Team Control Number

For office use only
T1________________
T2________________
T3________________
T4________________

74251

Problem Chosen

D

For office use only
F1________________
F2________________
F3________________
F4________________

2018
MCM/ICM
Summary Sheet

Entering the Era of Electric Vehicle

The rapid development of all-electric vehicles indicates that our world will one day migrate to an all-electric vehicle era. In this paper, we mainly focus on the migration process, and establish different models to tackle different problems that human may face in the transition period.

We start our investigation on current Tesla's charging network, and incorporate the queuing model to evaluate the connectivity as well as the capacity of the network. Preliminary research suggest that Tesla's charging network can well satisfy current charging demand, but will literally malfunction if all cars in the US switch to all-electric vehicles. From our model, we also calculate the number of stations needed for a 100 % migration, and further analyze the distribution of the charging stations.

Next, we study on the switch process of Ireland. To start with, we create the Queuing-optimal model, which includes methods of queuing model and integer programming. Further, we compute the optimal number of charging stations for a instantaneous switch in each city of Ireland, which suggests building 872 stations in Ireland. In addition, we establish the hierarchy population model to describe the population distribution in Dublin, and further formulate a concise construction methodcalness for Dublin using the Chess-playing algorithm we disigned. Inspired by the subtle bio-chemical process, we formulate the Enzyme model to predict the timeline of the transformation, and the result suggests it takes about at least 100 years from now for a 90 % migration.

In order to provide insights for selecting different investment policies, we found the Dynamic-path-simulating model based on state transition equation. We compare the different results of a variety of policies, and analyze the proper investment plan for governments to implemment. Additionally, we design a classification system helps a nation determine the general plan that they should follow. The results suggest a country should focus more on charging network construction if it has relatively low income level and under-developed technology level, while wealthy and high-tech countries should spend more investment in encouraging citizens to switch their vehicles to electric ones.

Finally, we briefly comment on the possible influence of novel technologies, and analyze how they will impact our alalyses of the migration process.

Entering the Era of Electric Vehicle

Contents

1 Introduction

It is reasonable to assumed, that the conventional non-electric vehicles will oneday be replaced by all-electric vehicles. Though the influence of electric cars on energy consumption is still heatedly debated[1], their increasingly easy accessibility[2] as well as the governers' support[3] indicate that the migration has already started.

Considering the factors mentioned above, the arrival of electric vehicles era is only a matter of time. Though it would be a fascinating world with all vehicles using cleaner energy, obstacles and unsolved problems still exist. Is the current electric energy supply network sufficient for a dramatic upsurge of electric vehicles ownership? If not, how should local authorities propose plans for the growing and evolving charging network?

With strong relevance to our daily life, these problems facing mankind are drawing great attention of the academic society. In 2016, Zhao et. al. proposed a network model for optimal charging station sitting and sizing[4]. In the same year, a model designed by Shao et. al. incorporate a queuing method to help optimizing location selection for charging stations[5]. Though these have provided powerful methods to tackle problems we may face, challenges remains. The fist model suggests building large charing stations with a service area of up to 128.5 square kilometers, while in the second model, geological parameters are simplified as rectangles and nodes. Further more, these models only focus on optimalizing the location of charging stations while ignoring the gradual process of the whole migration.

In this paper, we mainly focus on the process of migration from non-electric vehicles to electric ones. Five different models including the Queuing-optimal model, the hierarchical popolation model, Chess-playing algorithm, the Enzyme model and the Dynamic-path-simulating model are established to analysise different aspects of the problem.

2 Assumptions and Justifications

- The parameters of electric vehicles discussed in this paper come from Tesla Model3. Considering that this protocol is affordable for more people and represents the current technology of electric vehicles manufacture, we think it appropricate to use it as a typical example;
- We only focus on continential United State in this paper. It is impossible for current vehicles to travel across oceans, and the term "US" used in this paper refer to "the continential United State";
- The vehicle in the passing lane that is behind our vehicle is sufficiently far behind that we will not interfere with it.

3 Evaluating the Tesla Charging System

We first collect raw data describing the city, state and other location information from Tesla's offical website using web crawler techniques. Then the raw data was transfer to accurate coordinates of the charging stations with the help of Google API.

3.1 Connectivity of the Charging Network

To investigate whether Tesla on track allows a complete switch to all-electrics in US, the criterion used to evaluate whether the energy supply is sufficient is established. By "sufficient", we mean a electric vehicle driver can travel all over US with the help of the charging network.

To start with, we consider the extreme situation of only one electric vehicle in the whole nation. Supposed the driver start his journey with fully charged batteries and fully charge his car when necessary. Since the maximun mileage of the vehicle is approximately 310 $miles$[6], a supercharge station or a destination charge station will be needed within the mileage, or the drivers cannot drive further to other destinations. Under this notion, we exam the connectivity of the charging station using the method discribed in Figure 1.

$$min\{d(C_j, C_i)\} < r_{max}$$

Notations for Testifying Connectivity
1. C_j denotes the j^{th} location
2. C_i denotes the nearest charging spot of the j^{th} station
3. r_{max} denotes the maximum mileage

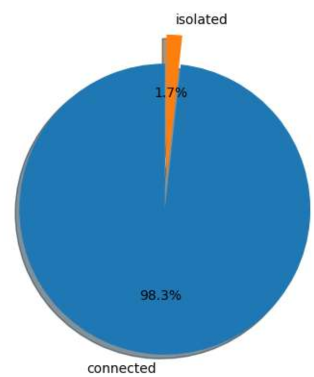


Figure 1: Connectivity of the Charging Network

The result clearly points out that distances between most of charging stations are less than the maximun mileage, meaning the network is well connected, while 1.7 % of the charging stations are disconnected to the network.

3.2 Capicity of the Charging Network

To further investigate the capacity of the Charging network, we consider other electric vehicles on the road, and see if a driver can still travel all over US conveniently. Different from the former case, the capacity of every charging station should be throughoutly analyzed. An overcrowded charging station will be no use for drivers in badly need for charging. In other words, when a driver need to wait for an unreasonablely long period of time, we can say that the charging station is literally malfunction.

3.2.1 Modeling the Population and Vehicle Distribution

Under above-stated considerations, a population and vehicle distribution model will be needed. To model the population distribution of US, we found the demograph stastics of 755 cities[7], as well as their coordinates and acreage. We assume the distribution obeys the

Gaussian distribution, thus we derivate the population distribution equation as shown in Figure 2. It should be noted that by calculation, the area of continential US is 70 times of the cities' area, and we expand the cities's area in our model.

$$\rho_{pop} = \frac{N}{(1-e^{-1}S_{city})}e^{-\frac{\pi d^2}{70S}}$$

Notifications of Gaussian Distribution
1. ρ_{pop} denotes density of the population at a specific point
2. S_{city} denotes the area of a city
3. d denotes the distance between a specific spot and the center of city
4. N denotes total population of city

Figure 2: Gaussian Distribution of population

We assume that the density of vehicles is propotional to the density of population, and people in each city share similar living strandards. Considering that there were about 1.8 vehicles available per U.S. household in 2016[8], every US citizen owns approximately 0.6 vehicles, and thus $\rho_{car} = 0.6\rho_{pop}$, which is our vehicle distribution model.

