Computation of isochrone regions for electric vehicles

Stud. Manvydas Sokolovas

Vilnius University
Faculty of Mathematics and Informatics
Institute of Computer Science
Department of Computational and Data Modeling
Universiteto str. 3, LT-01513 Vilnius, Lithuania
manvydas.sokolovas@mif.stud.vu.lt
https://mif.vu.lt/lt3/en/about/structure/institute-of-computer-science

Abstract. Electric vehicles (EV) have received a lot of attention in recent years. They are treated as a promising alternative to transport powered by internal combustion engines. Thus various EV routing related questions are getting even more relevant nowadays. This type of questions differ from standard routing problems because of their additional complexity. Electric vehicles route planning algorithms also have to consider a number of additional constrains specifically applied only for EVs, when conventional routing systems usually try to minimize only time or distance to the destination. In this paper we study multi-objective shortest path search algorithms with various type of weights which represent different properties of a road network. Road length, speed limit, energy consumption for a given path were taken into account. We also analyse the impact of time-varying road type-dependent speed coefficient on results of the shortest path algorithm. For sake of simplicity only synthetic low scale data was used for shortest path algorithms comparison. Only free open source technical tools and data were used.

Keywords: Multi-objective shortest paths · Electric vehicles · Graph theory.

1 Introduction

Sustainable transport is an integral part on the path to the cleaner environment. Thus various supportive government policies for Electric Vehicles (EV) together with people's desire to be more eco-friendly and to keep pace with technology, significantly increased sales of electric cars over the decade [1].

However, majority of electric vehicles have limited battery capacity. This means that their range until the driver have to stop to charge is very limited too. Due to this reason charging can take up a significant part of a total trip time or if there are no or very rare charging stations in the chosen route driver could start feeling range anxiety - uncertainty if battery level is enough to reach the destination. Due to this reason it could be necessary to detour from the most direct route and it is very important to carefully plan a trip route in advance by including electric vehicle battery constrains. Which makes various EV routing related questions even more relevant nowadays.

One of these questions could be how to make efficient EV routing algorithm which takes into account battery limitations, charging stations, road, weather conditions, driving habits. This type of algorithm could help to reduce drivers range anxiety by finding more energy efficient routes instead of searching only for the fastest paths.

Related Work There are already a lot of researches done in route planning field. Many of them uses Dijkstra algorithm [5]. It is a very efficient method to work with weighted graphs for finding the shortest path between two nodes. Not an exception and EV routing field.

Baum et al. [2, 3] propose new efficient EV route planning models which main focus is on realistic charging stops inclusion. Their models try to minimize overall trip time including time spent at charging stations. Also various charging power and battery swapping stations are taken into account.

Perger and Auer [6] shows that there is significant connection between roads topography and electric vehicles energy consumption, routes with high topographical variation tend to drain battery faster. Cauwer et al. [4] observes that the shortest path is not necessarily the most energy efficient one and gains in energy efficiency could significantly increase travel time.

Altogether, in this paper we work on multi-objective shortest path search algorithms with various type of weights which represent different properties of a road network and electric vehicle. We also present different ways of multiple weights inclusion in the road network and how it affects shortest path algorithm results.

2 Road network

In simple words a network is usually a big system consisting of multiple similar objects where some of them are connected to each other in order to allow some kind of movement or communication between them. It could be applied in many fields, some of them are telephone network, computer network or road network. The latter we will use as a base of our study.

We model road network as a graph $G = (V, E)$ where V represents vertices (also called nodes), and E—edges (also called links), the latter always connect some nodes. Similarly as roads could be one-way or bidirectional, graphs could also be directed or undirected depending if edges have orientations or not. Directed graph is a graph which edges have orientations. For example (see Fig. 1) an arrow from vertex A to B means that it is possible to travel from point A to B but not other way around.

Furthermore, typically, each edge is a 3-tuple $e = (u, v, w)$ where w represents a weight that is distance or travel time. But routing for EV requires additional parameters like battery capacity, current state of charge, road topography or energy consumption. In our study as a second weight to the journey time we use energy consumption for a given edge. For example (see Fig. 1) we can travel from the vertex B to C for the cost of 1 both for time and energy units each and the trip back takes the same time but 3 times more energy. Thus in road network it could represent hill, it does not take a lot of energy for EV to go down the hill (in some cases it could even regenerate some energy) but it takes a lot more energy to drive the same uphill.

