DOI: https://doi.org/10.53052/9788367652001.02

Mikołaj GRYGIEL[1]

Opiekun naukowy: Stanisław ZAWIŚLAK[2]

NAJKRÓTSZA ŚCIEŻKA I CYKL EULERA – APLIKACJA ORAZ HISTORIA

Streszczenie: W pracy omawiana jest własna aplikacja do wyznaczania najkrótszej ścieżki dla systemu dróg w Polsce. Jest wyposażona w moduł graficzny do prezentacji rozwiązania na mapie. Przedstawiono przykładowe wyniki dla przykładowych startów oraz destynacji. Wyniki porównano z ogólnodostępną usługą Google'a. Zamieszczono też kilka uwag historycznych o omawianym problemie oraz genezie idei cyklu Eulera, w tym o mostach Królewieckich.

Słowa kluczowe: aplikacja, najkrótsza ścieżka, okno graficzne, uwagi historyczne

SHORTEST PATH AND EULERIAN CYCLE IN GRAPHS – SOFTWARE AND SOME HISTORICAL REMARKS

Summary: In the present paper, own software is presented. The program determines the shortest route between the two pointed towns. There is available the graphical panel to present the solution on the screen. The problem was restricted to the simplified system of roads and highways in Poland. Some exemplary results were presented for different input data. The results were compared with open software available via Google. Additionally some historical remarks were given related to the discussed problem as well as the origins of Euler's idea on Konigsberg bridges.

Keywords: own software, shortest path determination, graphical panel, historical remarks

1. Introduction

Graph theory was originated in 1736 via publication in Latin of the paper [8] by Leonhard Euler – it is the common conviction. The mathematical notion of a graph was simultaneously formulated[3]. The graph $G(V,E)$ is a pair of two sets i.e., V set of vertices and E set of edges. Graphs are presented graphically in the plane or other geometrical surfaces. Frequently, a weighted graph is considered $G(V,E,W)$ where in the most simple case a weight function acts on edges; $W: E \to R$ i.e. to each edge e

[1] University of Bielsko-Biala, Faculty of Mechanical Engineering and Computer Science, Master level student: s54409@student.ath.edu.pl

[2] University Professor, PhD, D.Sc., the same affiliation, szawislak@ath.bielsko.pl

[3] In fact other name was used, the word 'graph' was formally introduced a few decades later.

we assign (function assigns) the weight *W(e).* A graph can be used for modeling of many different objects and problems. For example, graphs can represent family ties, flow of media or goods, electrical and mechanical systems as well as networks of roads, railways, gas and water pipelines, electric lines and social networks [16]. In the paper, we consider graphs as models of road connections between particular places or towns – represented by means of vertices. The following problems can be formulated: (a) *SSP* i.e. shortest path problem in a weighted, connected graph. The task consists in determination of the path between two given graph vertices (places, towns) which length in minimal. The path length is a sum of weights of all edges belonging to the considered path. Problem (b) is formulated for a simple, connected graph. It consist in determination of a special sequence of edges called an Eulerian cycle i.e. starting the route in a particular vertex (place) we have to go through each edge exactly once and return to the starting vertex. We say, that in such an object vertices can „be visited" several times. The Eulerian cycle (visits) passes through a particular vertex in such a way that it enters into a vertex and it goes out – one visit. Another similar problem is so called Hamiltonian cycle in which the route passes through every vertex solely ones what usually means that all the graph edges do not belong to the Hamiltonian cycle, simultaneously.

The next step consist in creation of algorithms of a particular graph theory problem. In general we can divide problems in such for which the exact solution can be found/constructed and other for which the output of algorithms gives us an approximate solution. In the second case sometimes artificial intelligence based algorithms are utilized [14]. So, the considered problem was developed and solved according to the pattern: real sketch, abstraction, digitalization, algorithms, theorems on correctness of algorithms, evaluation of effectiveness, applications, comparisons of running times of particular applications.

The goal of the paper is to present own application for the shortest path problem [6] restricted to the selected set of Polish towns. The program is equipped in a graphical window in which the resultant shortest path is presented graphically. The weighs i.e. road distances between the towns were established based on Internet resources. The results are compared with the adequate results obtained via open-source net tools.

Moreover, some historical remarks are given. The work is based upon the bachelor thesis of the author. Analyzing bibliography some interesting comments were spotted which are not commonly known and which are worth for further propagation. Therefore – based upon the given bibliography - some historical remarks will be given.