3.2.2 Capacity of Destination Charging stations

In terms of destination charging station, although more than 3000 of them are built in the US, their service capacity is relatively small due to its limited charging plugs and overlong charging time. Thus, their function is more like storing vehicles. Since people will book the station in advance, and park their vehicles there while charging for overnight. It is reasonable to assume that the destination charging stations are occupied most of the time.

We locate all the destination charging stations in US, and find the nearst city center of the station. Then, we subtract the numbers of Tesla connectors from our vehicles distribution model, to evaluate the influence as well as the capacity of the destination charging stations.

Table 1: Subtracting from Vehicles Model

Cities	Original	Substraction	Finals	Cities	Original	Substraction	Finals
1	347227.2	1	347226.2	5	381878.6	2	381876.6
2	152960.4	5	152955.4	6	96802.2	1	96801.2
3	360039.6	2	360037.6	7	347542.2	6	347536.2
4	381882.5	4	381878.5	8	152955.4	3	152952.4

*Original denotes the original number of vehicles in a city

The above figures are randomly selected from the 755 cities, and it shows that destination charging stations make little difference to the origin model, which approve our previous assumption, that these stations have relatively limited capacity.

3.2.3 Capacity of Supercharge Stations

In terms of supercharging station, which provide fast charging service for customers, we first quantify the service areas covered by the supercharge stations using the voronoi graph method. In this method, we determine the midpoint between each two supercharge stations, and generate a celluar pattern which well resolve the problem of service overlap. Since people tend to choose the nearest charging stations in real life, this method well defines the different service area of a station.

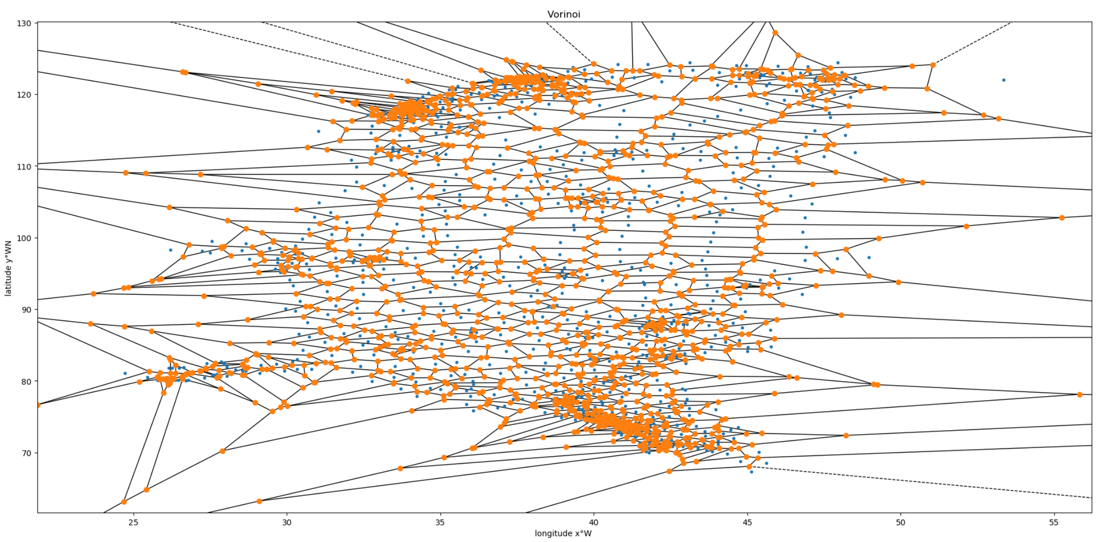


Figure 3: The Voronoi Graph and Service Area of Supercharge Stations

It should be noted that Tesla's website also offer information about supercharge stations which will be avilable at the end of 2018. We also include these stations in our model, and estimate they have 10 connectors each. Since our method does not take coast lines and the national boundaries into consideration, some service region become particular large as we can observe in Fig.3, and thus we abandon unreasonably big acreage, and replace it with a average acreage.

Then, a queuing model was incorporated in order to quantify capacity of the stations.

$$P_0 = [\sum_{k=0}^{c} -1 \frac{1}{k!}(\frac{\lambda}{\mu})^k + \frac{1}{c!}\frac{1}{1-\rho}(\frac{\lambda}{\mu})^c]$$

$$L_q = \frac{(c\rho)^c \rho}{c!(1-\rho)^2} P_0$$

$$W_q = \frac{L_q}{\lambda}$$

Notifications of Queuing Model[9](M/M/C)
1. P_0 denotes the probability of the situation when no one comes to charge 2. λ denotes the average arrival rate of vehicle visiting the station 3. μ denotes the average service rate of a single service 4. c denotes number of charging connectors 5. L_q denotes the length of the assumed waiting line. 6. W_q denotes the time drivers need to wait

We assumed that car owners will visit charging stations only once every 2 weeks, and thus $\lambda = \frac{1}{24\times7}\rho_{cars}S_{ser}$. To identify μ, it is reasonable to say drivers prefer to charge 30 min at one time since a 30-min-charge will provide considerable energy supply[10]. The Queuing model, which generates the waiting time for drivers is well established in this sense.

We further calculate the exact waiting time for each station, and the result indicate that drivers basically do not have to wait for charging with the current electric vehicles ratio of roughtly 0.3 %. However, when all vehicle migrate to electric ones, our model indicate that one will have to wait preposterously long to charge his vehicle. (See Table 3 & Table 4 for part of the results)

Table 2: Current Waiting Time

Stat.	Time(h)	Stat.	Time(h)
1	7.05×10^{-10}	5	1.03×10^{-8}
2	3.14×10^{-10}	6	5.40×10^{-14}
3	8.42×10^{-7}	7	2.94×10^{-5}
4	8.48×10^{-14}	8	3.27×10^{-18}

Table 3: All-electric Waiting Time

Stat.	Time(h)	Stat.	Time(h)
1	4325653.2	5	15803.0
2	1815.9	6	3622784373.2
3	2616801.6	7	199216123.1
4	145932921.4	8	2668789415.3

The results indicate that the supercharge station network is currently sufficent. On the other hand, our model also point out that present supercharge network cannot satified the demand for charging when 100% vehicles in US are all-electric vehicles.

3.3 Estimating Proper Charging Network for Complete Migration

According to our model, if everyone switch to all-electric vehicles in US, then more charging stations will be needed. Thus, we include add more charging stations in our model to shorten the waiting time of drivers, and see how many more charging stations will be needed in each service area.

We assume it is acceptable for drivers to wait 30 min in real life. Then, we multiply the number of charging stations in the origin service area. If the waiting time falls below 30 min, the algorithm stops adding charging stations in the area and hence we obtain the needed charging stations. Part of the result was shown in Table 5. By calculation, if everyone drive all-electric vehicles in US, 154,805 charging stations will be needed all over conteniential US.

