#everystreet algorithm

Matej Kerekrety
matej.kerekrety@gmail.com
www.everystreetchallenge.com

August 26, 2020

Abstract

A simple application of *Chinese postman problem* on #everystreet running challenge, where a runner attempts to visit every street of given area at least once. In order to run the challenge optimally and avoid unnecessary detours and running same street twice (or more often) we used an algorithm based on maximal matching which gives us an induced eulerian multigraph where optimal Euler route exists.

Keywords: graph theory, Chinese Postman Problem, maximal matching, every street challenge, running

1 Motivation

Over the COVID-19 lockdown many runners and cyclist found themselves captured within their cities. Some of them ended up running or spinning endless hours on virtual races such as Zwift, but some as e.g. Mark Beaumont [1] decided to join #everystreet challenge. Every street is a challenge originated by Rickey Gates [2, 3] who run *every single street* in city of San Francisco in fall 2018 which took him 46 days and run 1,303 miles.

Inspired by Mark Beaumont who did this challenge over the lockdown in Edinburgh, place where I spend the lockdown, and literally everybody else who managed to accomplish this challenge.[1] We tried to find a algorithm finding an optimal route for given street network area starting and ending in the same point.

2 Every street challenge

Rules of every street challenge are run or cycle[2] every single street of given (metropolitan) area which is usually a city or a neighborhood. Till this point

[1] I would have never had the patience and motivation to run that much in the city.

[2] Based on athlete preference can choose running or cycling this challenge. Most preferred and *original?* options is running.

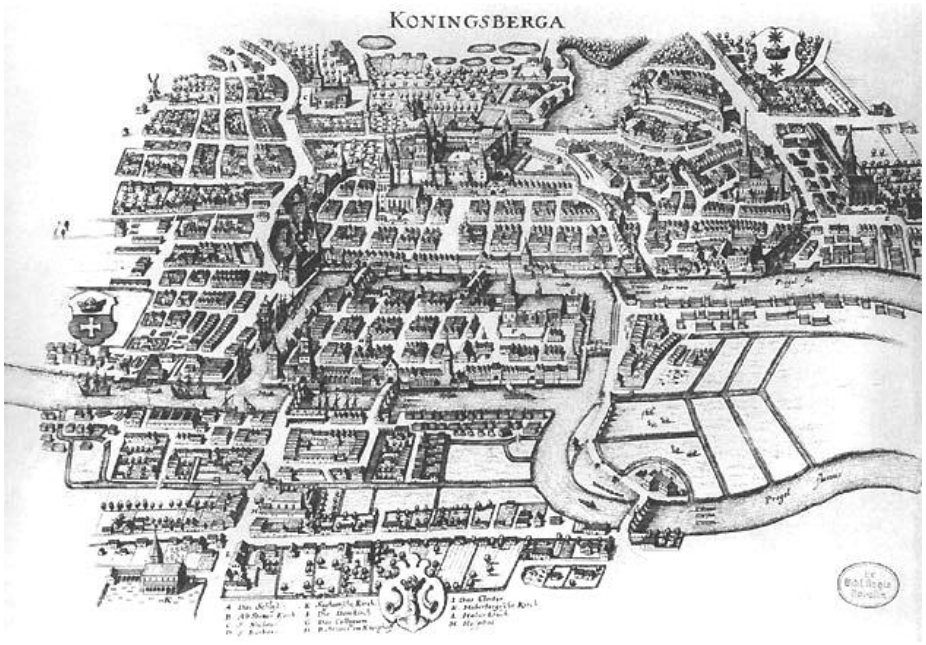

Figure 1: Map of East Prussian Königsberg from 1651 showing river Pregel with two islands (*Kneiphof* and *Lomse*) connected by seven bridges. [8]

the rules are simple and clear, but the problem is the street definition. How do we define a street? Do we include pedestrian paths, parks or highways? For simplicity, we consider a street network of edges and nodes (interception of two or more edges and dead-end roads) accessible by car. Assuming that runner[3] can run oneway roads in both direction, we do not consider road direction. In order to find such network we used Open Street Map API [4] with `drive` layer.

3 Problem

Finding a route for #everystreet challenge roots to the origin of Graph Theory - *the problem of Seven bridges of Königsberg*. Where we are tryning to find a route which visits every edge exactly once. However, in most cases, graph is not *Eulerian*, and we could not find a perfect option where visiting every street exactly once. Therefore, we can reframe this problem to more general and well-known problem of the *Chinese Postman*[4]. The problem tries to find the shortest closed path (or circuit) such that visits every edge of a (closed and undirected) graph.

[3] In an exceptional cases a cyclist *can* go in oneway roads.

[4] Also known as *Postman Tour* or *Route Inspection Problem*

3.1 Seven bridges of Königsberg

Seven bridges of Königsberg is a famous problem solved by Leonard Euler, which sets foundations to todays graph theory[5]. In east Prussian city of Königsberg flowed river Pregel which had two islands and seven bridges connecting them with the mainland, see figure 1. However, the graph representing Königsbergs bridges and islands has four nodes with degrees 3, 3, 3 and 5, Euler showed that there is no such *tour* that visits each bridge once.

