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.

Random Graph

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.

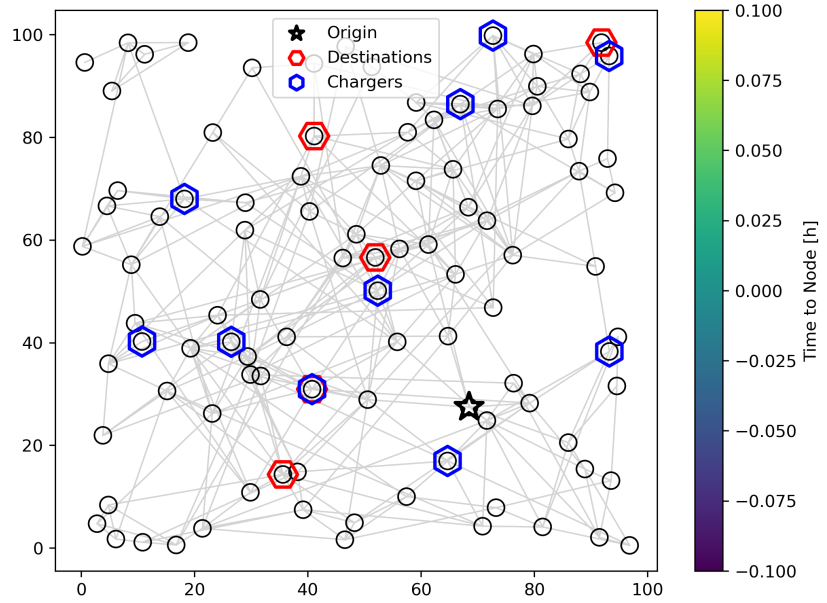


Figure 1: Randomly generated graph with cardinality of 100 with 1 origin, 5 destinations, and 10 chargers.

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.

California Graph

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.

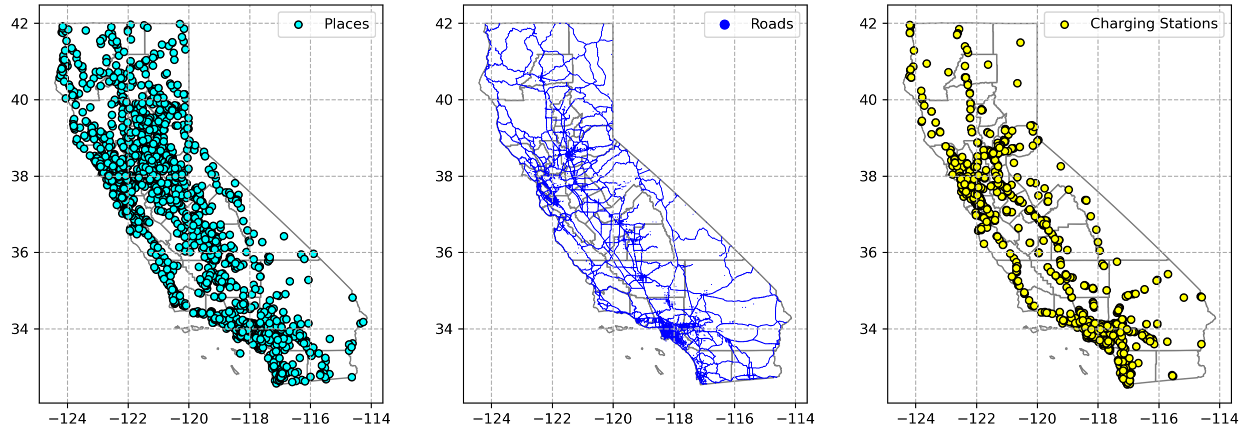


Figure 2: Components of California entity atlas

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 $R^* = \{O, \Phi_0, \dots, \Phi_N, D\} | R^* \subset V$ 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