Table 4: Charging stations needed to be built

Stat.	Coordinates	Needed stations	Stat.	Coordinates	Needed stations
1	(34.785,86.942)	153	5	(31.856,86.635)	187
2	(32.628 85.448)	91	6	(32.367,86.300)	121
3	(33.523 86.809)	137	7	(33.607,85.784)	73
4	(31.223 85.390)	111	8	(33.934,86.191)	97

3.4 Distribution Pattern of the Charging Network

With the new station construction plan generated from our model, we analyzed geological distribution of the charging stations.

We fistly define the cirterion for distinguishing among rural, urban and suburban areas. To simplify the model, we suppose all cities have a circle shape, which is close to most of the cities, and thus the radius of each area will be $R_{range} = \sqrt{\frac{70 \times S_{city}}{\pi}}$. Then, we compute the distances R_{char} of the area, and classify each charging stations based on our definitions of "urban","rural" and "suburban" areas, and finally calculate the distribution map of the charging network.

Region	cirteria
Urban	$R_{char} < r_{city}$
Suburban	$r_{city} \le R_{char} < R_{range}$
Rural	$R_{range} \le R_{char}$

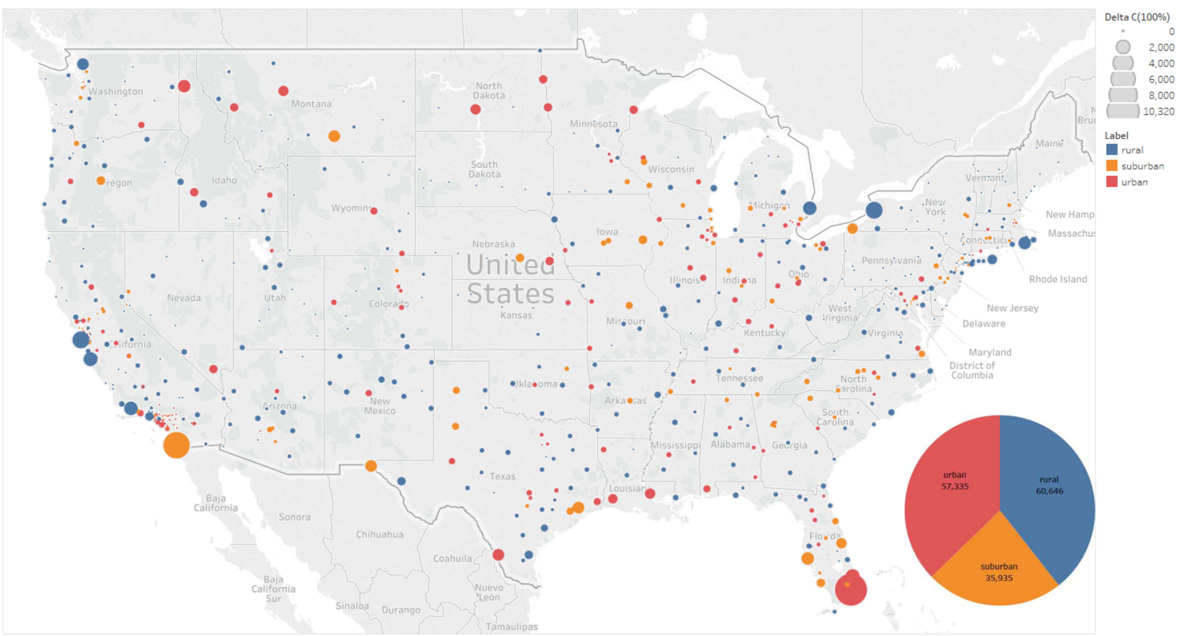


Figure 4: Distribution Pattern of the Charging Network

Under the our judging criterion, 57,335 charging stations will be distributed in the urban areas, whereas 60,646 and 35,935 stations will be in rural areas and suburban areas respectively. Also, more charging stations will be settled in California as one can observe in Figure 4. Though the definitions distinguishing these three ares are not very clear, we can roughly conclude that the distribution of charging station is not necessarily limited within relatively crowded urban areas, which is reasonable since there are probably more personal garages and driveway with power in urban areas.

4 The Migration of Ireland

After examing the status of the current Tesla charging network, we further investigate the whole migration dynamics from non-electric vehicles to all-electric vehicles. In this part, we choose Ireland as a example for our further study.

4.1 Modeling the Instantaneous Switch

We first consider a fantasy situation, that the migration requires no transition time, and then determine the optimal number, placement and charging stations in Ireland. It should be clarified, that although destination chargers play important roles in the real life, it have little influence on the capacity of charging network, and thus we ignore the effects of destination charging stations in our model.

4.1.1 Hierarchical Population Distribution Model of Ireland

With demographic data of Ireland[11], we firstly establish the population distribution model of Ireland. To estimate the total number of needed charging stations, a coarse population will be satisfacying. Thus, we use different population data of cities in Ireland to establish the first hierachy of our model.

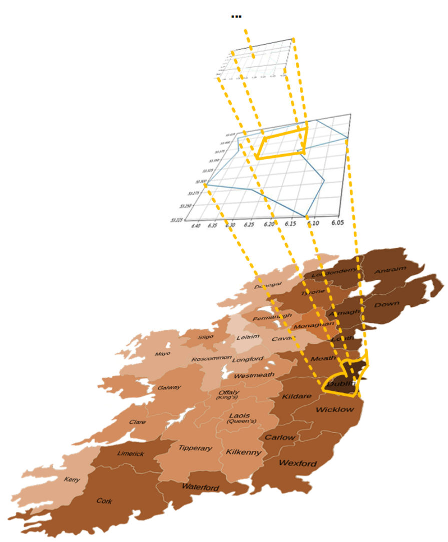


Figure 5: Hierarchical Population Model

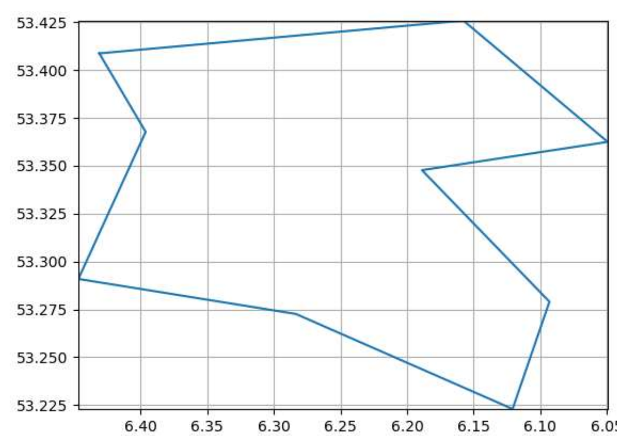


Figure 6: Griding Dublin

In addition, more detailed information such as population density will be needed in modeling optimal placement and distribution of charging stations, and thus we choose Dublin, the capical of Ireland to create the second hierarchy of our population model. To describe the population density of Dublin, we first set grids to divide the whole city into 64 smaller blocks, and choose the one that includes City Hall of Dublin as the city center block.

$$P_{pop} = 51760.88e^{-k_{dis}\frac{ln(2)}{4}}$$

Notifications of Discrete Distribution
1. P_{pop} denotes the population at a specific block 2. k_{dis} denotes distance from the centarl block

Then, we notice that satelite cities around Dublin such as Swords city only have half of the population of Dublin. Thus, we assume population density of the city center is twice as the density at the edge of the city, and formulate the discrete popultion model which determines the population of each small block.

