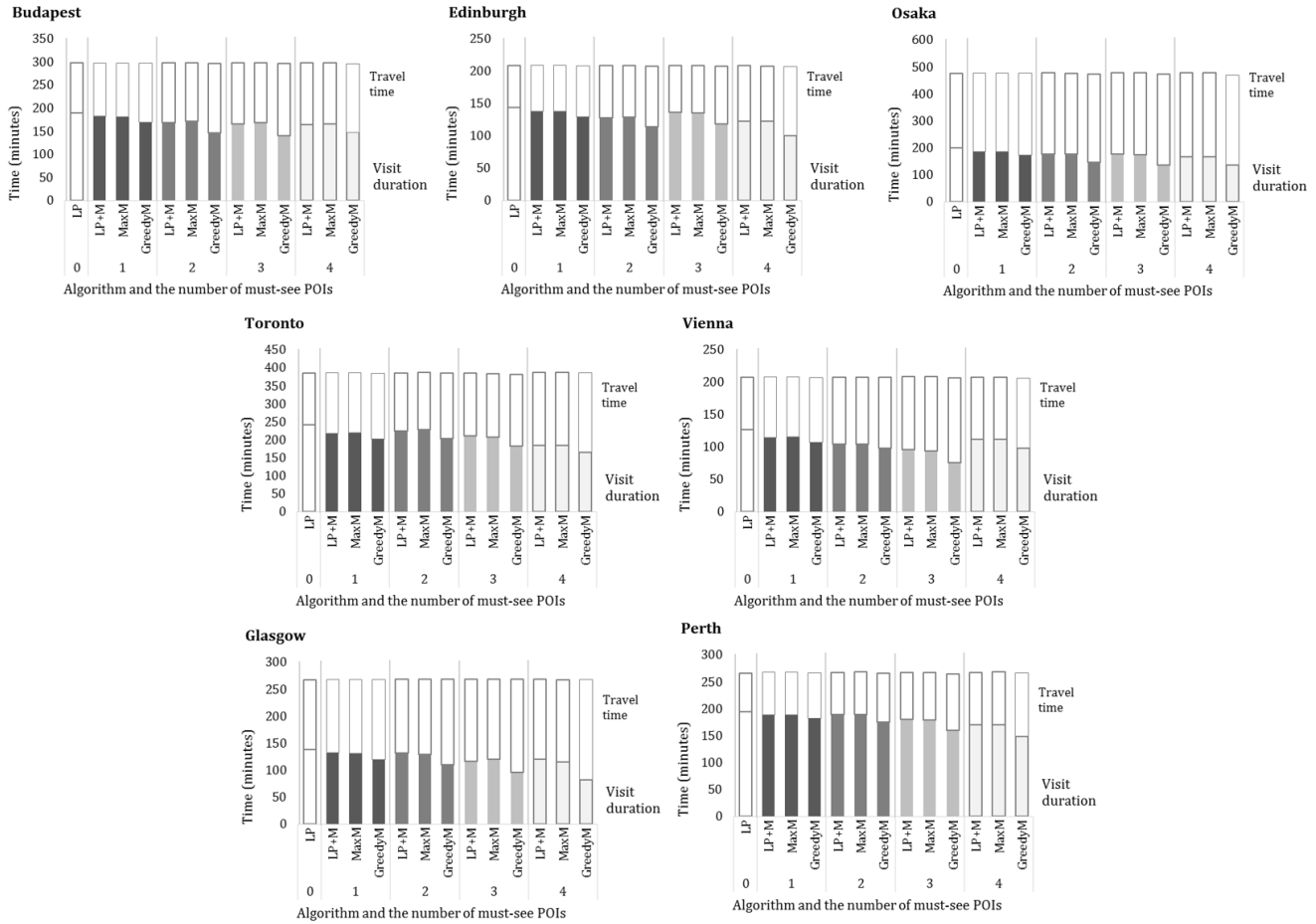
error including failed tours and clearly displays the retarding effects of must-see POIs.

The results on the average profit of tour itineraries confirm the superior performance that LP+M has over the MaxM and GreedyM algorithms. LP+M consistently recommends tour itineraries with greater profitability and a higher POI visitation rate than GreedyM, and found greater numbers of successful solutions than MaxM. As expected, the average profits decline with both approaches as the must-see set increases in size. However, regardless of the size of must-see POIs set, LP+M offers the best performance in terms of tour profit, followed by MaxM and then GreedyM.

