âĂć

# Fastest-Path Algorithm for Electrical Vehicles

A. B. Eriksen
Institute of Computer Science
Aalborg University
xxx@xxx.dk

M. A. Madsen
Institute of Computer Science
Aalborg University
xxx@xxx.dk

M. M. Andersen
Institute of Computer Science
Aalborg University
xxx@xxx.dk

S. B. Jensen
Institute of Computer Science
Aalborg University
xxx@xxx.dk

## ABSTRACT

## Keywords

Shortest-path algorithm with resource constraints, Electrical vehicles, Road-network, Route planing, Fisse

## 1. INTRODUCTION

As the adoption rate of electric vehicles increases, the need for a better and more complex route planing solution appears. Electrical vehicles generally have a shorter range as well as a significantly slower recharge time, compared to traditional gasoline cars. As the recharge time have become a significant factor in the total travel time, the time spend recharging should be taken into account by a route planning system, to accommodate for an otherwise large margin of error. Time spend recharging batteries is inherently variable, as the charge rate is affected by both the charge station's charge rate as well as the battery's current charge. When planing a fastest-path route for an electric vehicle, it is therefore paramount to be able to incorporate charge station's charge rates and electric vehicle's battery [1].

Another important aspect is the electric vehicle's energy consumption. With the now significant time spend recharging, a high energy consumption rate while driving, can in turn result in a slower route. The energy consumption of electrical vehicles generally increases polynomially with speed so this should also be taken into account by the route planning system.

It should be clear that no traditional shortest path algorithm is able to accommodate for the variables and relationships between the travel speed, the charge rate and the current charge of the electrical vehicle. In this article we present two different approaches to solving the problem of optimising a route plan for electrical vehicles. One is a linear program

Go into more detail about this solution

The other is naive solution which uses heuristics

Go into more detail about this solution

Both approaches are analysed on their running time and reliability, based on a real world road network as input.

## 2. NOTATION

In this section we will briefly explain the mathematical notation and model used in the paper. The graph representation consists of edges which represents roads and vertices which represents either charge stations or road intersections. We define a **road network** as an ordered pair $G = (V, E)$ comprising of a set $V$ of vertices together with a set $E$ of edges. Where $V$ is a finite set and $E$ is a binary relation on $V$. We further define the following functions:

$$E_d(u, v) \rightarrow d$$

$$E_v(u, v) \rightarrow v$$

These are partial functions defined by the edge $(u, v)$ and respectively returns a distance or a speed limit.

Each vertex is then defined as a *charge station*, described by the following partial function:

$$V_c(v) \rightarrow c$$

A road intersection is simply a vertex with charge rate $c = 0$, while a charge station is a vertex where $c > 0$. An electrical vehicle is specified by two parameters: It's battery capacity ($kWh$), consumption rate ($kWh/km$). The consumption rate is given by the following function:

$$C_R(v) = av^2 + bv + c$$

where $v$ is the speed of the vehicle.

Add notation about path, more about vehicle and maybe describe a solution

### 2.1 Shorter Not Equal Faster

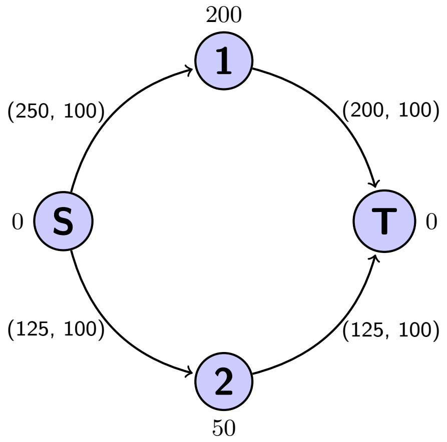

**Figure 1: A simple road-network with starting point s and end point t**

A shorter path does not necessarily mean a faster path for electrical vehicles. This is partly due to the fact that an electrical vehicles uses exponentially more energy as its speed increases but also due to the fact that charge times on charge stations varies a lot. Driving a longer path with a fast charge station on can therefore turn out to be a faster choice than driving a shorter path with a slow charge station on. This is illustrated in Figure **??**. In the example we assume our car has a battery capacity of 100 kWh and a energy-usages of 0,4 kWh/km. Path 1 consist of two edges with distance = 250 km and speed limit = 100 km/t and a charge station with a charge speed of 200 kWh/t. Path 2 consist of two edges with distance 200 km and speed limit 100 km/t and a charge station with a charge speed of 30 kWh/t. The total time of each path:

