Routing-Based Methodology for Evaluation of Electric Vehicle Supply Equipment (EVSE) Networks

Aaron I. Rabinowitz

Introduction

When discussing transportation networks, one must consider several linked and mutually supporting networks including road, rail, marine, and air infrastructure in addition to those energy and information networks which allow for vehicles to function. Multiple support systems are required for the efficient and economical operation of vehicles which underpins modern economies and standards of living. The various networks are coincident at a subset of nodes but, otherwise, may be operated in various manners by dissimilar stakeholders in service of divergent objectives. none of the related networks is a subset of any other but all share dependence relationships which may be asymmetrical.

The mode of vehicles in today's transportation sector, whether passenger or cargo oriented, are Internal Combustion Engine Vehicles (ICEVs) but a growing number of these vehicles are being supplanted by Plug-in Electric Vehicles (PEVs). This shift necessitates a re-alignment of the networks underpinning travel as the nodes of interaction between physical infrastructure and the energy system are incompatible between ICEVs and Battery Electric Vehicles (BEVs). ICEVs receive energy from a fuel distribution network which connects extraction operations to processing plants and ultimately to distribution stations. The fueling network is inherently inefficient due to constraints on the location of extraction sites, the economies of scale and access related to processing sites, and the requirement to expend fuel to transport fuel to end user locations which are widely distributed. It is, nevertheless, the case that this network has been subject to tremendous amounts of capital expenditure and operational optimization such that it's function and economic viability is ensured as long as extricable fuel supplies continue to be discovered and fuel-burning vehicles continue to be used.

PEVs, by contrast, draw energy from the electricity distribution network which also powers static loads such as those from buildings. Also the recipient of substantial and continuous investment, the power grid has been, historically, optimized to meet demand from buildings via the utilization of dispatchable resources including fuel-burning power plants. As the number of PEVs continues to rise, the load required to power these vehicles will become a greater portion of the overall load required. The interaction between vehicles and the power grid is heavily bound by constraints at coincident nodes and within their valency. Limitations exist on power transfer rates between the charger and battery as well as between the charger and power generation as a function of the distribution grid.

The most salient difference between the ICEV fuel delivery network and the EVSE network is at the interaction points. There are many more fueling stations in any given region of a developed country than public chargers. On the other hand, there are many more private charging locations than private fueling locations. PEVs are able to charge at low rates from many nodes including homes and buildings which are connected to the grid. Low rate charging is preferable for several reasons including lower costs per kWh and lower battery degradation. High rate charging is generally reserved for situations wherein charge time is relevant such as on itineraries whose distance is in excess of a vehicle's single charge range or for daily use by people unable to charge at home/work. The multifaceted nature of PEV charging contrasts to that of ICEV fueling which takes lace at public access stations and at similar rates for the vast majority of fueling events. A significant portion of PEV users can accomplish most charging without relying on public-access stations and these are encourages to maintain this behavior. The multi-tiered "charging pyramid" dictates that fewer public-access fast charging stations can be supported for a given number of PEVs compared to fueling stations and ICEVs.

As a result of the economic realities of charging stations, increases in network cardinality have, generally, been driven by incentives. The goal of these incentives is to create a network of public-access charging stations which is sufficiently ubiquitous and reliable that potential buyers will view PEVs as completely viable vehicles. A secondary goal would be to incentivise an efficient distribution of these stations such that profitable operations can be maintained after the subsidy period runs out. This document outlines methods of analysis for these networks which are fundamental and extensible and provide insights which can be used for further roll-out.

