Optimized Itinerary Planning for Tourist Attractions

Tarun Rambha
Indian Institute of Science (IISc), Bengaluru, India
tarunrambha@iisc.ac.in

Extended abstract submitted for presentation at the 12th *Triennial Symposium on Transportation Analysis conference (TRISTAN XII)*
June 22-27, 2025, Okinawa, Japan

October 31, 2024

Keywords: travel itinerary problem, integer programming, column generation, time-dependent graphs

1 INTRODUCTION

Globally, international tourism has increased since COVID, and receipts from this industry were found to exceed 1.5 trillion USD in 2023. Several tourism companies offer web and mobile services to help travelers plan their itineraries when visiting attractions within a city. For example, platforms such as *Go City*, *CityPASS*, *Paris Passlib'*, and *I Amsterdam* sell passes for multiple attractions (ranging from day passes to six-day passes). However, these services are often not integrated with mobility/routing applications. Users touring a new place often depend on public transportation or ride-hailing services and taxis and have to plan their itineraries manually, i.e., the order in which attractions must be visited and how to get from one attraction to another. This research focuses on optimizing costs of such itineraries using an integer programming model.

Although the problem resembles TSPs and VRPs at the outset, several features make this variant challenging. First, travel times between attractions are time-dependent (due to congestion for private transport modes or scheduled transit trips). Second, the duration of the visit may also depend on when a traveler begins touring an attraction. Certain times of the day might be crowded, or some attractions may start at specific times (such as guided tours). In addition, public transportation services, fares, opening hours for attractions, and entry prices could also vary from day to day. Furthermore, the problem parameters may not satisfy First In First Out (FIFO) properties. For example, a traveler can catch a faster metro/train service by starting later. In attractions with a network of queues, travelers arriving later can finish their tour earlier. From a mathematical standpoint, several of these features complicate modeling. For instance, the MTZ constraints for the TSP would take the form $t_j \geq t_i + \delta_i(t_i) + \tau_{ij}(t_i + \delta_i(t_i)) + -M(1 - x_{ij})$, where t_i represents arrival time at attraction i, τ_{ij} is the travel time between i and j and δ_i denotes the duration of visit at attraction i. The time dependence is usually non-linear, and this problem cannot be reformulated as one of the existing models.

The literature on time-dependent TSPs (Gendreau *et al.*, 2015, Montero *et al.*, 2017, Fontaine *et al.*, 2023) focuses on road networks and hence makes different assumptions that do not hold in this setting. Very few studies assume service times are time-dependent but assume FIFO ordering (Taş *et al.*, 2016). The literature on itinerary planning does not explore time-dependence in travel and visiting attractions and focuses mainly on heuristics (Kotiloglu *et al.*, 2017, Liao & Zheng, 2018). Our main contribution is formulating this problem as an integer programming model and proposing a column-generation approach to discover new itineraries integrated into a price-and-branch method. The period of interest is discretized into smaller intervals, and the pricing problem is formulated as a shortest path problem with conflicts.

2 METHODOLOGY

Table 1 – *Notation for the IP formulation*

Symbol	Description
Sets	
D	Set of days for the tour
A	Set of attractions to be visited
P_d	Set of time periods available to travel and to visit attractions on day d
W_i^d	Set of time periods or time window in which attraction i is open on day d
Parameters	
τ_{ij}^{dp}	Travel time between locations i and j on day d departing in period p
δ_i^{dp}	Duration of the visit to attraction i on day d when arriving in period p
κ_i^{dp}	Cost of visiting attraction i on day d in period p
π_d	Last time period for touring on day d (i.e., maximum of elements in P_d)
ω_i^d	Closing time period of attraction i on day d (i.e., maximum of elements in W_i^d)
λ_d	Starting location of the traveler on day d
μ_d	Ending location of the traveler on day d
Variables	
x_{ij}^{dp}	Binary variable which is one if the traveler moves from node i (origin location or attraction) to node j (destination location or attraction) on day d departing in period p, i.e., i, j
y_i^{dp}	Binary variable which is one traveler arrives at attraction i on day d in period p and is zero otherwise

Time is divided into equal intervals and are assumed to be contiguous. The Tourist Itinerary Planning (TIP) problem involves minimizing the cost of visiting a given set of attractions. It can be formulated using the following IP (some constraints are skipped due to space limitations).