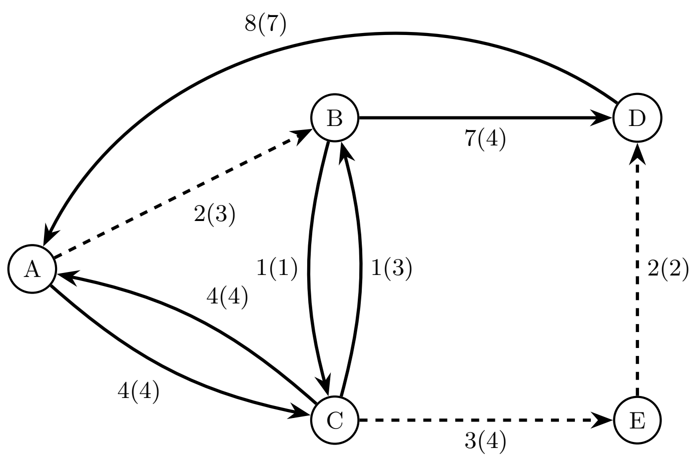

Fig. 1. Simple road network representation as a graph with multiple weights and multiple street types (solid—Highway, dashed—City)

3 Dijkstra - Shortest Path

3.1 Shortest Path Algorithms

Let $G = (V, E)$ be a directed graph with weights w, source $s \in V$ and target $t \in V$. The point-to-point shortest path problem is used to find the route between s and t for which the sum of weights w is as low as possible. The one-to-all shortest path problem is used to find the shortest paths between s and all other vertices. In this study we focus on the one-to-all shortest path problem.

Dijkstra There are already a lot of researches done in the field of route planning and it could be mentioned that Dijkstra algorithm [5] is usually used as a standard approach for one-to-all shortest path problem.

Dijkstra algorithm starts from the source vertex s and group all vertices to explored and yet unexplored, also it initially sets the distance to unvisited nodes as

infinity. Then from the current vertex it updates the length l (sum of w) to each directly connected node and goes to the one with the lowest weight. Furthermore it continues to check directly connected vertices and goes to the closest one. It saves all explored vertices with length values for it to the list and if finds shorter path to any node–updates its value.
Let $G = (V, E)$ be the road network represented in the Figure (1), with weights $w = (w_1, w_2)$, which are shown on top of the curves. For this example we will only use trip time as a weight. Step by step example of Dijkstra one-to-all shortest path algorithm is shown in the Table (1). The last line in the table represents final shortest path algorithm results. For example the value D^{8E} means that it cost 8 time units to travel from the source vertex A to the destination D and the path is A $\xrightarrow{2}$ B $\xrightarrow{1}$ C $\xrightarrow{3}$ E $\xrightarrow{2}$ D = 8.
Also if we search for the optimal route based on streets' efficiency (numbers in the brackets in Figure 1) Dijkstra will give different results: $A^0B^{3A}C^{4A}E^{7B}D^{8C}$ and now the preferable route from vertex A to D is A $\xrightarrow{3}$ B $\xrightarrow{4}$ D = 7. This route does not go through C and E nodes anymore.

Table 1. Dijkstra shortest path algorithm example

Source = A	
Explored	Unexplored
	A$^0B^{\infty}C^{\infty}D^{\infty}E^{\infty}$
A^0	B$^{2A}C^{4A}D^{\infty}E^{\infty}$
A$^0B^{2A}$	C$^{3B}D^{9B}E^{\infty}$
A$^0B^{2A}C^{3B}$	D$^{9B}E^{6C}$
A$^0B^{2A}C^{3B}E^{6C}$	D^{8E}
A$^0B^{2A}C^{3B}E^{6C}D^{8E}$	

3.2 Multi-objective Shortest Path Algorithms

In this section we will investigate different approaches of multiple weights inclusion in the road network and how it affects shortest path algorithm results. We will use the road network presented in the figure 1. The first weight represents time needed to complete given edge and the second weight (number in brackets) represents energy consumption for a given edge.
Thus, at first we show how multiple weights could be implemented as a simple linear function which represents driver preferences for what to optimize when route planning. Secondly, we show that different routes between two vertices could be chosen depending on the time of the day.

Linear Combination of Weights In this section we discuss the case if we could choose significance of each weight type and combine all weights on one edge to single number and then run shortest path algorithm on it.

We show that this type of weights combination could give different results depending on chosen parameters for each weight.
Let $G = (V, E)$ be the road network represented in the Figure (1) with weights $w = (w_1, w_2)$, which are shown on top of the curves. Then linear combination of weights w_1 and w_2 could be described as:

$$w_{total} = a \times w_1 + b \times w_2, \tag{1}$$

where a and b are non negative numbers. If $b = 0$, then the shortest path algorithm will only minimize w_1 and give the same results as in the Table (1). The other way around is when $a = 0$, then w_2 will be minimized.

Three examples with different a and b parameters were presented (Table 2). All of them gave similar results, vertexes were visited. The only difference from that the vertex D is always reached in the same order as optimizing energy efficiency. For future work it would be recommended to test more parameters combinations, especially with higher $\frac{a}{b}$ or $\frac{b}{a}$ ratios.