EVSE Network

For EVSE networks, the nodes are charging stations. The focus of this document is long distance travel and , thus, only DC Fast Charging (DCFC) nodes will be considered. Because interoperability between vehicles and chargers is not universal, the network will look different for different types of vehicle. An EVSE network's links are ephemeral end vehicle-dependent. Links exist between all chargers and all other chargers within range. This creates a highly connected network with significant locality. A key feature of any EVSE network is the low probability of availability at all nodes. Chargers are unreliable and may be unavailable for reasons including hardware and software faults, physical blockage, plug damage, payment system failures, and occupation. Because PEV fast charging events take a relatively long amount of time compared to ICEV fueling events and the relatively low number of plugs per station and stations per mile of road network, the probability of plug occupation is high, especially for central nodes. This can lead to a frustrating driving experience as limited prior knowledge of charger availability is the norm.

Constructing an EVSE network requires the locations of usable chargers, vehicle usable range, and a road network which has nodes coincident with the chargers, or nearly so. In this document, the road network will be referred to as the atlas. The links of the EVSE network are constructed by finding the lowest cost (however defined) route for all node pairs which do not violate cost limits. Routing along the network involves the additional step of adding origins and destinations to the network in the same manner as adding chargers.

Generic Example

Shown in Figure 1 is a randomly generated graph ($G = \{V, E\}$) with a cardinality of $N = 100$ on a 100 km by 100 km area with a link probability determined by $P(E|L_E) = e^{-L/S}$ where E is the link from V_i to V_j, L_E is the Pythagorean distance between V_i and V_j, and S is a characteristic distance, (10 km in this case). Among the nodes in the graph, one has been designated the origin, five have been assigned as destinations, and ten have been assigned as chargers.
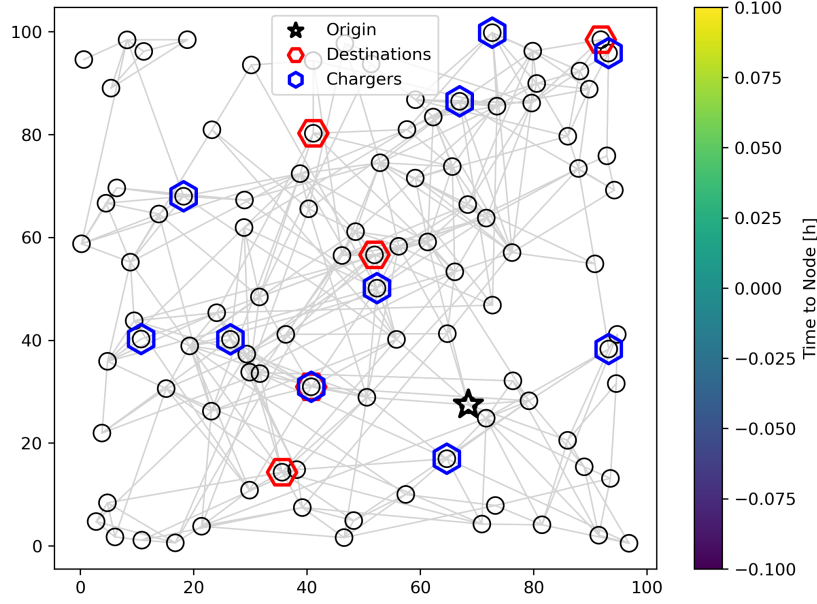

Figure 1: Randomly generated graph with cardinality of 100 with 1 origin, 5 destinations, and 10 chargers.

The links are traversed at speeds randomly selected from $\{35, 55, 90, 105\}$ kph. Like a real road network, all nodes are connected to the Giant Connected Component (GCC) but not directly to all others. Because the random graph is based on nodes drawn from a uniform random distribution, the distances between nodes should be normally distributed. Because the probability of a given link is a function of distance, one would also expect the nodes in the middle to have the highest valency and for valency to be Poisson distributed and this is, indeed, the case. The random graph shown in Figure 1 will be referenced throughout this document.

