Solving Traveling Salesman problem using R in context of {sf} & HERE API

Jindřich Lacko

**Abstrakt**

Problém obchodního cestujícího je jedním z nejstarších a teoreticky nejlépe zpracovaných témat v oblasti optimalizačních úloh, s vysokou mírou standardizace řešení. Příspěvek ilustruje možnosti praktického využití jednoho z nástrojů - rozšiřujícího balíčku {TSP} - v kontextu statistického programovacího jazyka R. Dále demonstruje využití tří alternativních metrik pro optimalizaci: vzdálenosti vzdušnou čarou, dojezdovou dráhu autem a čas cesty autem.

**Klíčové slova:** *optimalizace, TSP, R, HERE API*

**Abstract**

The traveling salesman problem is one of the oldest and theoretically best understood problems in optimization techniques. From theoretical point of view the approaches to its solution are highly standardized. This article demonstrates several empirically advantageous approaches to solving Traveling Salesman Problem within context of the statistical programming language R. It further demonstrates three alternative metrics for optimization: Euclidean distance, travel distance by car and travel time by car.

**Keywords:** *optimization, TSP, R, HERE API*

# 1. Problem statement

The Traveling Salesman Problem (TSP) is an optimization classic, with a number of well understood and highly standardized solutions available in the context of statistical programming language R.

The problem can be mathematically formulated, following (1), as: Given a weighted complete digraph (*Kn, c*), find a Hamiltonian cycle in *Kn* of minimum cost. The cost function *c* can be determined by the cost matrix [cij]. The cost – or distance – matrix thus plays a key role in defining, and solving, the TSP.

In this article I would like to share a practical example of solving the TSP using Open Street Map (2) data of bars via {osmdata} (3) and routing engine provided by HERE Global B.V. Kennedyplein 222-226, 5611 ZT Eindhoven, Netherlands as an Application Programming Interface (API) service and accessed from R via {hereR} (4). The actual solution will be found by utilizing the {TSP} (5) package.

The Prague School of Economics, is located in Žižkov. A formerly working class neighborhood, now rather gentrified, it has to this day retained some traces of its former rougher edges. One of these is an active night life.

A visit to the bars of Žižkov is therefore a familiar activity for many VŠE students, and can serve as an introduction to serious optimization techniques.

# 2. Data Acquisition

The first step in our exercise is acquiring data of Žižkov bars. A search is performed over the area of *core Žižkov*, defined as a polygon, using the OSM Overpass API. As there seems not to be a clear consensus over what constitutes a bar, restaurant or a pub in Prague I am including all three of the possible amenities.

|  |
| --- |
| **Figure 1**: Map of Žižkov with full set (n = 75) of bars.  Source: OpenStreetMap + authors own work |

We have located 75 bars, implying a cost matrix of 5625 elements. Not a huge one by today’s standards – but big enough to think twice about trying to solve using a pen and a piece of paper.

# 3. Euclidean distance TSP

The easiest cost matrix to calculate is plain “as the crow flies” distance.

The resulting matrix (Table 1) will be based on pure distance, with some differences in interpretation depending on coordinate reference system of underlying data (Euclidean (6) in projected CRS and spherical in unprojected CRS).

Calculating the cost matrix using plain distance is easy, and the cost matrix is symmetrical [cij] equals [cji]. The problem can be be formulated such as the Kn being considered is an undirected graph.

Solving the TSP for such a matrix is straightforward, as optimization toolset required is readily implemented in the {TSP} package.

The optimal route cost is 1697 and the metric minimized is meters of length.

From the visual overview (Figure 2) we can see an obvious shortcoming of the “as the crow flies” approach: it completely ignores other constraints except for distance – such as the layout of a road network.

Thus while the route shown is “optimal” in the sense that it forms the shortest path joining the five bars selected, it is not one that we could actually follow (unless we were a flying crow).

This shortcoming can be resolved by using an alternative cost matrix as input, while retaining the techniques of {TSP} for the actual route selection. A possible source of more applicable data are routing engines, available to R users via API interfacing packages.

The routing results give us several pieces of data:

* the routes as linestring objects in WGS84 (7) (for visualization later on)
* distance of the route (in meters)
* travel time (in seconds) both raw and adjusted for traffic
* fuel consumption (both combustion type and EV)

Having a variety of metrics will be helpful in construction of alternative distance matrices that can be applied under different empirical conditions.

# 4. Travel distance TSP

The next cost matrix will be based on route distance; notice that the matrix (Table 1) is not symmetrical. This is not surprising, as routing is not commutative – optimal route from A to B need not be the same as from B to A, given constraints such as one way roads. Žižkov is a veritable warren of one way streets. In this context the Kn being considered is a directed graph, unlike what was the case for the Euclidean distance.