$$\max \sum_{d \in D} \sum_{i \in A} \sum_{p \in P_d \cap W_i^d} \kappa_i^{dp} y_i^{dp} \tag{1}$$

$$\text{s.t.} \sum_{d \in D} \sum_{p \in P_d \cap W_i^d} y_i^{dp} \leq 1 \qquad \forall\, i \in A \tag{2}$$

$$\sum_{q \in P_d \cap W_j^d, q \geq p + \tau_{ij}^{dp}} y_j^{dq} \geq x_{ij}^{dp} \qquad \forall\, d \in D,\ i \in A \cup \{\lambda_d\}, j \in A,\ p \in P_d \tag{3}$$

$$\sum_{j \in A \cup \{\mu_d\} \setminus \{i\}} \sum_{q \in P_d, q \geq p + \delta_i^{dp}} x_{ij}^{dq} \geq y_i^{dp} \qquad \forall\, d \in D,\ i \in A,\ p \in P_d \cap W_i^d \tag{4}$$

$$\sum_{p \in P_d \cap W_i^d} (p + \delta_i^{dp}) y_i^{dp} \geq \omega_i^d \qquad \forall\, d \in D,\ i \in A \tag{5}$$

$$\sum_{i \in A} \sum_{p \in P_d} (p + \tau_{i\mu_d}^{dp}) x_{i\mu_d}^{dp} \leq \pi_d \qquad \forall\, d \in D \tag{6}$$

$$x_{ij}^{pd} \in \{0, 1\} \qquad \forall\, d \in D,\ i \in A \cup \{\lambda_d\},\ j \in A \cup \{\mu_d\},\ p \in P_d \tag{7}$$

$$y_i^{pd} \in \{0, 1\} \qquad \forall\, d \in D,\ i \in A,\ p \in P_d \cap W_i^d \tag{8}$$

An attraction is visited at most once due to (2). Constraint (3) sets the start time of touring attraction j depending on when the tourist left the attraction/origin i. If x_{ij}^{dp} is active in period p, then the traveler would reach attraction j at $p + \tau_{ij}^{dp}$ and can hence tour attraction j after that time which is captured in the left-hand side (note that waiting is allowed and hence the x variables may be active in periods outside the open hours of attractions but the y variables cannot). Constraint (4) sets the departure decisions at attraction i depending on when the traveler started touring attraction i. If y_i^{dp} is active in period p, then it takes the traveler $p+\delta_i^{dp}$ periods before they can travel to any other attraction/destination $j \neq i$. Visit to an attraction must finish before its closing time because of (5). Note that $\sum_{p\in P_d} p y_i^{dp}$ is the period in which the traveler starts visiting attraction i on day d and $\sum_{p\in P_d}(p+\delta_i^{dp})y_i^{dp}$ indicates the period of completion of the visit at attraction i. The traveler must be able to reach their destination for each day due to (6). Two more sets of constraints are not shown here. The first restricts the sum of x variables, leaving attraction i across days and time periods to be equal to one, and the second ensures that the traveler is only at one location in every period. Finally, (7) and (8) impose integrality constraints on the decision variables.

The above formulation can be reformulated using new binary path variables z_r^d, which connects the start locations λ_d with μ_d. We first create a time-expanded graph with four types of nodes copies: (1) the origin locations expanded as (λ_d, p), where $p \in P_d$, (2) the destination locations (μ_d, p), where $p \in P_d$, (3) 'in' attraction nodes at which one starts visiting an attraction and are of the type $(i, p, \text{'in'})$, and (4) 'out' attraction nodes which represent the end of an attraction visit and are of the type $(i, p, \text{'out'})$. Figure 1 shows an example graph with three attractions. The node copies for different periods are aligned vertically, with the ones at the bottom representing earlier time periods. The traveler starts at the bottom left origin node and proceeds towards the top right destination node. Waiting is modeled using the black arcs. The red edges are used for traveling between origins/destinations and between attractions. The purple edges connect the 'in' and 'out' nodes of an attraction and represent the visit duration. Attraction 1 does not open until the fourth period; hence, the first three periods only have an 'in' node. Likewise, Attraction 2 closes three time periods before the traveler's last time period. These nodes are needed since the FIFO property may not hold, as seen in some intersecting edges. Note that the time-expanded graph can be different for different days depending on traffic and transit schedules and properties of attractions. Let z_r^d be an itinerary the traveller chooses on day d and let c_r^d be its cost. Then, the set-partitioning version of the TIP problem can be written as $\min \sum_{d\in D}\sum_{r\in R_d} c_r^d z_r^d$, subject to $\sum_{d\in D}\sum_{r\in R_d} z_r^d = 1 \ \forall\ i \in A$, and $\sum_{r\in R_d} z_r^d \leq 1 \ \forall\ d \in D$.