5.3 Total POIs in Recommended Tours

As shown in Table 5, the average number of total POIs visited for each algorithm's tours tends to decline from the optimum as the must-see set increases. The must-see size of four often included more POIs in successful tours than smaller must-see sets. The GreedyM algorithm produced tours with a POI count that was on average lower than that for LP+M and MaxM. This result indicates sub-optimal performance of GreedyM, compared to LP+M and MaxM.

Figure 3: Average travel time and visit duration, top columns represent travel time while the bottom columns represent visit duration. Higher values for visit duration are better, while lower values for travel time are better.



Compared to GreedyM and MaxM, LP+M consistently recommends tour itineraries with the most number of POIs for the same number of must-see POI. This result shows that despite the same constraint in terms of must-see POIs, LP+M is able to recommend tours that satisfy these constraints and yet recommend better tours that comprises more POIs.

5.4 Utilized Budget of Recommended Tours

While the total time budget for the itineraries was never exceeded, GreedyM produced the shortest tours and wasted the largest amounts of available time, as shown in Figure 3. GreedyM also produced solutions with higher average travel times compared with the LP, LP+M and MaxM approaches. This result is not surprising given that GreedyM does not optimise the tour as a whole, but rather does it in two parts leading to inefficiencies.

The inclusion of must-see POIs as a model constraint when generating travel itineraries results in tours with lower profitability and higher travel times compared to those without must-see POIs. The extent of this reduced performance increases markedly as the size of the must-see set increases. Further, the likelihood of failing

to generate a valid itinerary increases substantially as more must-see POIs are included in the tour. Interestingly though, there is a high chance (around 30%) that a single must-see POI may be visited by an algorithm which ignores the must-see set when optimising.

5.5 Discussion

These results show the general good performance of our proposed LP+M algorithm over the GreedyM and MaxM baselines, while the LP algorithm was included to demonstrate an optimal solution that does not consider must-see POIs. Compared to LP, GreedyM and MaxM, our proposed LP+M algorithm is able to recommend the highest number of successful tour itineraries, i.e., those that include the must-see POIs. In addition, our LP+M algorithm is able to consistently recommend tour itineraries that comprises a higher number of POIs and a higher total tour profit, compared to the GreedyM and MaxM baselines, thus resulting in tour itineraries that are more satisfying for tourists based on these measures. The LP+M algorithm also utilizes the available time budget more efficiently than the GreedyM baselines, as it schedules tour itineraries with more time allocated to visiting POIs than travelling between POIs.

Based on these evaluation metrics, experimental results show that LP+M is the overall best performing algorithm, followed by MaxM then GreedyM.

6 CONCLUSION AND FUTURE WORK

In this work, we have proposed and formulated the TOURMUSTSEE problem, which incorporated a set of must-see POIs into travel itineraries, along with considerations of a starting/ending POIs and travel times between POIs and visit durations at POIs. We then solved this TOURMUSTSEE problem as an Integer Linear Program, and we denote this approach as the LP+M algorithm. Compared with various baselines, LP+M performed consistently better with higher average profits, a greater average number of visited POIs, and a higher POI visit duration. Our algorithm incorporates both travel time and POI visit durations and incorporates real-world data gathered from publicly accessible sources. Using a well established ILP model as the base for LP+M also ensures a robust and efficient optimisation strategy which produces consistent results.

This work fills a gap in existing research by providing a specific analysis of must-see POIs and their impact on optimal itineraries. Future work would include further evaluation with a more diverse range of alternate heuristics to validate this research, and to repeat the experiment on more cities and POI sets. Nonetheless, it is clear through our evaluations that LP+M successfully and effectively incorporates must-see POIs into optimised itineraries and out-performs various baselines. In future, we intend to extend this work to recommend tour itineraries that span multiple days and regions, and consider additional aspects such as weather, accommodation, restaurants and POI admission fees.

Acknowledgments. Kwan Hui Lim is supported by a Defence Science and Technology Group Postdoctoral Fellowship. The authors thank the reviewers for their helpful comments.