$$J_k = F(U_k, \dots) + F(U^*_{k+1}, \dots) \tag{1}$$

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. Both Dijkstra's and Bellman's methods work by starting at a given point on a graph with a cost of zero and assigning infinite cost to all other points. The edges of the graph are then explored and for each edge (u,v) if the cost $c(v)$ is greater than the path cost $c(u) + c(u,v)$ then the cost at v is updated to the path cost. This process is called relaxation. The fundamental difference between the methods is that Bellman's method calls for evaluating every edge to check if a relaxation is possible for $|V|$ iterations where $|V|$ is the cardinality of the set of vertices belonging to the graph where Dijkstra's method calls for evaluating only those edges belonging to the "closest" unseen node at each iteration. Dijkstra's method uses a heap queue (priority queue) to keep track of nodes where every node that is reached is added and, at the beginning of each iteration, the edges of node with the smallest cost are evaluated. Essentially, the benefit of Bellman's method over Dijkstra's is that it allows for "upstream" changes to be evaluated constantly where Dijkstra's method only focuses downstream. However Dijkstra's is more often used because it is far more efficient than Bellman's method for the same problem. Generic algorithms can be found in the Appendix. Both methods will usually produce the same globally optimal route. The cases were Bellman's method is needed generally have to do with negative edge costs which are not common in transportation and many other fields. Both methods may be deployed with forward integration (sources to targets) or reverse integration (targets to sources) as diagrammed in Figure 3. It is worth noting that **forward integration creates a tree from a single origin to all destinations** where **reverse integration creates a tree from all origins to a single destination**.

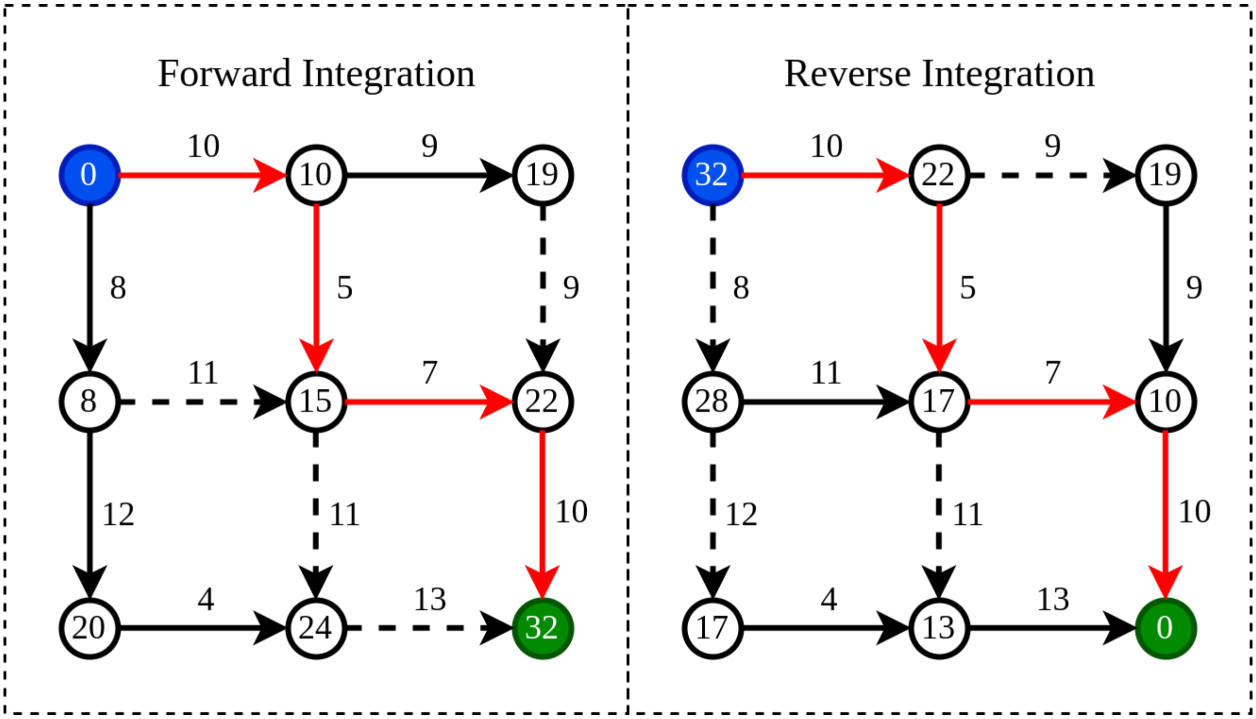


Figure 3: Comparison between Forward and Reverse Integration for Dijkstra's

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

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

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.