2. Formulation of the algorithmic approaches to the discussed problems

The graph theory problems are of different nature e.g.: (i) enumeration of families of graphs or subgraphs, (ii) checking a fulfilment of a particular property e.g. planarity or connectivity of a particular graph, (iii) optimization e.g. the shortest path problem [4,7], the minimal spanning tree problem [9 14], (iv) creation/generation of a particular graph object e.g. the Euler or Hamilton cycle, etc. In our case the 'iii' case is solved by means of own software and the case 'iv' is only discussed based upon the references.

We consider the weighted, connected graph representing the simplified road system between particular town of Poland. The route is stored as a series of tows i.e. graph vertices which are taken into account on the path. The total distance is calculated as distances between the particular pairs of vertices in the series. The matrix of connections is generated via the data collected from internet resources. Namely, the data was gathered form conadrogach.pl website. Roads distances were added directly to database. Average speed on them was calculated as 'distance' divided by 'speed'. The database stores the prices of the paid roads, as well.

3. Description of the classical algorithms

3.1. Dijkstra's algorithm

Dijkstra's algorithm [19, 20] serves to find shortest paths between source (starting node) to all other nodes of the graph. It can be applied only to graphs that have non-negative edge weights. The algorithm was created by Edsger Dijkstra in 1956 and published three years later in the paper *"A Note on Two Problems in Connection with Graphs"*. Despite the algorithm's age it is still used in many areas of modern day life like: efficient route optimization in digital mapping services like Google Maps, social network applications or telephone networks. [10, 12, 17]. In book [9], the algorithm is related to Danzig [18], but it is wrong information.

Dijkstra's algorithm pseudocode:

```
Dijkstra(G,l,s):
Input: graph G = (V,E), non-negative edge weight
Output:   foreach   every   node   u   reachable   form s,
d(u)is a distance from s to u

for  every node u in V do
    d[u] = ∞
    prev[u] = null
    d[s] = 0
Q = V
while Q is not empty do
    u = removeMin(Q)
for every edge (u,v) in E
          if d[v] > d[u] + l(u, v) do
                 d[v] = d[u] + l(u, v)
                 prev[v] = u
```

There are further implementations of this algorithm created by using heap or priority queue. The results of those changes is a decrease of time complexity. [5]

3.2. Bellman–Ford algorithm

In 1958 the algorithm solving the shortest path problem was published by Richard Bellman. The algorithm finds the shortest paths from source (starting node) to all other nodes of the graph but in comparison to Dijkstra's algorithm it can be applied to the graph with negative edge weight values. [5]

Bellman–Ford algorithm pseudocode:

```
Bellman-Ford(G,l,s):
Input: graph G = (V,E)
Output: foreach node u reachable from s, d(u) is a distance
from s to u

for every node u in V do
    d[v] = ∞
    prev[v] = null
    d[s] = 0
repeat |V|-1 times:
    for every edge (u,v) do
                  if d[v] > d[u] + l(u,v) do
                         d[v] = d[u] + l(u,v)
                         prev[v] = u
```

However, according to paper "Comparative Analysis between Dijkstra and Bellman- Ford Algorithms in Shortest Path Optimization", it is slower than Dijkstra's algorithm. The paper also states that Dijkstra's algorithm's speed advantage over Bellman–Ford algorithm is a main reason of it common usage in modern applications. [1, 13]

4. Own software for SPP in case of two criteria

Web application was developed by the authors of this article, which is solving the shortest path problem for Poland road network in case of two criteria separately. The first criterion is distance in kilometers and the second one is time in hours [3]. Application also has functionality to calculate costs depending on vehicle's fuel usage and current fuel price in PLN. The application can visualize the route, as well. The created software uses Dijkstra's algorithm which works with the adjacency matrix representation of the graph to find the shortest path between nodes which are cities in Poland.

4.1. Description of the project's application

Application uses SQL database storing the specific 47 cities in Poland – nodes of the graph and 127 roads connecting this set of cities – edges of the graph. Web application was created using web framework AST.NET combined with JavaScript library Leaflet and Leaflet Routing Machine.

4.2. Results' Comparison

Results of the Application and Google Maps are shown in this subsection. For example in Fig. 1. The route between Bydgoszcz and Zakopane is presented.