**Path 1:** route time = ((250km + 250km) / 100km/h) + (100kWh / 200kWh/h) = 5,5h

**Path 2:** route time = ((200 km + 200 km) / 100 km/h) + (60 kWh / 30 kWh/h) = 6 h

## 2.2 Optimization algorithm

The main goal of this paper is to design an algorithm to find the fastest path for an EV, considering both the time spent driving and the time spent charging. The hard thing about finding the fastest path is and discussed in section **??** there is no optimal substructure for this problem. The solution model we propose consists of continuous pruning and then solving the problem for the path with the lowest $T_{pmin}$, which is the time it takes to drive a path a maximum speed, without charging, until we end up with just a single path which is the fastest path. An initial way to prune is to find a path $P$ from $s$ to $t$, and solve it optimally. Solving $P$ optimally will return the time it will take to follow this path, we call this value $T_p$. We then observe that if another path $P'$ driven with max speed, ignoring the energy constraint, will take time $T_{p'}$. If $T_p \leq T_{p'}$ we can simply ignore the path $P'$, since the actual time used to drive $P'$ will be greater then or equal to $T_{p'}$. If we find a path which can be driven faster than $T_p$ we update $P$ and $T_{p'}$. When it comes to determining how a path is optimally solved, this will be discussed later in section **??**. Using pruning and a way to find out how a path is optimally solved, we can produce an algorithm with finds the fastest path as follows.

```
function FASTESTPATH($G, s, t$)
    $p \leftarrow$ initialpath($G, s, t$)
    $T_p \leftarrow$ solve($p$)
    $paths \leftarrow$ getpaths($G, s, t, T_p$)
    repeat
        $T_{p1} \leftarrow$ solve(paths[1])
        if $T_{p1} < T_p$ then
            $T_p \leftarrow T_{p1}$
            $p \leftarrow paths[1]$
        end if
        $paths.remove(paths[1])$
        for all $p \in paths$ do
            if $T_p \leq p.min$ then
                $paths.remove(p)$
            end if
        end for
    until $paths.size = 0$
    return $(p, T_p)$
end function
```

Where *initialpath*$(G, s, t)$ is the initial path we choose to solve e.g. shortest path. *solve*$(p)$ is a function which estimates the optimal solution to a path i.e. optimal charging times for each charging station and driving speed for each road segment in a path. *getpaths*$(G, s, t, T_p)$ is a function which finds all paths in $G$ between $s$ and $t$ with a max travling time of $t$. Having the algorithm we now investigate how to estimate an optimal solution for a path.

## 2.3 Optimisation problem

Finding an estimated optimal solution to a path, the objective function of the optimisation problem is concerned with time, and can be expressed as follows.

$$\underset{v_{1\ldots n}, charge_{1\ldots n}}{\text{minimize:}} \quad \sum_{i=1}^{n} \frac{Distance_i}{v_i} + charge_i \tag{1}$$

The purpose of this function is to minimize the time used on driving $\frac{distance_i}{v_i}$ plus the time used charging $charge_i$ for the entire path. The problem needs to be solve accordingly to a set of constraints, the constraints can be formulated at follows:
On all roadsegments the speed of the EV must be within the speedlimit, $v_i$ is the speed speed on roadsegment $i$, $lb_i$ and $ub_i$ is the lower and upper bound of the speed limit.

$$\forall_{i \in 1\ldots n} : \ lb_i \leq v_i \leq up_i \tag{2}$$

The battery level of the EV needs to be between 0 and the max capacity of the battery. This constraint is split up in two constraints, one ensuring that the EV will have enough energy to drive each road segment and one ensuring that the EV can not over charge at any charging station. $C_R(v)$ is the the energy consumption function of the EV.