For vehicles, the primary routing constraint will be range. Each vehicle has a limited amount of range and must utilize energizing stations to add range in order to complete certain trips. Graph nodes may be designated as energizing stations. In routing, when an energizing node is reached the vehicle's range is restored and time is added to the route. In this manner, more trips are possible as seen in Figure 4.

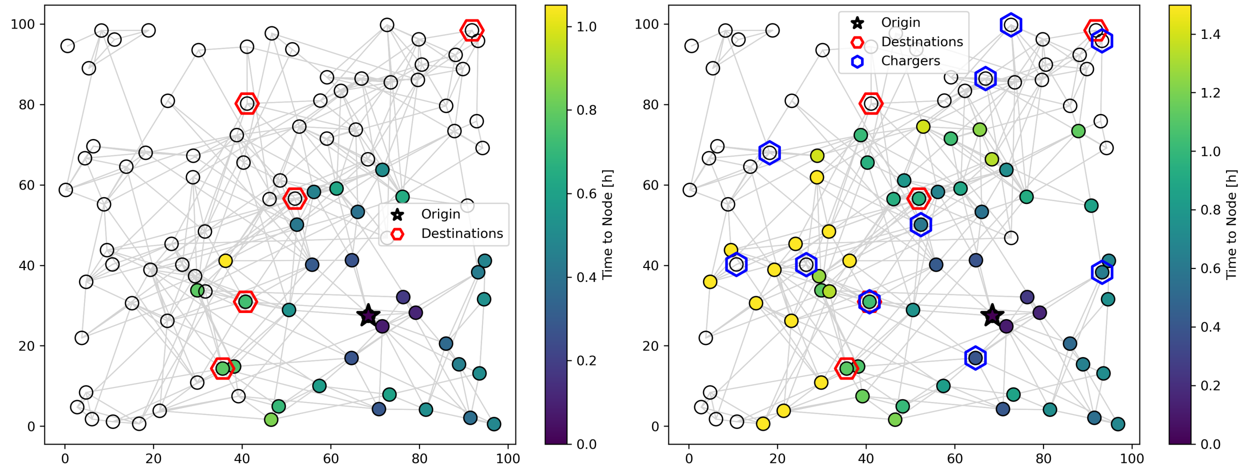


Figure 4: Routing with (right) and without (left) chargers

Stochastic Routing Methodology

The principle of stochastic optimization is to find the set of controls which returns the best expected outcome for an uncertain situation modeled as a set of discrete situations of known likelihood. The goal of stochastic optimization is

$$\min_{\overline{U}} \quad J(U) = \sum_{S \in \overline{S}} P(S) F(U,S) \tag{4}$$

where $\overline{S}$ is the set of possible scenarios, $P(S)$ is the probability of S, $F(U,S)$ is the cost of control vector U for scenario S, and $\overline{U}$ is the set of possible controls. A control vector $U = \{U_g, U_p\}$ contains general and specific controls (sometimes called first and second stage controls). General controls are shared among all scenarios and specific controls apply only ot one scenario. An example of stochastic optimization is provided in the Appendix.

For a BEV driving between an origin and a destination the controls relate to where to charge. If there is less than 100% certainty of a charger being usable (functional, accessible, and available) at the time of arrival to the charger, the optimization becomes stochastic. The technical non-functionality of a given charger may be known to the driver before the trip but accessibility and availability are not likely to be known prior to arrival. When a driver arrives at a charger and becomes aware that the charger is not usable the driver has the following options: (1) Wait at the charger until it becomes available, (2) re-route from current location to destination. If the charger is found to be non-functional or inaccessible then only option (2) is possible. Less than perfect reliability may result in an optimal route which is, otherwise, higher cost than that for perfect reliability. Less than perfect certainty may result in a route which is, otherwise, higher cost than that produced with perfect information. Sufficiently poor reliability/certainty may render a trip infeasible from the start or after several failed charge events.