It will need to be declared as asymmetrical to {TSP} solver; other than that the actual process of solving the matrix will be analogical to the “as the crow flies” matrix.

The optimal route cost is 3569 and the metric minimized is meters of length.

Since the HERE API is generous in terms of results provided it is not difficult to construct an alternative cost matrix, using a different metric. This could be either trip duration or petrol consumption.

In our specific situation both of these can be expected to be highly correlated with the plain distance results. All the streets in Žižkov are of very similar type, and both the average speed and petrol consumption are unlikely to vary greatly between the routes.

The most significant difference between the distance and time based TSP will be driven by current traffic, which is a factor HERE routing engine considers.

# 5. Travel time TSP

Finally one can focus on travel time optimization – a task relevant for emergency services / first responders as well as commercial fast delivery services.

The optimal route cost is 576 and the metric minimized is seconds of time.

The actual route is in our use case very similar to the distance based one. This will not necessarily be the case in other contexts, especially ones with greater variation of road types (city streets vs. highways).

# 6. Conclusion

This article demonstrates relevant empirical advantages to be gained from using:

* a standardized solution (the {TSP} package) to a well known and well understood problem (the Traveling Salesman Problem) within the context of R ecosystem
* construction of distance matrices from HERE API routing results, with option to optimize for multiple metrics (such as minimizing the travel distance, travel time and fuel consumption)

|  |
| --- |
| **Figure 2**: Map of Žižkov with sample (n = 5) of bars and connecting routes.  Source: OpenStreetMap + authors own work |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Table 1**: Cost Matrices  Euclidean Distance Cost Matrix   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | A | B | C | D | E | | A | 0 | 194 | 105 | 616 | 558 | | B | 194 | 0 | 125 | 535 | 365 | | C | 105 | 125 | 0 | 519 | 471 | | D | 616 | 535 | 519 | 0 | 514 | | E | 558 | 365 | 471 | 514 | 0 | | |
| Route Length Cost Matrix   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | |  | A | B | C | D | E | | A | 0 | 829 | 329 | 821 | 1286 | | B | 1096 | 0 | 183 | 893 | 951 | | C | 913 | 500 | 0 | 710 | 1023 | | D | 1027 | 1352 | 852 | 0 | 1412 | | E | 1299 | 363 | 546 | 1655 | 0 | | Route Time Cost Matrix   |  | A | B | C | D | E | | --- | --- | --- | --- | --- | --- | | A | 0 | 160 | 84 | 137 | 206 | | B | 189 | 0 | 37 | 196 | 135 | | C | 144 | 74 | 0 | 139 | 144 | | D | 150 | 212 | 136 | 0 | 212 | | E | 220 | 46 | 83 | 248 | 0 | |

**References**

1. GUTIN, G. and PUNNEN, A. P. *The Traveling Salesman Problem and Its Variations*. Springer Science & Business Media, 2006. ISBN 978-0-306-48213-7.

2. OPENSTREETMAP CONTRIBUTORS. *Planet dump retrieved from https://planet.osm.org*. Online. 2017. Available from: <https://www.openstreetmap.org/>

3. PADGHAM, Mark, LOVELACE, Robin, SALMON, Maëlle and RUDIS, Bob. osmdata. *Journal of Open Source Software*. Online. 26 June 2017. Vol. 2, no. 14, p. 305. [Accessed 7 October 2022]. DOI [10.21105/joss.00305](https://doi.org/10.21105/joss.00305).

4. UNTERFINGER, Merlin and POSSENRIEDE, Daniel. *hereR: “sf”- Based Interface to the “HERE” REST APIs*. Online. 0.9.0. R. 5 August 2022. [Accessed 7 October 2022]. Available from: <https://CRAN.R-project.org/package=hereR>

5. HAHSLER, Michael and HORNIK, Kurt. TSP—Infrastructure for the Traveling Salesperson Problem. *Journal of Statistical Software*. Online. 2008. Vol. 23, p. 1–21. [Accessed 7 October 2022]. DOI [10.18637/jss.v023.i02](https://doi.org/10.18637/jss.v023.i02).

6. HARTSHORNE, Robin. *Geometry: Euclid and Beyond*. Springer Science & Business Media, 2013. ISBN 978-0-387-22676-7.

7. DMA TR 8350.2: *Department of Defense World Geodetic System 1984, Its Definition and Relationships With Local Geodetic Systems*. Defense Mapping Agency Systems Center 8613 Lee Highway, Fairfax VA 22031-2138, 1991.

**Contact**

University of Economics in Prague, Faculty of Informatics and Statistics, Department of Econometrics

W. Churchill Sq. 4, 130 67 Prague 3, Czech Republic

E-mail: jindra.lacko@vse.cz