REFERENCES

- [1] Idir Benouaret and Dominique Lenne. 2016. A Composite Recommendation System for Planning Tourist Visits. In *Proc. of the 2016 IEEE/WIC/ACM Intl. Conf. on Web Intelligence (WI'16)*. 626–631.
- [2] Michel Berkelaar, Kjell Eikland, and Peter Notebaert. 2004. Ipsolve: Open source (mixed-integer) linear programming system. (2004). <http://Ipsolve.sourceforge.net/>.
- [3] Laarabi Bochar and Bouchaib Radi. 2016. A new approach to treat the selective travelling salesman problem. *Intl. Mathematical Forum* 11, 16 (2016), 757–768.
- [4] Igo Brillhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2013. Where shall we go today? Planning touristic tours with TripBuilder. In *Proc. of the 22nd ACM Intl. Conf. on Information and Knowledge Management (CIKM'13)*. 757–762.
- [5] Igo Brillhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2014. TripBuilder: A Tool for Recommending Sightseeing Tours. In *Proc. of the 36th European Conf. on Information Retrieval (ECIR'14)*. 771–774.
- [6] Igo Ramalho Brillhante, Jose Antonio Macedo, Franco Maria Nardini, Raffaele Perego, and Chiara Renso. 2015. On planning sightseeing tours with TripBuilder. *Information Processing & Management* 51, 2 (2015), 1–15.
- [7] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. 2010. Automatic construction of travel itineraries using social breadcrumbs. In *Proc. of the 21st ACM Conf. on Hypertext and Hypermedia (HT'10)*. 35–44.
- [8] Munmun De Choudhury, Moran Feldman, Sihem Amer-Yahia, Nadav Golbandi, Ronny Lempel, and Cong Yu. 2010. Constructing travel itineraries from tagged geo-temporal breadcrumbs. In *Proc. of the 19th Intl. Conf. on World wide web (WWW'10)*. 1083–1084.
- [9] Dominique Feillet, Pierre Dejax, and Michel Gendreau. 2005. Traveling Salesman Problems with Profits. *Transportation Science* 39, 2 (2005), 188–205.
- [10] Michel Gendreau, Gilbert Laporte, and Frederic Semet. 1998. A branch-and-cut algorithm for the undirected selective traveling salesman problem. *Networks* 32, 4 (1998), 263–273.
- [11] Aldy Gunawan, Hoong Chuin Lau, and Pieter Vansteenwegen. 2016. Orienteering Problem: A survey of recent variants, solution approaches and applications. *European Journal of Operational Research* 255, 2 (2016), 315–332.
- [12] Christoph Kofler, Luz Caballero, Maria Menendez, Valentina Occhialini, and Martha Larson. 2011. Near2me: An authentic and personalized social media-based recommender for travel destinations. In *Proc. of the 3rd ACM SIGMM Intl. Workshop on Social Media (WSM'11)*. 47–52.
- [13] Gilbert Laporte and Silvano Martello. 1990. The selective traveling salesman problem. *Discrete applied mathematics* 26, 2-3 (1990), 193–207.
- [14] Gilbert Laporte, Hélène Mercure, and Yves Norbert. 1984. Optimal tour planning with specified nodes. *RAIRO - Operations Research (Recherche Opérationnelle)* 18, 3 (1984), 203–210.
- [15] Kenneth Wai-Ting Leung, Dik Lun Lee, and Wang-Chien Lee. 2011. CLR: a collaborative location recommendation framework based on co-clustering. In *Proc. of the 34th Intl. ACM SIGIR Conf. on Research and development in Information Retrieval (SIGIR'11)*. 305–314.
- [16] Zhenping Li, Rui-Sheng Wang, Hong-Wei Liu, and Wenfeng Zhou. 2010. Models and Algorithms for the Constrained Orienteering Problem. In *Proc. of the Ninth Intl. Symposium on Operations Research and Its Applications (ISORA'10)*. 89–97.
- [17] Kwan Hui Lim. 2015. Recommending Tours and Places-of-Interest based on User Interests from Geo-tagged Photos. In *Proc. of the 2015 SIGMOD PhD Symposium (SIGMOD'15)*. 33–38.