Algorithms

Where deterministic routing occurs on a graph G, stochastic routing occurs on the spanning set of possible graphs $\overline{G}$ which has infinite cardinality in which all graph are isomorphic to all others ($G_i \cong G_j \ \forall\ i, j \in \hat{G}$). A corollary is that producing a route for $\overline{G}$ requires that there is some flat bijection $f : V_G \rightarrow V_H$ such that $(u,v) \in E_G \iff (f(u), f(v)) \in E_H \ \forall u,v \in V_G$ common to all graphs in $\overline{G}$ which contains the primary structure of the graph (which nodes and links are present). The flat isomorphic graph is $G' = \{V', E'\}$. Variability in route cost is caused by the differences in non-primary node and edge properties between graphs. An example would be a road network where the streets are always present but have variable speeds, some of which may be zero, at a given time.

In reality, optimization must occur over a finite set. Thus stochastic optimization occurs on $\hat{G} \subset \overline{G}$ of cardinality N. Higher values of N better approximate $\overline{G}$ at the cost of increased computational load. Dijkstra's method for stochastic routing results in a single general optimal route R^*. R^* is found using a Monte-Carlo style solution wherein N integrations are simultaneously computed along the same paths experiencing different parameters along the way. The goal of the optimization is to produce the route with the lowest expected cost $\mathbb{J}^* = F(R^*, \hat{G})$ where R^* is the expected optimal route.

R^* is subject to charger usability for O/D pairs of a greater distance than the vehicle's remaining range. The net result of charging (or fueling etc.) is that the vehicle's range is reset, time is added to the route, and money is added to the route price. Time additions are due to charge time and a possible delay due to queuing. Price additions are due to charge pricing. As such, the chargers in $\hat{G}$ (present at a subset of nodes V_C') are defined by function Θ whose parameters are $\hat{\Psi} = \{\Psi_G, \quad \forall G \in \hat{G}\}$ where $\Psi_G = \{\psi_v^r, \psi_v^p, \psi_v^d, \psi_v^c \ \forall v \in V_C'\}$. The parameters of Θ are the range completion of the charge event (the reset range), the charge power, the wait time before charging, and the charging cost respectively. If the vehicle's remaining range is greater than ψ_v^r no charge event will occur. For stochastic optimization, all parameters are sampled from defined distributions each time the charger is visited.

Depending on the configuration of G, certain routes may be feasible or infeasible (because they cause the vehicle range to go below a minimum value) potentially creating a discontinuous cost space where the costs of the feasible routes must be considered alongside those of infeasible routes. A probabilistic risk measure, the superquantile provides an elegant solution. The superquantile function for a given distribution is

$$S_p(D) = \frac{1}{1-p} \int_p^1 Q_\alpha(D) d\alpha \tag{5}$$

where D is a given Probability Distribution Function (PDF), $p \in [0,1]$ is a probability threshold, and Q_α is the quantile function of a given PDF for probability α. In effect, the superquantile is the mean of the PDF after p. For route R on graphs $\hat{G}$ of cardinality N, there will be a revealed distribution D_r of remaining range at all points. The remaining range constraint can thus be written as

$$S_p(-D_r) \geq -r^{min} \tag{6}$$

where r^{min} is the lowest allowable remaining range and p is set by the user. In effect, (6) guarantees that the expected value of the worst $1-p$ portion of outcomes is within the feasible range. General values of other route costs are computed similarly. The routing framework described above will be referred to as Stochastic Cost with Risk Allowance Minimization (SCRAM).

SCRAM-Dijkstra (SCRAM-D)

The SCRAM-D algorithm is outlined below.