California Example

The state of California contains 1,618 towns and cities, collectively referred to as places, per the US Census Bureau and 1,906 public DC charging stations per Alternative Fuels Data Center (AFDC). Itineraries entirely contained in California will originate and terminate at California places. If additional range is needed beyond the remaining range at the start of the trip, Electric Vehicles (EVs) will utilize one of the DC charging stations. The places and stations are completely connected by roads. Thus, an entity atlas can be computed by routing from each entity to each entity. For the purposes of reducing computation time and memory requirements, a limit on link

ranges can be implemented which may result in a non-completely-connected entity atlas which should, nevertheless, be entirely contained within the GCC. The components of the California entity atlas are shown in Figure 2.
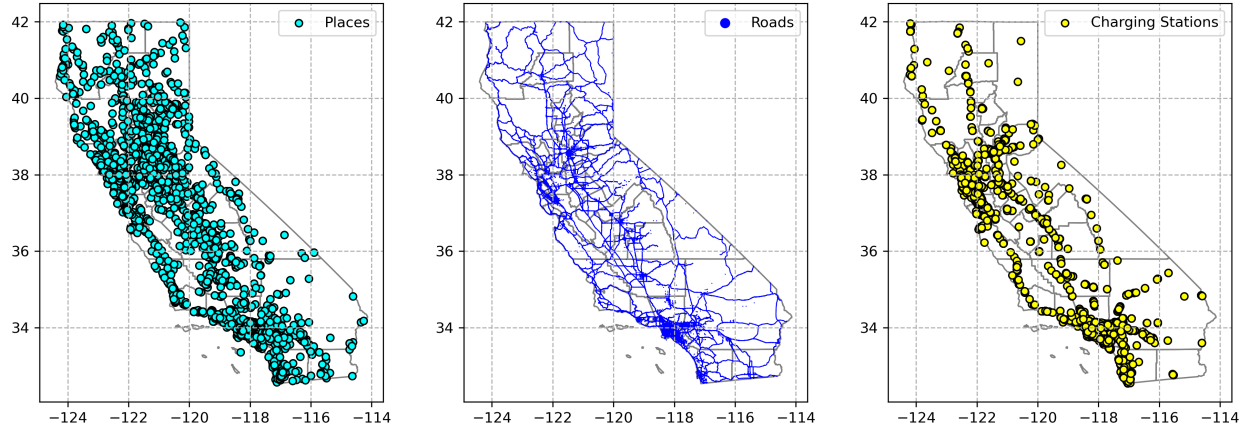

Figure 2: Components of California entity atlas

Routing

Background

For a graph $G = \{V, E\}$ where $O \in V$ is an origin and $D \in V$ is a destination, there is an optimal route $R^* = \{O, \Phi_0, \ldots, \Phi_N, D\} | R^* \subset V$ such that $J^* = F(G, R^*)$ is the global minimum value of J for routing cost function $J = F(G, R)$. There are, in essence, two approaches to finding R^*, Dijkstra's method and Bellman's method. Both methods are founded on the Hamilton-Jacobi-Bellman (HJB) equation