It is worthy to note that one can model much more subtle population model using the hierarchical model, and in this paper we only create the second hierarchical model of Dublin. Similar approaches can be duplicated to generate even more detailed populations models of Ireland.

4.1.2 Determining the Network of Charging stations

With the established population model, we start modeling the number, placement and distribution of charging stations. Assuming the vehicle density is propotional to the population density, we also consider that 2,631,093 vehicles are registered in Ireland[12], and thus we can describe the vehicles distribution in Ireland.

The number of needed charging stations was analysised base on the hierarchical population distribution model. The queuing model was again incorporate in the process, and we assume each charging station has 6 connectors. Given the bondary condition, which is drivers shouldn't be waiting for more than 30 min, we compute the number of total charging stations in Ireland, as well as number of stations in every city. Considering one can only build integer number of stations, we also use interger programming to generalize the final results.

$$min \sum_{ij} (n_{ij} - a_{ij})^2$$

$$s.t. \quad n_{ij} \in N^*$$

Notifications of Integer Programming
1. n_{ij} denotes the final number of stations in a specific block 2. a_{ij} denotes non-integer optimal number of stations in a specific block

The total number of needed charging stations in Ireland is 872, while the capital city Dublin will need 247 stations. Part of the data describing the needed stations in other cities and the distribution network of charging station are illustrated in Figure 7.

City	Number of Stations
Carlow	11
Dublin	247
Kildare	41
Kilkenny	19
Laois	16
Longford	8
Louth	24
Meath	36
Offaly	15
Cork	100

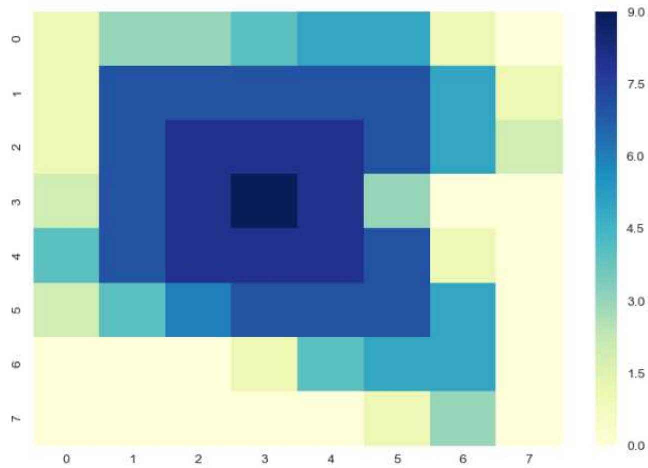


Figure 7: Stations in Ireland Cities and Stations Distribution in Dublin

4.2 Evolution of the Charging Network

Though we have already calculated the number, placement and distribution of charging stations, problems still exist in more realistic situations. Since a prompt switch would be impossible, one would have to determine a proposal for evolving the network.

4.2.1 Urban or Rural: Proposal on Constructing Methodicalness

To tackle the above-mentioned delimma, we first come up with a semi-greedy model, which firstly build one necessary charging station in each of the blocks to maintian connectivity, and then start building from the blocks with maximun population density. However, the method behind this algorithm is rather ambigulous. It cannot quantify the real value of connectivity and will inavoidably lead to unequal development after the firstly construct the basic charging station over the city. While building stations in the high population density areas will surely alleviate the high pressure from querying for charging, it neglects the demand of the owners to explore nearby areas.

We later come up with the Chess-playing algorithm that can well tackle this trouble. To explain the algorithm, we see the whole Dublin as a chessboard, and the different blocks can be considered as chess nodes. By constructing charging station, the local authority is putting a chess on the chessboard, and each time they build a station, they will receive a score. The purpose of this chess game is to get good marks, and thus the rule which determines the score of each move is of vital importance in this algorithm. To incorpotate the value of blocks as well as the connectivity of the network, the marking rule should adopt distinctions mentioned as follow:

- The more vehicles in the block, the more scores it aquires;
- The more charging station in the block, the less scores it aquires;
- The more vehicles in the block's neighbour blocks, the more scores it aquires;

- The more chargring stations in the block's neighbour blocks, the less scores it aquires;

Based on these notions, we came up with the equation determining the score for each block. It should be noted that the free variable λ reflects the algorith's emphsis on connectivity, and it is possible to be related to the geological pattern and governer's policy. In this section, we simply let $\lambda = 1$.

$$V_{i,j} = \sum_{(x,y)\in U_{i,j}} \frac{D_{i,j}}{1 + \frac{\lambda\sqrt{(x-i)^2+(y-j)^2}}{D_{x,y}}}$$

$$U_{i,j} = \{(x,y) \mid |x-i| \le 1, |y-j| \le 1\}$$

(i-1,j-1)	...	...
...	(i,j)	...
...	...	(i+1,j+1)

Notifications of Marking Equation
1. $V_{i,j}$ denotes the mark of a block
2. $D_{i,j}$ denotes number of stations need to be built
3. $U_{i,j}$ denotes the square neighbouthood of a block
4. If $D_{x,y} = 0$, then $\frac{D_{i,j}}{1+\frac{\lambda\sqrt{(x-i)^2+(y-j)^2}}{D_{x,y}}} = 0$

We later formulate the Chess-playing Algorithm, and later compare its constructing methodicalness with the semi-greed method (see Figure 8 and Figure 9, color reveals only relative density). Although these two methods will lead to the same charging network distribution(see Figure 7), the construction process of them is quite different. Interestingly, one can easily find out that the distribution of charging stations is more even using the Chess-playing algorithm, which means the algorithm well describes the balance of population density and the value of connectivity.

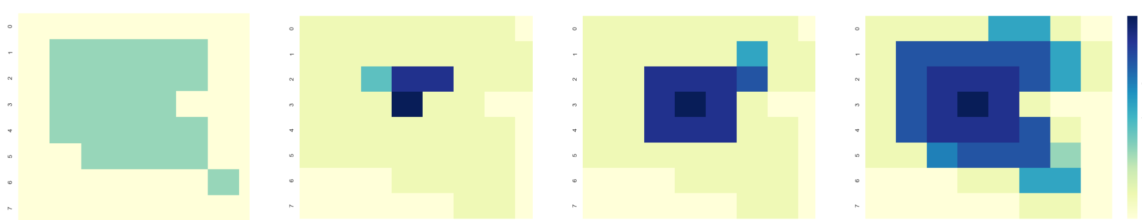


Figure 8: Constructing Process of Semi-greedy Algorithm(10%;30%;50%;90% respectively)