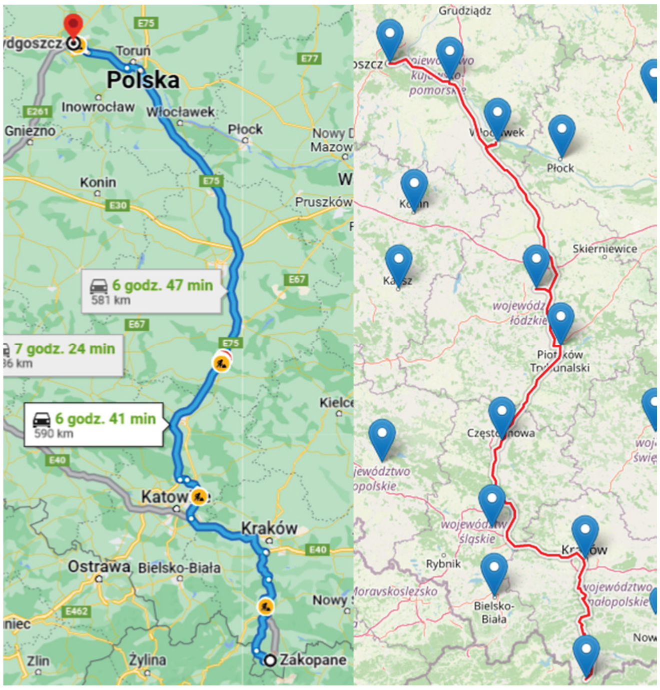


Figure 1. Comparison of the results of Google maps (left side) with time criteria in our application (right side)

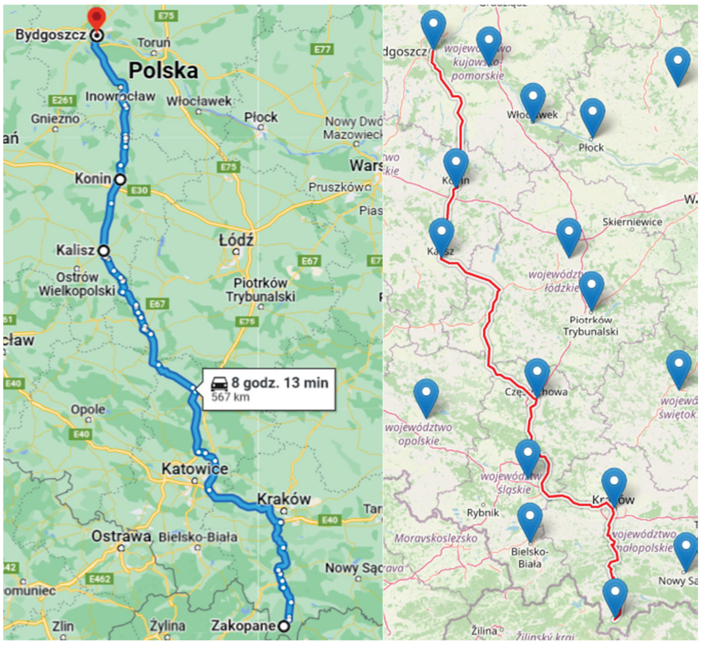


Figure 2. Comparison of the results of Google maps after making it more similar (left side) with distance criteria in our application (right side)

As we can see there are no major differences between the two results. The shortest path in case of distance that Google maps can find between picked Cites: Bydgoszcz and Zakopane is different from obtained by our application and it equals 581 km. When we tried to make Google map road more alike to the one form our application the Google map distance was reduced to 567 km.

Unfortunately the author's application has two major problems. The first one is that there are only 47 cities and 127 roads that connect them, which limits the amount of possible routes and lower precision of outcomes. The second problem is that the longer paths always deviate from the route only to go thought nearby city centers instead of driving via beltways, causing the slightly less optimal results at longer routes.