Algorithm 1 SCRAM-D Routing Algorithm

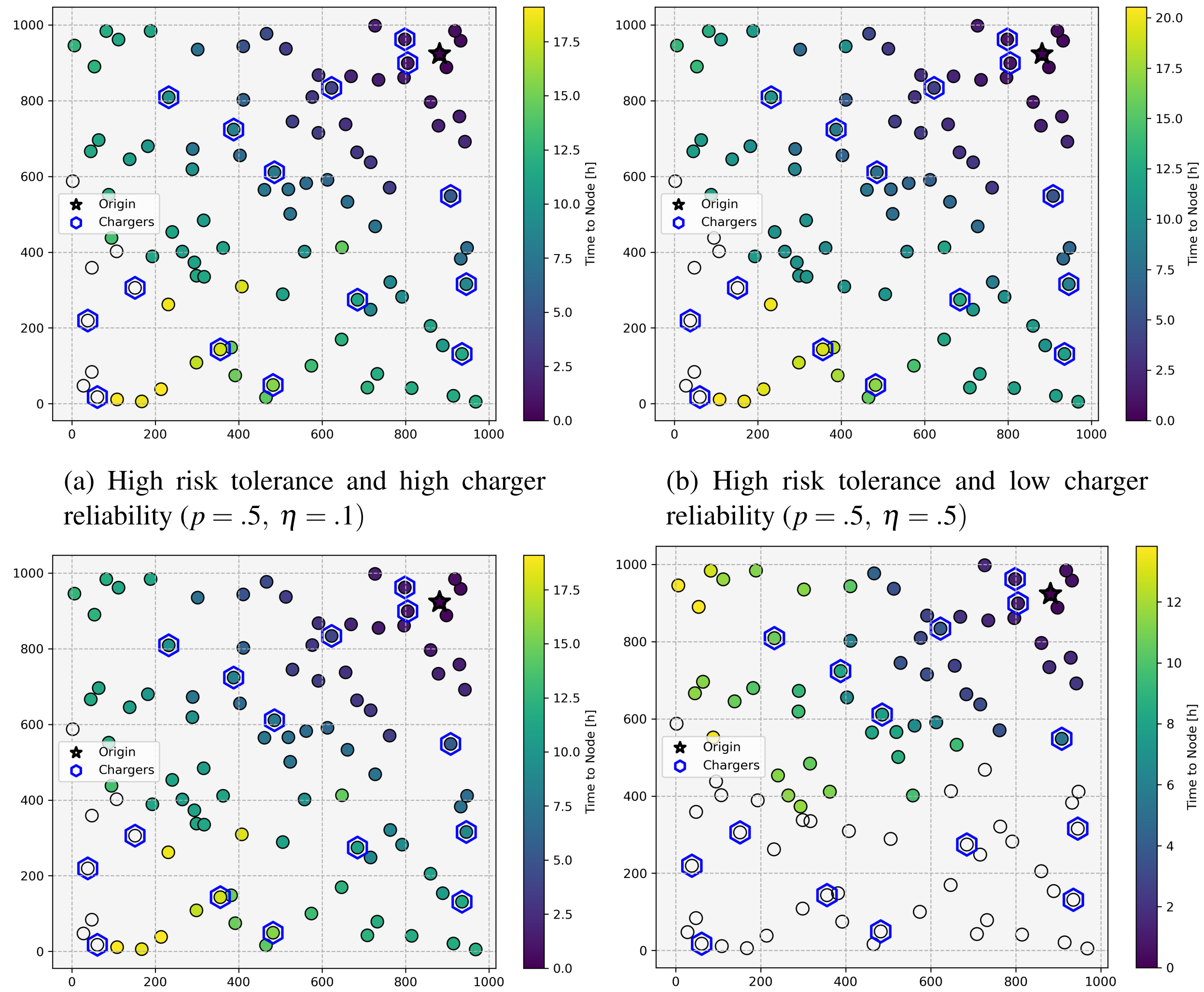
```
$G' = f(G \in \hat{G}) = \{V', E'\}$    ▷ Flat isomorph common to all graphs in $\hat{G}$
$\hat{C} = \{\hat{C_V}, \hat{C_E}\}$    ▷ Costs for graph elements for each graph in $\hat{G}$
$\hat{\Psi} = \{\Psi_v, \quad \forall v \in V'_C \subset V'\}$    ▷ Charger parameters for the subset of nodes with chargers in $G'$
$O \in V'$    ▷ Single origin node
$D \in V'$    ▷ Set of destination nodes

$S = \{\}$    ▷ Set of visited nodes
$W = \{\{\infty\}_{G \in \hat{G}}\}_{v \in V'}$    ▷ All node weights initialized to infinity for all graphs
$R = \{\{\}\}_{d \in D}$    ▷ Initializing empty path for each destination
$F = \{(w_{o_0}, o_0)\}$
  ▷ Heap queue containing nodes to be visited. Elements are tuples (weight, node) and the heap is ordered by weight. The heap is initialized with the origin node.

while $F \neq \emptyset$ do    ▷ Iterate while there are reachable nodes which have not been visited
    $w_v, v = f_0$
    $F = F \not\cup f_0$    ▷ Remove current node from heap queue
    if $v \in V'_C$ then    ▷ If current node is a charger ...
        $l_G, w_{v,G} = \theta_G(l_G, w_{v,G}, \psi_G)$    ▷ Update remaining range and cost after charging
    end if
    $E'_v = \{(v_s, v_t) \in E' | v_s = v\}$
    for all $(v, v_t) \in E'_v$ do    ▷ Iterate through links of $v$
        $\mathbb{E}[w_{v_t}] = \mathbb{E}[\{w_v + c_{v,(v,v_t)} + c_{v_t}\}_{G \in \hat{G}}]$    ▷ Compute an expectation for cost of current path
        if $\mathbb{E}[w_{v_t}] < w_{v_t}$ then    ▷ If current path represents savings ...
            $r_{v_t} = r_v \cup \{v_t\}$    ▷ Update path to $v_t$
            $w_{v_t} = \mathbb{E}[w_{v_t}]$    ▷ Update cost at $v_t$
            if $v_t \notin S$ then
                $F = F \cup \{(w_t, v_t)\}$    ▷ If $v_t$ has not been visited add to heap queue
            end if
        end if
    end for
    $S = S \cup \{v\}$    ▷ Add current node to set of visited nodes
end while
```