Table 2. Dijkstra shortest path algorithm example with linear combination of weights

a	b	Dijkstra results
1	1	$\mathrm{A}^0 B^{5A} C^{7B} E^{14C} D^{16B}$
1	2	$\mathrm{A}^0 B^{8A} C^{11B} E^{22C} D^{23B}$
3	1	$\mathrm{A}^0 B^{9A} C^{13B} E^{26C} D^{34B}$

Weights Multiplier The idea is to create non-negative speed multiplier m which reduces average speed v (increases trip time) for given edges based on street type and time of the day. This type of factor represents traffic intensity in particular type of streets during the day and is simply applied (2) by multiplying base speed from multiplier.

$$v_{new} = m \cdot v_{base} \tag{2}$$

Factor m values are provided in the Table 3. The lowest speed multipliers (the highest average speed decrease) are assigned for the rush hours in the morning, when people travel to work and in the afternoon when they drive back home. City streets are also assigned with larger average speed decrease than highway.

Let $G = (V, E)$ be the road network represented in the Figure (1) with weights $w = (w_1, w_2)$, which are shown on top of the curves. Solid lines represent highways and dashed lines represent city streets.

All possible combinations were tested (Table 4) and it was observed that in the most cases path to the vertex D goes through node B, except at night, when standard, not rescaled weights were used. It was also observed that during

Table 3. Speed multiplier values by street type and time of the day

	Time of the day						
Street type	6 to 10	10 to 15	15 to 16	16 to 19	19 to 22	22 to 23	23 to 6
Highway	0.8	1	0.8	0.8	0.8	1	1
City	0.3	0.6	0.6	0.3	0.6	0.6	1

the rush hours, when average speed coefficient reaches the lowest point for city streets, it is faster to go from A to C than from A to B.

Table 4. Dijkstra shortest path algorithm example with weights dependence on street type and time

Time	Dijkstra results
6 to 10	$\mathrm{A}^0 C^{5A} B^{6.25C} E^{15C} D^{15B}$
10 to 15	$\mathrm{A}^0 B^{3.3A} C^{4A} E^{9C} D^{10.3B}$
15 to 16	$\mathrm{A}^0 B^{3.3A} C^{4.55B} E^{9.55C} D^{12.05B}$
16 to 19	$\mathrm{A}^0 C^{5A} B^{6.25C} E^{15C} D^{15B}$
19 to 22	$\mathrm{A}^0 B^{3.3A} C^{4.55B} E^{9.55C} D^{12.05B}$
22 to 23	$\mathrm{A}^0 B^{3.3A} C^{4A} E^{9C} D^{10.3B}$
23 to 6	$\mathrm{A}^0 B^{2A} C^{3B} E^{6C} D^{8E}$

4 Conclusion

Creating efficient routing algorithm for electric vehicles is an important task. It should take into account multiple EV related constrains like limited battery capacity, charging stations, energy consumption by given road parameters. Successful algorithm creation could help EV drivers plan their trips better by having an option to compare the fastest and the most efficient routes and choose the one which suits them best. In this paper we present how multiple weights could be implemented as a simple linear function which represents driver preferences for what to optimize when route planning. Also, we show that different routes between two vertices could be chosen depending on the time of the day.

Future work For future work it would be recommended to test shortest path algorithm with linear weights combination including more different parameters, especially with higher $\frac{a}{b}$ or $\frac{b}{a}$ ratios. Also, it would be greatly recommended to test these algorithms on real world data.

References

1. Agency, I.E.: Global ev outlook 2020. https://doi.org/https://doi.org/https://doi.org/10.1787/d394399e-en

2. Baum, M., Dibbelt, J., Gemsa, A., Wagner, D.: Towards route planning algorithms for electric vehicles with realistic constraints. Computer Science - Research and Development **31**(1), 105–109 (2016). https://doi.org/10.1007/s00450-014-0287-3, https://doi.org/10.1007/s00450-014-0287-3
3. Baum, M., Dibbelt, J., Gemsa, A., Wagner, D., Zündorf, T.: Shortest feasible paths with charging stops for battery electric vehicles. Transportation Science **53** (07 2019). https://doi.org/10.1287/trsc.2018.0889
4. De Cauwer, C., Verbeke, W., Van Mierlo, J., Coosemans, T.: A model for range estimation and energy-efficient routing of electric vehicles in real-world conditions. IEEE Transactions on Intelligent Transportation Systems **21**, 2787 – 2800 (06 2019). https://doi.org/10.1109/TITS.2019.2918019
5. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numerische mathematik **1**(1), 269–271 (1959)
6. Perger, T., Auer, H.: Energy efficient route planning for electric vehicles with special consideration of the topography and battery lifetime. Energy Efficiency **13**(8), 1705–1726 (2020). https://doi.org/10.1007/s12053-020-09900-5, https://doi.org/10.1007/s12053-020-09900-5