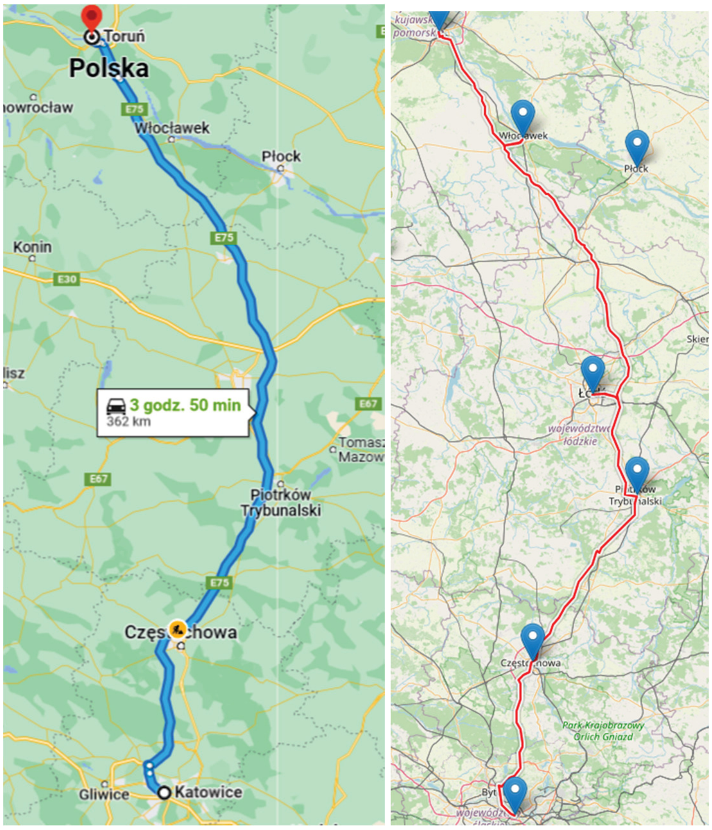


Figure 3. Comparison of the results of Google (left side) with time criteria in our application (right side)

The figure above shows some differences in the Google Maps and the created application's route, especially when the route is going close to cites. The conclusion may be that the author's application main issue is the Data on which the algorithm operates – nodes are the city centers and edges are roads between them. However that problem can be solved by changing the graph structure – the crossroads would replace city centers as nodes, but it will result in huge increase in stored data and time needed for algorithm to work.

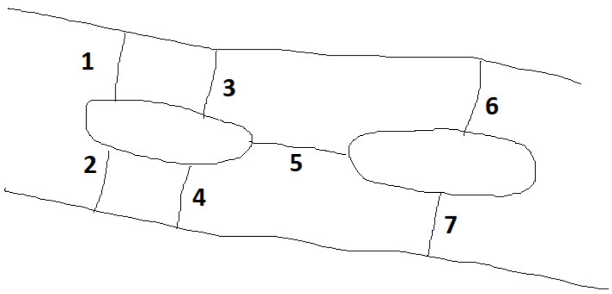
Table 1. Result comparison. Vehicle fuel consumption is considered as 6l for every 100km with Fuel price for 14th September, 2022. [21]

Case	Distance [km]		Addditional information			
	Application	Goole-based	Assumed speed $[\frac{km}{h}]$	Cost of fuel [PLN]	Evaluated by application	
					Total cost	Time of journey
Fig. 1	602	590	69	236	236	8h 44 min
Fig. 2	572	567	61	224	224	9h 23 min
Fig. 3	394	362	77	154	154	5h 8 min

5. Koenigsberg/Kaliningrad Bridges – some historical remarks

The historical facts related to the problem of Koenigsberg Bridges can be found in some papers e.g. [2, 11, 16].

Table 2. Koenigsberg Bridges (author of the phots: Stanisław Zawiślak)



Scheme of islands and bridges across the Pregel river

Today bridges no 1 and 2 are united into one.


Bridge no 5 between islands.



Two islands can be seen,

Bridge no 6.


United bridges 1 & 2 (background)

Bridge no 7 (Bridges no 3 and 4 do not exist, nowadays.)

The third mentioned paper [16] is extremely interesting because it contains hypotheses and proofs related to the formulation of the problem and explains how it was delivered to Euler who lived within the years 1707 to 1783. Author wrote that there is not any evidence that Euler visited Koenigsberg personally! The problem was announced to him by Carl Gottlieb Ehler who was a scientists and a diplomat living in Danzig, but having co-workers from Koenigsberg. It should be emphasized that Euler was a professor at the University in St. Peterburg– the capital of then Russian Empire. Ehler was a Gdansk mathematician and astronomer. He visited the Tsar Court and met Euler by this occasion. Ehler delivered a problem to Euler. The history is based upon the artefact i.e. fifteen original letters from Ehler to Euler and five from Euler to Ehler (1735-1742) – found in Tartu Library (Estonia). One of the letters contains the commonly known layout of the Pregel river with two islands. So, the paper [16] is highly recommended for reading.