Routing-Based Metrology

SCRAM is valid for any N. The added value of SCRAM is that is that it allows for risk-tolerance-sensitive routing. This is of particular value for PEVs because charger usability is not guaranteed. When a PEV driver arrives at a charger it may be non-functional or inaccessible in which case it cannot be used or unavailable in which case the driver must wait to use it. Non-functionality and inaccessibility are complete failures where unavailability is a partial failure. Complete failures are modeled using ψ_v^r. If the vehicle's remaining range is greater than ψ_v^r then the vehicle does not charge, if $\psi_v^r = 0$ then no vehicle can charge which is equivalent to a complete failure. Thus

ψ_v^r can be sampled from $\{0, r\}$ with defined probabilities for each. Partial failures are modeled with the ψ_v^d term which is sampled from $[0, \infty)$ and added to the time required for route R which includes v. Risk tolerance is modeled using the superquantile function parameter $p \in [0, 1]$ where higher values indicate a tighter focus on worse outcomes. **Thus R^* is a function of both graph parameter distributions and driver risk tolerance**.

Consider a randomly generated graph of 100 nodes on a 1,000 by 1,000 km grid where chargers are located at 15 nodes. Link probabilities are proportional to distance by the formula $P((v, u)) = \exp(\frac{-\|(v,u)\|}{s})$ where the characteristic distance s is 200 km. A driver is currently at a single origin node and wants to find R^* for every other node. The driver's car has a range of 400 km. The driver has risk tolerance p. Charger parameters are randomly sample as above where the probability of charger complete failure is η. Risk tolerance applies both the the probability of route completion and the metric of optimization which is travel time in this case. A factorial of cases for $p \in \{.5, .9\}$ and $\eta \in \{.1, .5\}$ are shown in Figure 5.



(a) High risk tolerance and high charger reliability ($p = .5,\ \eta = .1$)

(b) High risk tolerance and low charger reliability ($p = .5,\ \eta = .5$)

(c) Low risk tolerance and high charger reliability ($p = .9,\ \eta = .1$)

(d) Low risk tolerance and low charger reliability ($p = .9,\ \eta = .5$)

Figure 5: Expected travel times for optimal routes generated by SCRAM-D for varying charger reliability and driver risk tolerance

As seen in Figure 5 SCRAM-D optimal routes are effected by both reliability and risk tolerance in an additive manner. The high risk tolerance scenarios in the top row show a small effect of charger reliability which leads to slightly longer travel times. The high reliability scenarios in the first column shows a small effect of risk tolerance, even smaller than the previously mentioned effect. However, the low risk tolerance scenarios in the second row show a huge effect of charger reliability and the low reliability scenarios in the second column show a huge effect of risk tolerance. This result reinforces the need for SCRAM - **the underlying parameters of PEV charging networks go hand-in-hand with user risk tolerance to effect PEV dis-utility**.