$$J_k = F(U_k, \ldots) + F(U_{k+1}^*, \ldots) \tag{1}$$

which states that the cost of a given control at the current step is the sum of the cost of the control and the cost of the optimal control at the next step subject to the control for a time-varying Optimal Control Problem (OCP). A corollary is that the optimal control at a given step is influenced by the optimal controls at all following steps. Dijkstra's method implements this concept via forward induction where Bellman's method implements this concept via backward induction followed by forward evaluation. The methods are diagrammed in Figure 3 and algorithms for both can be found in the Appendix.
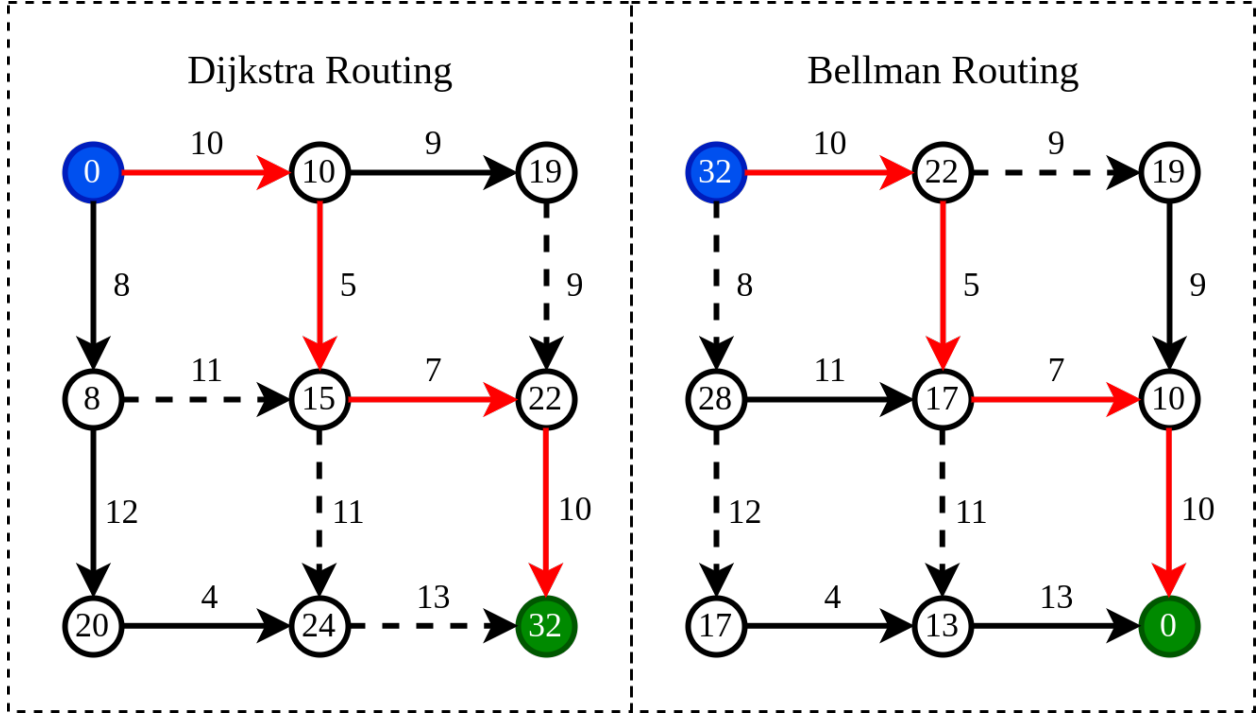
Figure 3: Comparison between Dijkstra and Bellman routing methods

It is worth noting that **Dijkstra's algorithm creates a tree from a single origin to all destinations** where **Bellman's algorithm creates a tree from all origins to a single destination**. The optimal paths created by both algorithms for a single O/D pair on a common graph will usually be the same unless there are many similarly expensive paths. The path produced by Bellman's method will be the globally optimal path where Dijkstra's method may produce a locally optimal path which is not globally optimal. The difference is that Dijkstra's method utilizes forward-integration where Bellman's method utilizes backward-integration followed by forward-evaluation. The separate integration and evaluation steps guarantee global optimality form any origin node. The integration stages in both methods are similar. For a un-directed graph, the difference between forward and backward integration is which nodes are the origins and which nodes are the destinations.

Route Constraints

In the basic formulation presented in the Appendix, minimum cost routes will be found for all destinations. However, in reality, routes can be infeasible for many reasons. Herein two reasons will be focused on these being excessive route cost and insufficient route probability. The cost of a route (R) is

$$J_R = \sum_{v \in V_R} \sum_{f \in F_V} f(v) + \sum_{e \in E_R} \sum_{f \in F_E} f(e) \tag{2}$$

where V_R is the set of nodes in R, E_R is the set of links connecting V_R in sequence, F_V is the set of cost functions for nodes and F_E is the set of cost functions for links. The probability of route R is

5

$$P(R) = \prod_{v \in V_R} P(v) * \prod_{e \in E_R} P(e) \tag{3}$$

In general, costs can be either positive or negative and limits on cost can have upper and lower limits. Probabilities for nodes and links will never be less than 0 or more than 1 meaning that increasing route cardinality can only lead to sustained or lowered route probability.