Theorem 1. *A connected graph is eulerian if and only if every node has even degree.*

A closed graph is called *eulerian* if it contains Euler tour [7, 6].

3.2 Chinese Postman problem

However, most of the real street networks are very rarely *eulerian* and we often need to find such route which visits every edge, therefore, we relaxed visits of each edge condition to at least once. Finding such route is solving Chinese Postman problem, named after Chinese mathematician Kuan Mei-Ko [6].

3.3 Optimal route

An optimal route visits every edge at least once and total length is smallest among all possible routes. Let w_i be weight or length of edge e_i, then total length of tour $W = n_0 \circ e_0 \circ n_1 \circ ... \circ n_0$ is $\sum_{i \in W} w_i$, where n_i is node.

To find such route we used solution proposed by Edmonds and Johnson [5] in paper *Matching, Euler tours and the Chinese postman* from 1973, which uses matching theory to find such graph where Eulerian tour exists[6].

For the sake to keeping this work short, we will not explain all details, rather state a very rough sketch of the solution described in the Edmonds and Johnsons work [5].

If the street graph G is eulerian (all nodes have even degree) then we are done. If not then we are solving *Chinese Postman problem* and *some* edges will be visited more than once, therefore, let $1 + x_i$ be the number edge e_i is in the tour. Let G' be the graph formed from G by adding extra x_i copies of edge e_i. In the end, all nodes in graph G' *should* have even degree. Finding x_i of the optimal route is equivalent to finding integer $x_i \geq 0$ for each edge e_i of G such that $\sum w_i x_i$ is minimal subject to $\sum_i (1 + x_i) \equiv 0 \ (mod\ 2)$ where the sum is over all edges meeting in node n_j[7], for each node of the graph G.

In order to link this problem to matching problem, let the *node-edge incidence matrix* $(a_{e_i n_j})$, $n_i \in N$ and $e_i \in E$ be defined as:

[5] In Eulers times graph theory was called *geometry of position [9]*

[6] This is the crucial point of the paper, in the original graph we can't find *Euler route*, but we can create induced eulerian multigraph where we can find the route.

[7] $(mod\ 2)$ guaranties that the degree of each node is even.

$$a_{e_i n_j} = \begin{cases} 1 & if\ edge\ e_i\ meets\ node\ n_j \\ 0 & otherwise \end{cases}$$

Then the problem is to find $x_i \leq 0$, $e_i \in E$

$$\sum_{e_i \in E} a_{e_i n_j}(1 + x_i) \equiv 0\ (mod\ 2)$$

and minimalizing $\sum w_i x_i$.

$$\sum_{e_i \in E} a_{e_i n_j} x_i \equiv \sum_{e_i \in E} a_{e_i n_j}\ (mod\ 2)$$

The right side stands for the degree of node n_j, let b_j zero or one based on

$$b_j \equiv \sum_{e_i \in E} a_{e_i n_j}\ (mod\ 2)$$

Therefore, b_j is zero if node n_j has even degree and one if odd. The #everystreet algorithm can be solved by finding such x_i, $e_i \in E$:

$$x_i \leq 0,\ is\ integer, \forall\ e_i \in E \tag{1}$$

$$l_j \leq 0,\ is\ integer, \forall\ n_j \in N \tag{2}$$

$$\sum_{e_i \in E} a_{e_i n_j} x_i - 2l_j = b_j, \forall\ n_j \in N \tag{3}$$

$$z = \sum_{e_i \in E} w_i x_i\ is\ minimalized \tag{4}$$

where l_j is a *adjoining loop*, an edge which connects node with itself.

The rest of the Edmonds and Johnsons paper describes the matching algorithm and solving the linear program, for more details follow *Matching, Euler tours and the Chinese postman*[5]. In #everystreet algorithm we used one of the possible solution described in following section.

4 Algorithm

One of the proposed solution in Edmonds and Johnsons paper is to form a complete graph G_p of odd nodes with lengths corresponding to shortest distance between odd degree nodes in the original graph G, then finding minimal matching in G_p. However, every odd degree is endpoint of shortest path between matched pair, numbers x_i will add up that every node have even degree, satisfying equation (1).

This method should be the optimum to minimalizing z in equation (4) subject to (1), (2) and (3). Let consider that our method is optimal. Numbers x_i can be zero or one[8]. Edges with $x_i = 1$ are those which forms a path between

[8]Otherwise (mod 2) will reduce them.