The author of the discussed paper describes also a visit of Euler in Warsaw travelling from Berlin to St. Petersburg.

Today, the Koenigsberg's name is Kaliningrad. The town is a local capital of Russian territory i.e. province – enclave between Poland and Lithuania. In Table 2, the contemporary bridges' photos are presented.

6. Conlusions and final remarks

In the paper two topics were described. Own web application was described. Some remarks on the problem of Konigsberg Bridges were presented. The historical background was highlighted based upon the analyzed references.

The application calculates and shows the shortest route between the towns in Poland. The graphical image of the route is presented in special illustrative window. The Dijkstra's algorithm was utilized. The comparisons with the commercial tools confirmed that our software gives the results which are comparable with the professional ones.

The differences are caused by our simplified model of the road network. It is connected with the assumptions on number of towns as well as on number and types of roads – which were taken into account. For simplicity the ring roads around big cities are omitted in input data set. Nevertheless the results are highly compatible.

REFERENCES

1. ABUSALIM S.W., IBRAHIM R. et.al.: Comparative analysis between Dijkstra and Bellman-Ford algorithms in shortest path optimization. IOP Conference Series: Material Science and Engineering, 917(1), (2020).
2. ALEXANDERSON G.: About the cover: Euler and Königsberg's Bridges: A historical view. Bulletin of the American Mathematical Society, 43(2006)4, 567-573.
3. CLIMACO NAMORADO J. C.; MARTINS VIEIRA, E. Q.: A bicriterion shortest path algorithm. European Journal of Operational Research, 11(1982)4, 399-404.

4. CHERKASSKY B. V.; GOLDBERG A. V.; RADZIK T.: Shortest paths algorithms: Theory and experimental evaluation. Mathematical programming, 1996, 73.2: 129-174.
5. DASGUPTA S., CHRISTOS P., VAZIRANI U.: Algorithms, (in Polish) WN PWN, Warszawa (2012).
6. DEO N., PANG Chi-Yin: Shortest-path algorithms: Taxonomy and annotation. Networks, 1984, 14.2: 275-323.
7. DREYFUS S. E.: An appraisal of some shortest-path algorithms. Operations research, 1969, 17.3: 395-412.
8. EULER L.: Solutio problematis ad geometriam situs pertinentis, Comment. Academiae Sci. I. Petropolitanae, 8, 128-140 (1736).
9. FILIPOWICZ B.: Operational research. Chosen calculation methods and algorithms (in Polish), Poldex (1997).
10. GALLO G., PALLOTTINO S.: Shortest path algorithms. Annals of operations research, 1988, 13.1: 1-79.
11. HOPKINS B., WILSON R. J.: The Truth about Königsberg, The College Mathematics Journal, 35(3), 198-207 (2004), DOI: 10.1080/07468342.2004.11922073
12. JOHNSON D. B.: A note on Dijkstra's shortest path algorithm. Journal of the ACM (JACM), 1973, 20.3: 385-388.
13. MAGZHAN K., JANI H.M.: A review and evaluations of shortest path algorithms. International journal of scientific and technology research, 2(6), 99-104 (2013).
14. PAGACZ A., RAIDL G., ZAWIŚLAK S.: Evolutionary approach to constrained minimum spanning tree problem–commercial software based application. Evolutionary Computation & Global Optimization, 331-341 (2006).
15. SZNAJDER R.: On known and less known relations of Leonhard Euler with Poland. Studia Historae Stientiarum, 15, 74-110 (2016).
16. ZAWIŚLAK S., RYSIŃSKI J. (ed.): Graph-Based Modelling in Science, Technology and Art. Springer (2022).
17. ZHAN, F. B., NOON, C. E.: Shortest path algorithms: an evaluation using real road networks. Transportation science, 32(1) 65-73, (1998).
18. Web page: *https://en.wikipedia.org/wiki/George_Dantzig*; (Accessed 15.7.2022).
19. Web page: *https://mathshistory.st-andrews.ac.uk/Biographies/Dijkstra/*; (Accessed 21.7.2022).
20. Web page: *https://www.geeksforgeeks.org/applications-of-dijkstras-shortest-path-algorithm/*; (Accessed 21.7.2022).
21. Web page: *https://pl.fuelo.net/?lang=en*; (Accessed 16.9.2022).