Stochastic Routing

EV Routing

Routing-Based Metrology

ICEV/EV Cost Differential Tree

Weighted Charger Centrality

Charger Availability Value-of-Information

Routing Algorithms

A generic formulation of Dijkstra's algorithm is as follows:

Algorithm 1 Dijkstra Routing Algorithm

$G = \{V, E\}$ ▷ Graph consisting of nodes and links
$C = \{c_e, \quad \forall e \in E\}$ ▷ Traversal cost corresponding to each link
$O \in V$ ▷ Single origin node
$D \in V$ ▷ Set of destination nodes

$S = \{\}$ ▷ Set of seen nodes
$W = \{\infty, \quad \forall v \in V\}$ ▷ All node weights initialized to infinity
$P = \{\{\}, \quad \forall d \in D\}$ ▷ Initializing empty path for each destination
$v = o_0$ ▷ Start by processing the origin node
while $D \not\subset S$ **do**   ▷ Iterating while not all destinations have been seen
    for all $e \in E$ **do**   ▷ Iterating through links
        if $e_0 = v$ **then**   ▷ If a link originates at the current vertex
            if $W_v + C_e < W_{e_1}$ **then**   ▷ Store current path and path cost if it represents savings
                $P_{e_1} = P_v \cup \{e_1\}$
                $W_{e_1} = W_v + C_e$
            end if
        end if
    end for
    $S = S \cup \{v\}$ ▷ Add current node to set of seen nodes
    $W_t = \infty$
    for all $u \in E \not\supset S$ **do**   ▷ Find the nearest unseen node for processing
        if $W_u \le W_c$ **then**
            $v = u$
        end if
    end for
end while

A generic formulation of Bellman's algorithm is as follows:

Algorithm 2 Bellman Routing Algorithm

$G = \{V, E\}$ ▷ Graph consisting of nodes and links
$C = \{c_e, \quad \forall e \in E\}$ ▷ Traversal cost corresponding to each link
$O \in V$ ▷ Set of origin nodes
$D \in V$ ▷ Single destination node

$S = \{\}$ ▷ Set of seen nodes
$W = \{\infty, \quad \forall v \in V\}$ ▷ All node weights initialized to infinity
$v = d_0$ ▷ Start by processing the destination node
while $D \not\subset S$ **do**   ▷ Iterating while not all origins have been seen
    for all $e \in E$ **do**   ▷ Iterating through links
        if $e_1 = v$ **then**   ▷ If a link terminates at the current vertex
            if $W_v + C_e < W_{e_0}$ **then**   ▷ Store current path if it represents savings
                $W_{e_0} = W_v + C_e$
            end if
        end if
    end for
    $S = S \cup \{v\}$ ▷ Add current node to set of seen nodes
    $W_t = \infty$
    for all $u \in E \not\subset S$ **do**   ▷ Find the nearest unseen node for processing
        if $W_u \leq W_c$ **then**
            $v = u$
        end if
    end for
end while
$P = \{\{\}, \quad \forall o \in O\}$ ▷ Initializing empty path for each origin
for all $o \in O$ **do**   ▷ Iterating on origins
    $v = o$
    while $v \neq d_0$ **do**   ▷ Iterating until destination is reached
        $W_t = \infty$
        $V_t = \{\}$
        for all $e \in E$ **do**   ▷ Finding the closest successor node
            if $e_0 = v$ **then**
                if $W_{e_0} \leq W_t$ **then**
                    $V_t = \{e_0\}$
                end if
            end if
        end for
        $P_v = P_v \cup V_t$ ▷ Adding closest successor node to path
    end while
end for