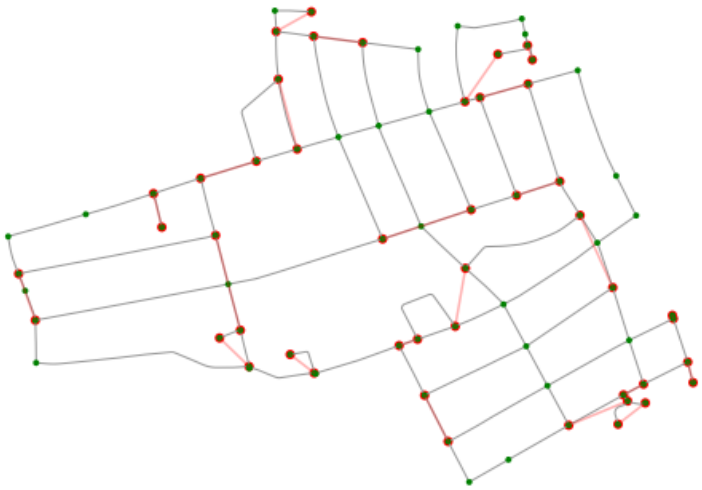

Figure 2: Minimal matching (red nodes and edges) of odd degree nodes plotted on original graph G, The Grange, Edinburgh Scotland

two odd degree nodes. Odd numbers x_i meet odd degree nodes and even meet even degree nodes. Therefore, we can focus only on odd degree nodes and follow their paths to another odd degree node pair. We can remove all edges having $x_i = 1$ except those which forms a path between odd degree nodes. However, the matching in G_p is minimal and paths between odd degree nodes are shortest, therefore, the solution is optimal to (1), (2) and (3) [5].

This algorithm gives us numbers x_i which forms *eulerian multigraph* G' and Euler circuit can be found. To find a Euler circuit we will use Hierholzer algorithm [10]. The #everystreet algorithm states as follow:

1. Find all odd degree nodes ($b_j \equiv 1 \ (mod\ 2)$) in original graph G
2. Find shortest distance between odd degree nodes, using Dijkstra algorithm [11]
3. Form a complete graph G_p of odd degrees nodes with weights of shortest distance between them
4. Find minimal matching in G_p with Edmonds matching algorithm [12, 13]
5. Add edges e_i given x_i according to matching into *eulerian* multigraph G'
6. Find Euler circuit in G' with Hierholzer algorithm [10].

4.1 Algorithm complexity

Chinese Postman problem or #everystreet algorithm can be solved in polynomial time [5]. As we have shown the solution can be compose of three algorithm

Dijkstra, Edmonds matching and Hierholzer. Each of these algorithms has polynomial complexity. Dijkstra algorithm has complexity $\Theta(|E|+|N|log(|N|))$ [11], Hierholzer has $O(|E|)$ and it is purely dependent on number of edges [10]. The most computationally heavy algorithm is Edmonds maximal matching algorithm with complexity $O(|N|^3)$ [12, 13]. Therefore, #everystreet algorithm has complexity $O(|N|^3)$.

Figure 3: Every street of The Grange, Edinburgh, Scotland run according the algorithm

5 Program

The `www.everystreetchallenge.com` is a simple web application where backend is powered with Flask app build on `osmnx`, `networkx` and `network` python packages while frontend is a React app using `Leaflet` package for the map interactions.

6 Conclusion

Aim of this work was to provide an algorithm which finds in given metropolitan area an optimal route for #everystreet challenge which visits every street at least once and total length is shortest among all possible routes. In order to do that, we link this problem to *Chinese Postman problem* and used solution from Edmonds and Johnsons paper to find induced eulerian multigraph based on minimal matching and find Euler circuit. As a result of this work is a simple web application, `www.everystreetchallenge.com`.

References

[1] Beaumont M. (2020), Strava Profile,
https://www.strava.com/athletes/8288853

[2] Gates R. (2019), Every Single Street with Rickey Gates,
https://www.everysinglestreet.com/why

[3] Turner K. (2019), Every Single Street, Strava stories,
https://blog.strava.com/every-single-street-17484/

[4] Open Street Map (2020),
http://openstreetmap.org

[5] Edmonds, J. and Johnson, E.L. *Matching, Euler tours and the Chinese postman.* Mathematical Programming 5, 88124 (1973). https://doi.org/10.1007/BF01580113

[6] Bondy J. A. and Murty U. S. R. (1976), *Graph theory with applications*, ISBN 0333177916

[7] Diestel. R, (2005), *Graph Theory Graduate Texts in Mathematics* Springer

[8] Erben, M., (1652). *Knigsberg 1651.*
https://en.wikipedia.org/wiki/Knigsberg#/media/File:Image-Koenigsberg,_Map_by_Merian-Erben_1652.jpg

[9] Euler L. (1741), *Commentarii academiae scientiarum Petropolitanae*, Volume 8, pp. 128-140.
https://scholarlycommons.pacific.edu/euler-works/53/

[10] Fleischner H. (2016), *Algorithms in Graph Theory*,
https://www.dbai.tuwien.ac.at/staff/kronegger/misc/AlgorithmsInGraphTheory_Script.pdf

[11] Cormen, T. H. (2001) *Introduction to algorithms*, Cambridge, Mass: MIT Press.

[12] Galil Z., (1986) *Efficient algorithms for finding maximum matching in graphs*,
https://www.semanticscholar.org/paper/Efficient-algorithms-for-finding-maximum-matching-Galil/ef1b31b4728615a52e3b8084379a4897b8e526ea

[13] Edmonds J. (2008), *Weighted maximum matching in general graphs.*,
http://jorisvr.nl/files/graphmatching/20130407/mwmatching.py