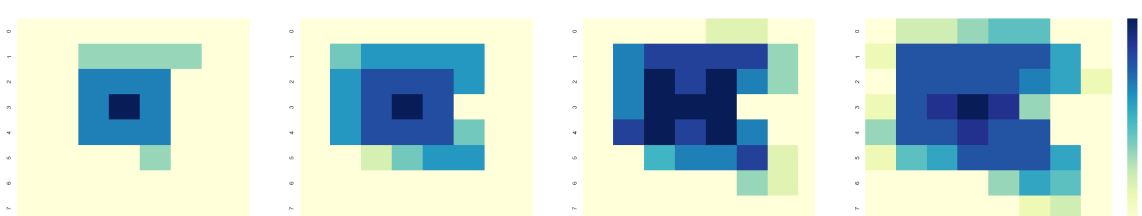


Figure 9: Constructing Process of Chess-playing Algorithm(10%;30%;50%;90%)

4.2.2 Catalytic Behavior of Charging Network: Timeline for the Growth Plan

To estimate the whole timeline for the charging network evolution, a differential equation model should be formulated to describe the transforming rate of electric vehicles. Generally speaking, the speed of transformantion should be positively related to the completeness of charging network construction and the ratio of non-electric vehicles. However, when we attempt to incorporate Logistic model, which is $\frac{de}{dt} = \lambda e(1-e)$, it can not properly predict the transforming rate estimated by ITEDD model[13], and the model is very sensitive to the input data. This means the Logistic model cannot well describe the mechanism behind the transformation.

Out of fortuitous, we notice the simularity between the role of the charging network and the enzyme in a bio-chemical reaction. In a catalyst circulation, the concentration of enzyme is positively related to the speed of reaction, and the substrate, which can be considered as the non-electric vehicles in this case, is transfromed into the product, which can be considered as the electric vehicles. Further, the enzymatic reaction is well described by the classical Michaelis-Menten equation, and thus the differential equation is formulated under the inspiration of this equation.

The Michaelis-Menten equation:

$$v_0 = \frac{K_{cat}[E_t][S]}{K_s + [S]}$$

The Transformation Equation:

$$v_{trans} = -\frac{df}{dt} = \frac{\alpha \frac{C_{sta}}{P_{pop}} f}{\beta + f}$$

Notifications of Transformation Equation
1. f denotes percentage of non-electric vehicles(unit: %)
2. C_{sta} denotes number of current stations
3. P_{pop} denotes population of Ireland (unit: $10 thouandpeople$)

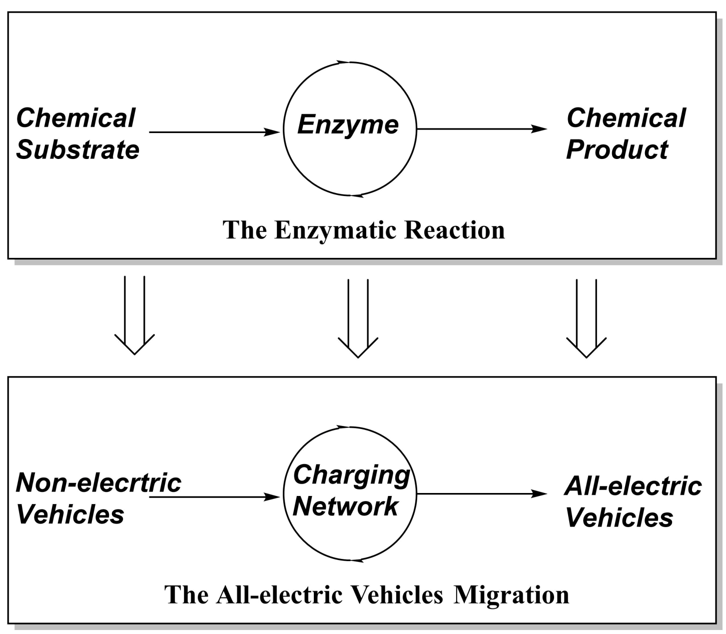


Figure 10: The Michaelis-Menten equation and Transformation Equation

It is interesting to mention, that the $\frac{C_{sta}}{P_{pop}}$ can be regarded as the concentration of the "enzyme", the ratio f can be regared as the concentration of the substrate. Also, from our previous model, one can easily point out that the most proper number of charging station is directly proporational to the ratio of electric vehicles, since either more or less stations will lead to excess supply or demand, and thus C_{sta} is a function of f, and can be written as $C_{sta} = 8.72(100 - f)$.

With the three possible migration pattern purposed in ITEDD model[13], which are defined as *Low PEV Penetration case*, *Reference case* and *High PEV Penetration case*. We each pick data of 2020, 2025 and 2030 to solve variables α and β, and further slove the differential equation. In the equation, f_1 refers to the starting ratio of non-electric vehicles while f_2 refers the final ratio.

Since we only pick 3 data to solve the equation, we can use the unused data of year 2035 and 2040 to exam our enzyme model. Surprisingly, results of our model perfectly tally the unused data in the ITEDD model.

The Growth Rate perdicting equation:

$$t = -\frac{54.7}{\alpha}\left(\frac{\beta+100}{100} ln\left(\frac{100-f_2}{100-f_1}\right) + \frac{\beta}{100} ln\left(\frac{f_1}{f_2}\right)\right)$$

$\diamondsuit$ **Low PEV Penetration case**:

$\alpha = -0.341776639$; $\beta = 1.629728764$

$\diamondsuit$ **Reference case**:

$\alpha = -0.546681674$; $\beta = 2.53155639$

$\diamondsuit$ **High PEV Penetration case**:

$\alpha = -1.623577793$; $\beta = 6.955675891$

⋄ Low PEV Penetration case		
ratio(%)	ITEDD model (y)	Enzyme model (y)
5.50	2035	2035.72
8.12	2040	2042.59
⋄ Reference case		
ratio(%)	ITEDD model (y)	Enzyme model (y)
9.28	2035	2036.31
8.12	2040	2042.91
⋄ High PEV Penetration case		
ratio(%)	ITEDD model (y)	Enzyme model (y)
5.50	2035	2034.85
8.12	2040	2039.63

The great accuracy of our model encourages us to further calculate the timeline of all-electric vehicles migration of Ireland. We assume the starting ratio of all-electric vehicles in Ireland is 0.1 %, and compute the growing timeline.

Table 5: Timeline of All-electric Vehicles Migration

ratio(%)	year(y)(Low)	year(y)(Reference)	year(y)(High)
10	28.98	22.37	14.55
30	72.73	50.93	26.18
50	128.79	86.75	39.51
90	392.12	253.36	98.88
99	766.91	489.83	182.07

Since different policy or unknown issues, the stength of enzyme-substrate interaction varifies, which is the reason why we make 3 types of estimation of the timeline. Interestingly, the results from the enzyme model indicate that the growing rate of all-electric vehicles falls significantly when most of the vehicles have been switched to electric ones, which is reasonable since the ratio of non-electric vehicles becomes very small. Also, we conclude that it will take a long period time for our world to migrate to real all-electric vehicles era under current policy.

5 Dynamic-path-simulating model: Choosing the Optimal Policy

Although enzyme model well predicts the growning rate of the electric vehicle migration, it cannot provide valuable suggestions for policy making. What we can know is only the speed of the migration rate under a given investment policy. In order to generate a classifying system for policy immplementation, parameters such as the governments limited invest ability as well as other subtle details should also be taken into consideration. Herein, a state transformation model was thus established to help identify key factors for policy making.

