The Traveler's Problem

Iva Babukova

September 2016

In this work we present the Traveler's Problem (TP), a computational task whose extensions and variations are often encountered by travelers around the world. The task is concerned with creating a valid travel schedule, using airplanes as a means of transportation and in accordance with certain constraints specified by the traveler.

1 Problem Formulation

Each instance of TP consists of:

1. A set of airports A. Each airport a_i in A represents a location the traveler can begin their commute in, visit as a desired destination, or connect in on the way to their destination.

2. The total travel time T, within which the traveler must have visited all destinations and returned to the home point.

3. A set of flights F. Each flight has arrival and departure airport, price, date and duration. A flight from airport a_i to airport a_j is denoted as f_{ij}. The cost and the duration of f_{ij} at date d are c_{ijd} and t_{ijd} respectively. d is a non-negative integer less than or equal to T. The duration and the price are positive numbers.

4. The trip starts and ends at the same airport a_0, which is referred to as the *home point*.

5. A set of *destinations* D, $D \subseteq A$. The traveler must visit every airport in D.

A solution to any instance of TP is a sequence s of valid flights, $f_1, f_2, ..., f_n$, where n is equal to the cardinality of s. We say that s contains valid flights if the flights in s have the following properties:

1. The departure airport of f_1 and the arrival airport of f_n must be both equal to a_0, that is:

$$f_1 = f_{0j}$$
$$f_n = f_{i0}$$

for some $i, j \leq |A|$

2. For any two consecutive flights in s: f_k, f_{k+1}, the arrival airport of f_k is equal to the departure airport of f_{k+1}.
3. The flights in s have dates which occur in increasing, temporal order (flights earlier in the list happen earlier in time).
4. For any two consecutive flights f_k and f_{k+1} in s, the date of f_{k+1} must be bigger than or equal to the sum of date and the duration of f_k.
5. The sum of the date and the duration of the last flight in s must be less than or equal to T.
6. The date of the first flight in s must be bigger than or equal to 0, assuming that the traveler starts their trip at day 0.
7. Every airport in D is included in at least two flights in s, which must be consecutive (follows from condition **??**).
8. A valid sequence of flights may contain 0, 1 or more flights to and from airports that are not destinations. Such airports are called *connections*.

The *optimization* version of TP (TPO) asks for an *optimal* solution, which is a solution that minimizes the total sum of the prices of the flights in s.

The *decision* version of TP (TPD) asks whether there exists a valid sequence of flights s, such that the sum of the costs of all flights in s is less than or equal to some integer B. The solution of this problem is a 'yes' or 'no' answer.

There exists a variety of additional constraints and extensions that can be added to TP. Our problem formulation has only presented the hard constraints so far. Every valid solution to a TP instance must satisfy these constraints. In real-world problems, travelers may have additional preferences (soft constraints) with regards to their travel. These are discussed in the next section.

1.1 Soft Constraints

It may be desirable to search for a solution that satisfies some of the following requirements:

- Travelers may wish to spend a certain amount of t_i days in each destination d_i. t_i may be specified as a lower or a higher bound.
- Travelers may wish to take only direct flights. In such requirement, a valid solution must contain flights only to and from airports in the set of destinations D.

- Some travelers may want to spend as little time on flying as possible. In such case, we wish to find a solution with minimal total flight duration time (that is the sum of the duration of each flight in a solution of TP).
- Travelers may require to spend a given date at a given destination, for example due to an event occurring on that date in this destination.
- Travelers may have a certain price threshold (budget) B, such that if the total travel cost exceeds that threshold the traveler will discard a solution as invalid. In such case, an algorithm for TPD is required.

It is therefore suggested that any attempt at an investigation of TP assumes as an additional non-functional requirement that any proposed model to solve TP is flexible and can be easily extended by adding, removing and modifying the aforementioned soft constraints.

2 Worked Example

We present an example instance of TP and its most optimal solution. Note that in this example we do not wish to satisfy any of the previously listed soft constraints.