- [18] Kwan Hui Lim, Jeffrey Chan, Shanika Karunasekera, and Christopher Leckie. 2017. Personalized Itinerary Recommendation with Queuing Time Awareness. In *Proc. of the 40th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'17)*. 325–334.
- [19] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2016. Towards Next Generation Touring: Personalized Group Tours. In *Proc. of the 26th Intl. Conf. on Automated Planning and Scheduling (ICAPS'16)*. 412–420.
- [20] Kwan Hui Lim, Jeffrey Chan, Christopher Leckie, and Shanika Karunasekera. 2018. Personalized Trip Recommendation for Tourists based on User Interests, Points of Interest Visit Durations and Visit Recency. *Knowledge and Information Systems* 54, 2 (2018), 375–406.
- [21] Clair E. Miller, Albert W. Tucker, and Richard A. Zemlin. 1960. Integer programming formulation of traveling salesman problems. *J. ACM* 7, 4 (1960), 326–329.
- [22] Ksenia D. Mukhina, Stepan V. Rakitin, and Alexander A. Visheratin. 2017. Detection of tourists attraction points using Instagram profiles. *Procedia Computer Science* 108 (2017), 315–332.
- [23] Daniele Quercia, Rossano Schifanella, and Luca Maria Aiello. 2014. The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proc. of the 25th ACM Conf. on Hypertext and Social Media (HT'14)*. 116–125.
- [24] Yue Shi, Pavel Serdyukov, Alan Hanjalic, and Martha Larson. 2011. Personalized Landmark Recommendation Based on Geotags from Photo Sharing Sites. In *Proc. of the Fifth Intl. AAAI Conf. on Weblogs and Social Media (ICWSM'11)*. 622–625.
- [25] Bart Thomee, David A. Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. 2016. YFCC100M: The New Data in Multimedia Research. *Commun. ACM* 59, 2 (2016), 64–73.
- [26] Theodore Tsiligruides. 1984. Heuristic methods applied to Orienteering. *Journal of the Operational Research Society* 35, 9 (1984), 797–809.
- [27] Pieter Vansteenwegen, Wouter Souffriau, and Dirk Van Oudheusden. 2011. The Orienteering problem: A survey. *European Journal of Operational Research* 209, 1 (2011), 1–10.
- [28] Xiaoting Wang, Christopher Leckie, Jeffery Chan, Kwan Hui Lim, and Tharshan Vaithianathan. 2016. Improving Personalized Trip Recommendation to Avoid Crowds Using Pedestrian Sensor Data. In *Proc. of the 25th ACM Intl. Conf. on Information and Knowledge Management (CIKM'16)*. 25–34.
- [29] Wikipedia. 2016. List of sights and historic places in Budapest. (2016). https://en.wikipedia.org/w/index.php?title=List_of_sights_and_historic_places_in_Budapest.
- [30] Wikipedia. 2016. Tourist attractions in Toronto. (2016). https://en.wikipedia.org/w/index.php?title=Attractions_in_Toronto.
- [31] Wikipedia. 2016. Tourist attractions in Vienna. (2016). https://en.wikipedia.org/w/index.php?title=Tourist_attractions_in_Vienna.
- [32] Wikipedia. 2016. Visitor attractions in Edinburgh. (2016). https://en.wikipedia.org/wiki/Category:Visitor_attractions_in_Edinburgh.
- [33] Wolfgang Wörndl and Alexander Hefele. 2016. Generating Paths Through Discovered Places-of-Interests for City Trip Planning. In *Information and Communication Technologies in Tourism*. Springer Intl. Publishing, 441–453.
- [34] Lina Yao, Quan Z. Sheng, Yongrui Qin, Xianzhi Wang, Ali Shemshadi, and Qi He. 2015. Context-aware Point-of-Interest Recommendation Using Tensor Factorization with Social Regularization. In *Proc. of the 38th Intl. ACM SIGIR Conf. on Research and Development in Information Retrieval (SIGIR'15)*. 1007–1010.
- [35] Quan Yuan, Gao Cong, Zongyang Ma, Aixun Sun, and Nadia Magnenat Thalmann. 2013. Time-aware point-of-interest recommendation. In *Proc. of the 36th Intl. ACM SIGIR Conf. on Research and development in information retrieval (SIGIR'13)*. 363–372.