To explain our model, we assume there are two major ways for governers to trigger the migration. For one way, they can motive the conventional automobile owners to buy new electrical cars. For the other is to invest in charging network and maintain the current electric car owners. However, the governer only have limited resources, and they will have to choose the ratio of their investment in either approach. If the governers only focus on one approach, they will ignore the other aspect and thus the migration rate will probably decrease. To quantify the expected electrical cars ratio in period t(denoted C_t) under particular policy, we need a function with the following characteristics:

- C_t should positivel related to cars buyers who ownered conventional cars in the $t-1$.
- More transformation would occur if the govenment invests more to the new owners.
- Current electrical owners will feel more unsatisfied when the investment to them becomes more scarced, which can cause more drop in C_t. However, supplus can't increase C_t from C_{t-1}.

We come up with the Dynamic-path-simulating model to depict the evolution of the C_t and try to provide with a desirable policy.

$$\max \quad \{C_T\}$$

$$s.t. \quad C_{t+1} = k(1-x_t)(1-C_t) + f(x_t - c_t)x_t$$

$$f(x) = \begin{cases} e^{gx} & x \le 0 \\ 1 & x > 0 \end{cases}$$

The first part on the right hand of the state-transition equation represents the new demand for electric vehicles. The second part describes the maintained part from the last period.The paramater k and b represents the sensibility of the potential buyers and owners respectively. However, since the constrains are very complex, we can't optimize it perfectly. Therefore, we compare some common policies and wish for satisfying solutions.

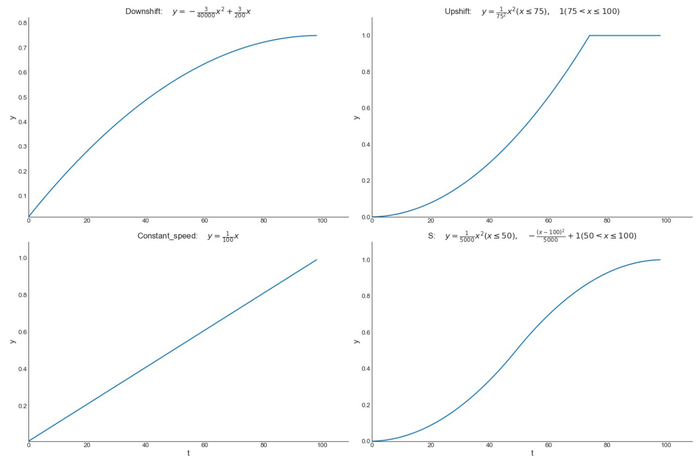


Figure 11: Different Investment Approaches

Above are four different growth models represented by specific functions.It is worth mentioning that since the C_t will likely to reach a rather high level at the end, all the arternate policies should mainly consentrate on the current electrical cars owners. Which remains to disscuss the evolving path to reach that level. Also, all the functions satisfy $\int_1^{100} f(x)dx = 50$ in order to make sure the difference in policies' effect on C_t is purely caused by the model itself.

In addtion to these static models, which make decisions without consideration of the current C_t, we add a simple dynamic strategy $x_t = c_t$as our fifth model denoted as dynamic.

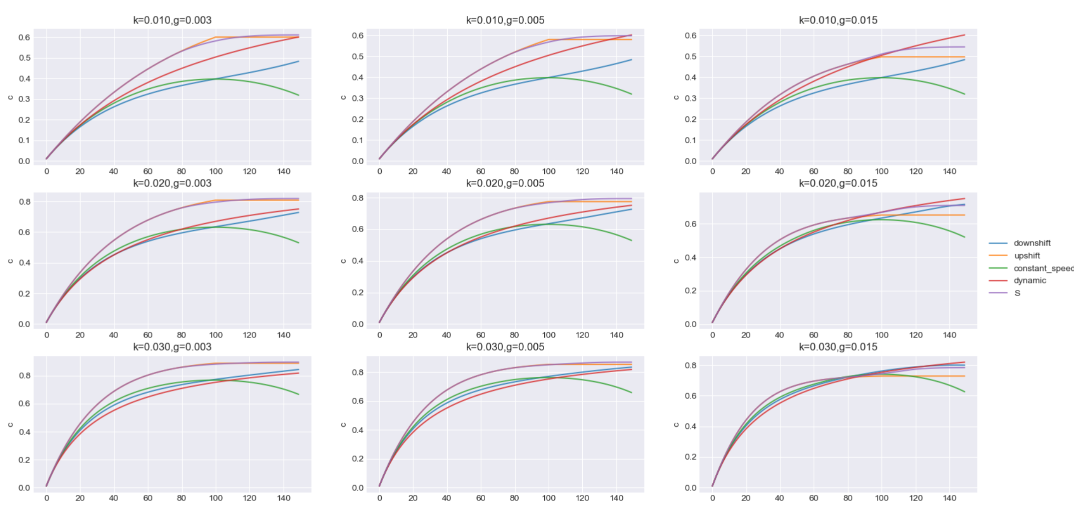


Figure 12: Different Investment Approaches

In Figure 12, we can find that the S and dynamic policy are the most remarkable among all those alternate models. The S policy has the biggest contributions to the C_t in the early periods. However, the dynamic policy shows the greatest potential to help C_t climb up to a new level. There is also some difference between them. With the increasing k, the gap in $C_1$40 given S and dynamic policy becomes larger. When the g grows, the potential distinction between dynamic policy and the others comes clearer.

It's not so hard to understand the result. Since the S policy emphasize the most on the potential buyers in the early period, higher k means the potential owners react more violently, which will accelarate the transfomation better. On the other hand, bigger g means the more dangerous if we neglect the current electrical cars owners' demand. Therefore, response to them percisely is the best way the contribute the C_t, leading to the dynamic policy.

The analysis above also give us way to build up a classification system, determining a country whether they should adopt the S policy or the dynamic one. If a country enjoy high k and low g, then it should choose S policy, otherwise if it is with high g and low k, dynamic policy may be its best choose of growth model.

When it comes to the determinants of the k value, the citizens in the countries with high productivity and technology development level are more likely to have a open mind and enough purchasing power for the electrical cars. However, imbalance in wealth distribution can lead to a lower actual purchasing power.

Due to the substitution effect between the electric vehicles and the public transportation system, a well-developed public transpotation system contributes to a smaller b.

To be specific, we select four indexs, per capita income level(I) and per capita patent amount(P), GINI index(G) and public transportation ridership(R) to build the classification system, that is:

$$score_i = a * I_i * + b * P_i * + c * G_i * + d * R_i *$$

$$I_i * = \frac{I_i}{\max_j \{I_j\}} P_i * = \frac{P_i}{\max_j \{P_j\}} G_i * = 1 - \frac{G_i}{\max_j \{G_j\}} R_i * = 1 - \frac{R_i}{\max_j \{R_j\}}$$

To explain our classification, we choose five countries to analyse.