Example. A traveler wishes to visit 4 airports from a set of 7 airports available to travel to and from:

Glasgow (G), Berlin (B), Milan (M), Amsterdam (A), Paris (P), Frankfurt (F), London (L).

Airport G is the home point, F and L are connections, and B, M, A and P are the destinations. The travel time of the traveler is 15 days. All available flights are listed on Table**??**. For simplicity, the duration of each flight is assumed to be 1 day.

Solution. A valid solution of the TP instance in the example above is the sequence of flights s, where the each flight is represented by its flight number, specified in the first column of Table **??**:

$$s = \{GA1, AP4, PM6, MF9, FB11, BG13\}$$

The total cost of the flights in s is 699.

Optimal Solution. The optimal solution of the example TP instance is the sequence of flights s', where the each flight is represented by its flight number, specified in the first column of Table **??**:

$$s' = \{GA1, AP4, PM6, MF9, FB11, BL13, LG14\}$$

The total cost of the flights in s' is 483.

s' is the optimal solution, as the total cost of the flights is minimized.

Flight No	**Departs**	**Arrives**	**Date**	**Price**
GA1	G	A	1	74
GF1	G	F	1	86
GL3	G	L	3	25
AP4	A	P	4	58
LG14	L	G	14	24
PL12	P	L	12	45
PM6	P	M	6	71
MF9	M	F	9	39
FB11	F	B	11	122
FM8	F	M	8	234
BG13	B	G	13	335
BL13	B	L	13	102

Table 1: List of flights with departure and arrival airports, flight date and price.

In this work, we present the traveler's problem (TP). An instance of the problem serves as a worked example, for which two possible solutions are given, one of which is optimal. We have not found any extensive study on TP and its variations in the existing literature. An investigation of this problem could be of both theoretical and practical interest.

3 Complexity of TP

We state a theorem about the complexity of TP and and prove it.

Theorem 1. The decision version of TP is NP-complete.

Proof. This proof first shows that TP is in NP. To prove NP-completeness, we construct a polynomial-time reduction from a known NP-complete problem Π to TP. We choose Π to be the traveling salesman problem (TSP). Its NP-completeness follows by a reduction from the Hamiltonian Cycle problem. The proof is presented in [**?**].

TP is in NP since a non-deterministic algorithm can guess a sequence of flights s and check in polynomial time whether s is a valid solution.

Let π be an instance of TSP that has a set C of n cities, distance $\mathrm{d}(c_i, c_j)$ between each pair of cities $c_i, c_j \in C$, positive integer B. The question we ask is whether there exists a tour that starts and ends at the same city c_1 and visits all cities in C, having length B or less.

Let π' be an instance of TPD. Let the set of airports in π' C' be identical to the set of cities C in π (a city in π is an airport in π'). Let each airport in C' also be a destination. Let T be sufficiently large number that allows for all airports to be visited. Let the cost of flight from a given airport a_i to an airport a_j be an integer that is equal to the distance between a_i and a_j. Let B be the upper bound on the allowed total flights cost.

Suppose that the sequence $\gamma = < c_1, c_2, ..., c_n >$ is solution of π. Let s be the sequence of flights that solves π'. From the construction of the π' instance it follows that the flights in s visit airports in C' that are identical to the cities in C in the same order γ. Therefore, a solution of π is also a solution of π'.

If s is a solution of π', where the airports in C' are visited in the sequence $\gamma' = < a_1, a_2, ..., a_n >$, then γ' is a valid solution of π.

We showed that π has a solution if and only if there exits a solution to π'.

The transformation from a TSP instance to an instance of TP can be done in polynomial time. For each of the $n(n$-1)/2 distances d(c_i, c_j) that must be specified in π, it is sufficient to check that the same cost is assigned to the flights from a_i to a_j for all dates.

Therefore, TP is in NP and TSP can be reduced to TPD in polynomial time, from which it follows that TP is NP-complete.

□

4 Existing work

No existing work on this problem has been found so far in the literature. However, a similar problem, namely TSP has been a subject to extensive research in the recent years.

TSP

TSP with time windows (TSPTW)

vehicle-routing problem (VRP)

job-shop scheduling problem (JSSP)