More to come ...

Appendix: Stochastic Optimization Example - Crop Allocation

Problem Description:

A farmer has 100 total acres of fields to plant and may choose from among three crops. Each crop has a different cost per acre for seeds. Depending on meteorological conditions, each crop will have a different revenue per acre in low, medium, and high yield scenarios. Seeds must be purchased in the fall but the farmer has until the spring to actually decide what to plant and at that point he will have more information about the weather for the coming year. Any excess seeds may be re-sold for a 20% loss. What quantities of seed should the farmer purchase in order to maximize profit?

Discussion:

In the example the farmer has to make one set of decisions (what quantities of seeds to buy for the three crops) which holds for all yield scenarios. These are general controls. The farmer then has a set of decisions (what quantities of seed to plant and to resell for each crop) which may vary in different yield scenarios. These are specific controls. The problem contains 3 general controls and 9 specific controls (3 per scenario) for a total of 12 controls. The expected profit for the farmer at the time of seed purchase is the total cost from (4) over the number of scenarios.

If the farmer had complete information about the following year's weather at the time of seed purchase he could optimize both seed purchase and seed planting for specific scenarios. In this case the farmer could compute optimal controls for each scenario individually (6 specific controls per scenario). The difference between the expected profit from the stochastic optimization and the mean value of the profits from the deterministic optimizations is called the "cost-of-uncertainty" or the "value-of-information".

A few things are worth mentioning specifically. first, the cost-of-uncertainty should be positive (expected profit less than mean profit). In the example, the farmer may have to buy more seeds than he can possibly plant due to uncertainty and have to sell the excess at a loss. In the deterministic scenarios the optimal solution will never call for purchasing more seeds than needed for planting if the resell value is less than the purchase value. It is intuitive to see that the farmer is paying a price for his decision to purchase seeds before it is possible to predict the weather with sufficient accuracy. Second, stochastic optimization reduces to deterministic optimization with the removal of general controls. In other words, a stochastic optimization problem is an optimization problem with at least one general control and need not have any specific controls.

Appendix: Generic Routing Algorithm Formulations

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

Algorithm 2 Dijkstra Routing Algorithm

```
$G = \{V, E\}$                                  ▷ Graph consisting of nodes and links
$C = \{C_V, C_E\}$                              ▷ Traversal costs corresponding to each node and link
$O \in V$                                       ▷ Single origin node
$D \in V$                                       ▷ Set of destination nodes

$S = \{\}$                                      ▷ Set of visited nodes
$W = \{\infty\}_{v \in V}$                      ▷ All node weights initialized to infinity
$P = \{\{\}\}_{d \in D}$                        ▷ Initializing empty path for each destination
$F = \{(w_{o_0}, o_0)\}$
  ▷ Heap queue containing nodes to be visited. Elements are tuples (weight, node) and the heap
is ordered by weight. The heap is initialized with the origin node.

while $F \neq \emptyset$ do                     ▷ Iterate while there are reachable nodes which have not been visited
    $w_v, v = f_0$
    $F = F \not\cup f_0$                        ▷ Remove current node from heap queue
    $E_v = \{(v_s, v_t) \in E | v_s = v\}$
    for all $(v, v_t) \in E_v$ do               ▷ Iterate through links of $v$
        if $w_v + c_{v,(v,v_t)} < w_{v_t}$ then ▷ If current path represents savings ...
            $p_{v_t} = p_v \cup \{v_t\}$        ▷ Update path to $v_t$
            $w_{v_t} = w_v + c_{v,(v,v_t)}$     ▷ Update cost at $v_t$
            if $v_t \notin S$ then
                $F = F \cup \{(w_t, v_t)\}$     ▷ If $v_t$ has not been visited add to heap queue
            end if
        end if
    end for
    $S = S \cup \{v\}$                          ▷ Add current node to set of visited nodes
end while
```