Table 6: Classification Sytem

country	income(I)	patent(P)	GINI index(G)	transportation(R)
Japan	33894	0.0016	24.9	0.740
U.S.A	57413	0.00094	40.8	0.496
Korea	27539	0.0021	31.6	0.720
Australia	49928	0.00098	34.7	0.05
China	9481	0.00029	61	0.2

(*To simplify we choose $a = b = c = d = 1$)

Thus, we can conclude that Chinese governmenors focus more on encouraging potential buyers, whereas the US, Japan and Korea should attach more importance to charging network construction. In addition, the Australia government should place considerable emphasis on both aspects.

6 Comments on Influence of New Technologies

6.1 Influence of Share-transportations

Car sharing has changed dramatically in the last decade, and it's developing even faster in recent years owing to both new technologies in vehicles and in the software that enables the service[14]. At first glance, one would argue that with the booming development of share vehicles, fewer charging stations will be needed since there will be less cars on the street. However, the demand for vehicle usage will not decrease, which means the share-technology is more like replacing some unfrequently used vehicles with fewer busy cars.

From our perspective, the real effect of share technology is the abrupt transformation it will bring. For frequently used share-vehicles, its upgradation will be much more faster than private ones. Also, sharing-technology corporations will buy more vehicles at one time, which means one may wittness an abrupt upserge of all-electric vehicles. Further, with the notion of going for cleaner, more efficient transportation, sharing-technology corporations may show more interest in the new technologies. In conclusion, share technology will probably boost the migration from conventional vehicles to all-electric ones. For our models, we are expecting to see a even more faster migration than the *High PEV Penetration case*.

6.2 Influence of Charging Technology

In general, we think the development of charging technology mainly falls into two categories. On one hand, the battery-swap technology will significant change the habits of all-electric vehicles users. On the other hand, the development of powerful battery with high energy density may will also make great difference.

For the bettery-swap technology, that one can change his vehicle's bettery at the station, the charging time is significantly shorten. Thus, consider our queuing model which include a parameter of charging time, modification will be necessary when we model on the bettery-swap stations, and the waiting time can be shorten to 10 min or less.

For powerful batteries with high energy density, drivers will not need to charge their vehicles frequently. Once again, in our queuing model, the charging frenquency of drivers is considered, and we may need to reestimate how often will a driver visit the charging station with their vehicles equiped with high energy power pack.

6.3 Influence of Novel Ways of Transportation

Novel ways of transportations, including flying cars, Hyperloops can also influence our model. Although many of these novel ideas are still far from pratical use, we should also consider their effects.

In our opinion, these new technologies are actually repalcing the role that auto mobiles in our life. Assume the development of such new ways of transportations can bring down the average car ownership of man kind, models describing the distribution and the growing patterns of vehicles will be needing appropriate adjustment. For example, there will be less needed charging stations in a specific region, and the construction methodicalness will also change with different distribution of stations needed to be built.

7 Sensitivity Testing

7.1 The Queuing-optimal model

To analyze the sensitivity of Queuing-optimal model, we modify the population data of Ireland by 1 %, and use the Queuing-optimal model to calculate the number of new optimal charging stations. We compare the results to the original ones, and find out that the modification yields no significant difference, and thus the Queuing-optimal model is non-sensitive to the population data. (ROD. denotes "ratio of observed difference")

Table 7: Subtracting from Vehicles Model

Cities	ROD.(%)	Cities	ROD.(%)	Cities	ROD.(%)	Cities	ROD.(%)
Carlow	0	Laois	0	Offaly	0	Clare	0
Dublin	1.2	Longford	0	Westmeat	0	Galway	0
Kildare	2.4	Louth	0	Wexford	0	Leitrim	0
Kilkenny	0	Meath	2.8	Wicklow	0	Mayo	4.2
Roscomm	0	Sligo	8.3	Cavan	7.1	Donegal	0

7.2 The Enzyme model

To analyze the sensitivity of enzyme model, we modify one of the three initiating data respectively by 1 %, and resolved equation respectively. Then we compute the required time for a 95 % migration, and compare the result to the original model. The results suggest that the small changes in the initiating data won't trigger great deviation in the result, and thus the Enzyme model is non-sensitive initiating data.

Table 8: Sensitivity of Enzyme Model

type/data	$f1 \times 101\%$	$f2 \times 101\%$	$f3 \times 101\%$
low penetration case	-3.88 %	+3.04 %	-1.15 %
reference case	-4.57 %	+3.44 %	-0.93 %
high pennetration case	-2.95 %	+2.37 %	-0.68 %

(* f_i denotes the used three data of 2020,2025,2030 to slove the equation)

8 Strengths and Weakness

8.1 Strengths

- We start our investigation based on comprehensive, multi-source data, which is vital for establishing a solid model.
- The hierarchical popolation model provide a powerful approach to handle situations with large differences in population distribution, and thus our model aquires relatively high robustness.

- Incorporate bio-chemistry process into the model, which is a new way of understanding the realtionship between charging network and drivers.
- Incorporate principles of economics in the Dynamic-path-simulating model, which help identify factors for policy making.

8.2 Weaknesses

- Factors including cost of the construction plan are not quantitatively analysised in the models.
- We do not investigate the complex mechanisms behind the enzymatic reaction, and thus do not apply them in our method.
- In the Queuing-optimal-model, we do not study the acceptable charging time of electric vehicles.
- We do not consider the carrying capacity of the local eleric power system.

9 Conclusions

In this paper, we mainly create five models, which are the Queuing-optimal model for estimating charging stations needed to be constructed; the hierarchical popolation model to describe the population destribution of the country ;the Chess-playing algorithm for determining constructing methodcalness; the Enzyme model to predict the timeline of the migration, and the Dynamic-path-simulating model to provide suggested investing patterns for policy making.

We evaluate the connectivity as well as the capacity of Tesla's charging network, and we analyze the required number of charging stations and their distribution in urban suburban and rural areas. We further investigate the migration process of Ireland, and provide optimal construction methodcalness. We also precdict the growing timeline based on the Enzyme model. A classification system was also established in order to help different governers formulate their own plan to successfully migrate away from conventional auto mobiles to all-electric cars.