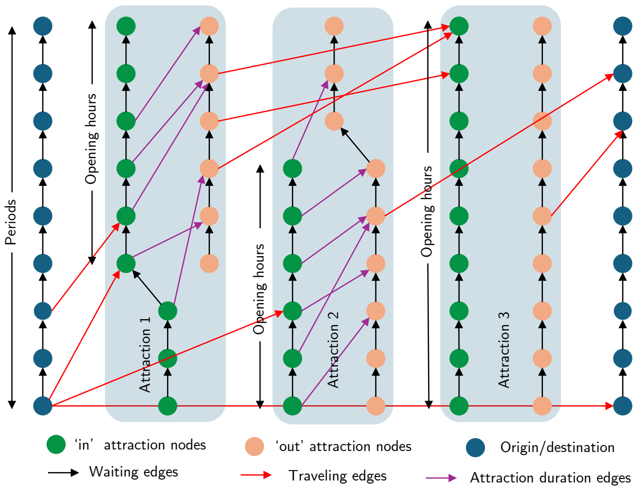

Figure 1 – *Time expanded graph for the pricing problem*

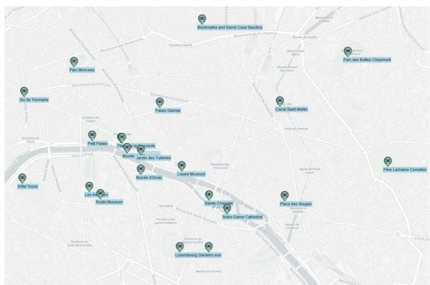

Figure 2 – *Sample attractions*

For multi-day itineraries, we generate the initial solutions to this problem greedily by finding attractions that can be visited on the first day, removing them from the set of available attractions for the next day, and proceeding similarly. Using the dual values for each attraction, we set up pricing problems by modifying the costs of the purple edges. The pricing problem is a variant of the shortest path, which prevents re-visiting the same attraction. Since it is formulated on a time-expanded graph, it can be viewed as a shortest path with conflicts (Şuvak *et al.*, 2021, Cerulli *et al.*, 2023) where only at most one of the purple edges is allowed to carry flows.

3 RESULTS

We tested the two approaches using attractions in Paris (Figure 2) and an intel i7-12700 machine. The edge costs were drawn from a uniform distribution and were assumed to vary over different days and time periods. Table 2 shows the performance of the IP (with a runtime limit of 10 min) and column generation (CG). We tested 5 instances with different numbers of attractions (column 1) and 1–5 days (rows). The CG method (using an integer program for the pricing) finds the optimal solution faster than IP in all cases. In many cases, the IP does not find an integer incumbent; in some cases, the CG approach does not find an initial greedy solution.

Table 2 – *Objective value (runtime in s or GAP in % when terminated early). INF: Infeasible, NFS: No feasible solution found. 1-day solutions were infeasible.*

Days	2		3		4		5	
Attr.	MIP	CG	MIP	CG	MIP	CG	MIP	CG
5	51 (2.3)	51 (1.2)	51 (11.6)	51 (1.9)	51 (34.9)	51 (4.9)	51 (42.3)	51 (6.0)
10	124.5 (0.8)	124.5 (17.9)	124.5 (0.8)	124.5 (25.6)	125.5 (2.3)	124.5 (40.7)	125.5 (2.3)	124.5 (54.3)
15	NFS	NFS	NFS	163 (34.7)	NFS	163 (69.4)	171 (11.6)	163 (81.7)
20	NFS	NFS	NFS	185 (87.8)	NFS	185 (112.7)	NFS	185 (140.1)
25	NFS	NFS	NFS	NFS	NFS	211 (161.0)	NFS	211 (210.2)

4 DISCUSSION

For the future, Paerto-optimal multimodal (transit/private travel) journeys can be used between attractions and origins/destinations to create multiple edges from a node in the time-expanded graph. One could extend the column generation approach to a branch and price method for exact solutions. Finally, preprocessing the graph by removing nodes that cannot be reached (or cannot reach the destination) before a specific time can speed up runtimes.

References

Cerulli, Raffaele, Guerriero, Francesca, Scalzo, Edoardo, & Sorgente, Carmine. 2023. Shortest paths with exclusive-disjunction arc pairs conflicts. *Computers Operations Research*, **152**, 106158.

Fontaine, Romain, Dibangoye, Jilles, & Solnon, Christine. 2023. Exact and anytime approach for solving the time dependent traveling salesman problem with time windows. *European Journal of Operational Research*, **311**(3), 833–844.

Gendreau, Michel, Ghiani, Gianpaolo, & Guerriero, Emanuela. 2015. Time-dependent routing problems: A review. *Computers Operations Research*, **64**, 189–197.

Kotiloglu, S., Lappas, T., Pelechrinis, K., & Repoussis, P.P. 2017. Personalized multi-period tour recommendations. *Tourism Management*, **62**, 76–88.

Liao, Zhixue, & Zheng, Weimin. 2018. Using a heuristic algorithm to design a personalized day tour route in a time-dependent stochastic environment. *Tourism Management*, **68**, 284–300.

Montero, Agustín, Méndez-Díaz, Isabel, & Miranda-Bront, Juan José. 2017. An integer programming approach for the time-dependent traveling salesman problem with time windows. *Computers Operations Research*, **88**, 280–289.

Taş, Duygu, Gendreau, Michel, Jabali, Ola, & Laporte, Gilbert. 2016. The traveling salesman problem with time-dependent service times. *European Journal of Operational Research*, **248**(2), 372–383.

Şuvak, Zeynep, Altınel, İ. Kuban, & Aras, Necati. 2021. Minimum cost flow problem with conflicts. *Networks*, **78**(4), 421–442.