$$\begin{split}\forall_{i \in 1\ldots n} : \ 0 \leq \sum_{j=1}^{i} chargerate_j * charge_j - \\ \sum_{j=1}^{i} distance_j * C_R(v_j) \leq maxbattery\end{split} \tag{3}$$

$$\begin{aligned}&\forall_{i\in 1\ldots n-1}:\ 0\leq\sum_{j=i}^{i+1}\mathit{chargerate}_j*\mathit{charge}_j-\\&\mathit{distance}_i*C_R(v_i)\leq\mathit{maxbattery}\end{aligned}\tag{4}$$

It should also not be posible for the EV to spend a negative amount of time at a chargestation.

$$\forall_{i\in 1\ldots n}:\ 0\leq\mathit{charge}_i\tag{5}$$

This optimization problem is almost a linear programming problem. We to however suffer from a non-linear constraint $\mathit{distance}_i * C_R(v_i)$ since the consumption rate $C_R(v)$ of the EV is not linear. The function can however be estimated using linearization. Using linearization we end up with the following linear programming problem.

minimizing the sum of time spend driving and the sum of time spend charging.

$$\underset{\mathit{speed}_{1\ldots n},\mathit{charge}_{1\ldots n}}{\text{minimize}}\quad\sum_{i=1}^{n}\frac{\mathit{Distance}_i}{\mathit{speed}_i}+\mathit{charge}_i\tag{6}$$

subject to:

$$\forall_{i\in 1\ldots n}:\ \sum_{j=1}^{m}\mathit{Selectedlines}[i,j]=1\tag{7}$$

$$\forall_{i\in 1\ldots n,j\in 1\ldots m}:\quad 0\leq\mathit{Selectedlines}[i,j]\leq 1\tag{8}$$

$$\forall_{i\in 1\ldots n,j\in 1\ldots m}:\ \mathit{speed}[i,j]\leq\mathit{Selectedlines}[i,j]*\mathit{Points}1[i,j]\tag{9}$$

$$\forall_{i\in 1\ldots n,j\in 1\ldots m}:\ \mathit{speed}[i,j]\geq\mathit{Selectedlines}[i,j]*\mathit{Points}2[i,j]\tag{10}$$

$$\begin{aligned}\forall_{k\in 1\ldots n}\ :\ 0\leq\sum_{i=1}^{k}\mathit{chargerate}[i]*\mathit{charge}[i]\\-\sum_{i=1}^{k}\mathit{distance}[i](\sum_{j=1}^{m}\mathit{LinesA}[i,j]*\mathit{speed}[i,j]\\+\sum_{j=1}^{m}\mathit{Selectedlines}[i,j]*\mathit{LinesB}[i,j])\leq\mathit{maxbattery\ capacity}\end{aligned}\tag{11}$$

Describe a more naive solution based on heuristics

# 3. EXPERIMENTS

In this section we present how the experiments have being conducted.

## 3.1 Setup

To facilitate the experiments, we've had to find a real world road network graph dataset which has road distances, speed limits and charging stations of varying power output. Such dataset does not openly exist to our knowledge, instead we have used OpenStreetMaps, henceforth *OSM*. One can read more about it at `http://www.openstreetmap.org/about`. OSM has a concept of *Ways* and *Nodes*. Ways represent geographical planar objects, e.g. roads. A node is a geographical point (latitude,longitude). A way is then constituted of several nodes, and some tags which describe meta-information such as the name of the way, and what type of construct it is. As such, it follows that if way $A$ and $B$ intersect in node $x$, they will share the node $x$, known as an intersection in the road network.

These objects can easily be converted into a road network structure, by filtering ways of type *road* and use these ways as edges in the road network. To get speed limits we can derive general speed limits from the type of the roads, OSM carries such information as motorway,residential,tertiary for roads. Since we are not interested in the way a road curves, we ignore intermediate points on a way, which do not correspond to an intersection.

## 3.2 Citations

# 4. CONCLUSIONS

## 4.1 References

# 5. REFERENCES

[1] Nobody Jr. My article, 2006.