References

[1] Charles Q. Choi. (2008). The Energy Debates: Electric Cars.
https://www.livescience.com/3172-energy-debates-electric-cars.html

[2] Jack Steward. (2017). WHEN YOU'LL GET YOUR TESLA MODEL 3AND HOW TO GET IT FASTER. https://www.wired.com/story/tesla-model-3-delivery-timeline/

[3] http://money.cnn.com/2017/09/11/news/china-gaselectric-car-ban/index.html

[4] Zhao shuqiang, Li Zhiwei, Dang lei. (2016). Optimal EV charging station sitting and sizing based on urban traffic network information. *Electric Power Automation Equipemnt. 36.10.*

[5] Shao sai, Guan wei, Bi jun. (2016). Charging Location Problem with Queue and Range in Competitive Multi-site Service. *Journal of Transportation Systems Engineering and Information Technology. 2016.16.6*

[6] Joel Hruska. (2017). EPA Reveals How the Tesla Model 3 Gets 310 Miles of Range https://www.extremetech.com/extreme/253939-epa-discloses-teslas-model-3-hits-310-miles-range

[7] Population Density for U.S. Cities Statistics. (2016)
http://www.governing.com/gov-data/population-density-land-area-cities-map.html

[8] Car Ownership in U.S. Cities Data and Map. (2016)
http://www.governing.com/gov-data/car-ownership-numbers-of-vehicles-by-city-map.html

[9] Qian et. al. (2012) Operation Research. ISBN 978-7-302-28879

[10] Charge on the Road.(2018) https://www.tesla.com/supercharger

[11] Population of Ireland.
http://www.cso.ie/px/pxeirestat/Statire/SelectVarVal/Define.asp?maintable

[12] Number of Vehicles in Ireland.
http://www.cso.ie/px/pxeirestat/Statire/SelectVarVal/Define.asp?maintable

[13] Melissa Lynes. (2017) Plug-in electric vehicles: future market conditions and adoption rates
https://www.eia.gov/outlooks/ieo/pev.php

[14] Michael Coates. (2017). 10 Best Car Sharing Programs in USA
http://www.cleanfleetreport.com/best-car-sharing/

Advice on Desgining the Electric Vehicle Growth Model

Based on the research we have done, the key factors worth considering when meeting choices between the various growth models of electric vehicles can be divided into two categories:

- The potential buyers' sensitivity
- The current owners' sensitivity

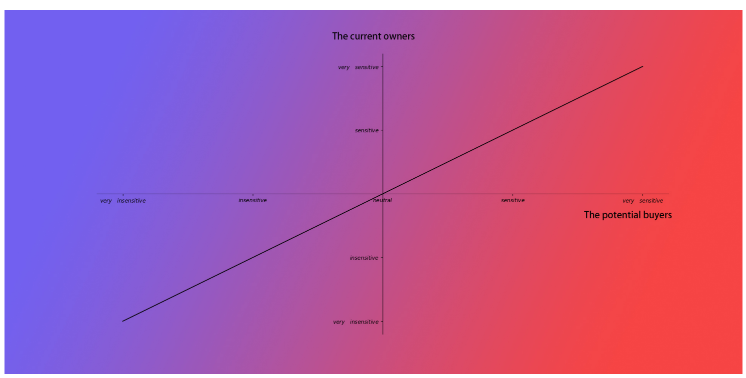
Deside if your country's potential buyers have high sensitivity:

- High income per capita, which means they can have adequate buying power for the electric vehicles.
- High technology development level, which means the country has the potential to produce affordable and desirable cars which will easily attract the potential buyers.
- Great balance in wealth distribution.

Deside if your country's current owners have high sensitivity:

- Poor developed public transpotation. Due to the lack of substitution effect:the electric vehicle owners would suffer great inconvenience when faced with low-level trasportation network and insufficient infrastructure.

With the help of the following graph, we can decide the suitable growth model:



- The redder a country belongs, the more it should concentrate on the potential consumers,such as the subsidy on electric vehicles new buyers or on the industry for further research.
- The more bule a country belongs, the more it should concentrate on the current owners, such as investing more in charging network or bettery technology.

Also it can help set a gas vehicle-ban date:

- For the lucky countries located in the reddest area, they can have a brave try to choose the high-wheeling growth model, which is expected to reach the 50 percentage stage in 40 years.
- For other neutral countries,they can choose a modest speed growth model, which takes more than 65 years to reach the same level.
- As the countries sited in the most blue area, they should be patient and take the low-speed growth model,which takes about 100 years.

APPENDIX

```
Code
#Google API
#coding:utf-8
import urllib
from urllib2 import urlopen
import urllib2
import urllib3
import json
from bs4 import BeautifulSoup
import pandas as pd
import requests

def getLocation(location):
    values={}
    location = location.replace(' ','')
    values['address']=location
    values['password']='AIzaSyAOlDwLTXYOLX2MWCmF2QBpOiES7IPIxZg'
    data=urllib.urlencode(values)
    url='https://maps.googleapis.com/maps/api/geocode/json?'+data
    print(url)
    req=urllib2.Request(url)
    rep=urlopen(req).read().decode('utf-8')
    content=json.loads(rep)
    try:
        lat =content.get('results')[0]['geometry']['location']['lat']
        lng =content.get('results')[0]['geometry']['location']['lng']
        return [location,lat, lng]
    except:
        print('fail')
        return [location,0,0]
if __name__=="__main__":
    metaLocation=pd.read_csv('carownership.csv',encoding='gbk')['city']
    print(metaLocation.head())
    locations=[]
    for i in metaLocation:
        print(i)
        locations.append(getLocation(i))
    locations=pd.DataFrame(locations)
    locations.to_csv('tesla_supercharger_location.csv')

#web crawler for Destination chargers
```

```python
# -*- coding:UTF-8 -*-
from bs4 import BeautifulSoup
import requests
import re
import urllib.request
import pandas as pd

if __name__ == "__main__":
    server = 'https://www.tesla.com'
    target = 'https://www.tesla.com/findus/list/chargers/United+States'
    req = requests.get(url=target)
    html = req.text
    div_bf = BeautifulSoup(html, "lxml")
    div = div_bf.find_all('section', class_='find-us-list-state')
    a_bf = BeautifulSoup(str(div[0]), "lxml")
    toFind = []
    a = a_bf.find_all('a')
    for each in a:
        toFind.append(server + each.get('href'))

    b=a_bf.find_all('span',class_='street-address')
    c=a_bf.find_all('span',class_='locality')
    sa=[]
    lo=[]
    for each in b:
        sa.append(each.string)
    for each in c:
        lo.append(each.string)

    detail=[]
    for local in toFind:
        print(repr(local))
        req = urllib.request.urlopen(local)
        html0 = req.read().decode('utf-8')
        pattern=re.compile(r'</strong><br />(.*?) Tesla Connectors?, up to
(.*?)kW.<br />(.*?)</p> ',re.S)
        items=re.findall(pattern,html0)[0]
        detail.append([items[0],items[1],items[2]])
        print(pd.DataFrame(detail))



#web crawler for supercharger
```

```python
# -*- coding:UTF-8 -*-
from bs4 import BeautifulSoup
import requests
import re
import urllib.request
import pandas as pd

if __name__ == "__main__":
    server = 'https://www.tesla.com'
    target = 'https://www.tesla.com/findus/list/superchargers/United+States'
    req = urllib.request.urlopen(target)
    html = req.read().decode('utf-8')
    div_bf = BeautifulSoup(html, "lxml")
    div = div_bf.find_all('section', class_='find-us-list-state')
    a_bf = BeautifulSoup(str(div[0]), "lxml")
    toFind = []
    a = a_bf.find_all('a')
    for each in a:
        toFind.append(server + each.get('href'))

    b=a_bf.find_all('span',class_='street-address')
    c=a_bf.find_all('span',class_='locality')
    sa=[]
    lo=[]
    for each in b:
        sa.append(each.string)
    for each in c:
        lo.append(each.string)

    detail=[]
    i=0
    for local in toFind:
        print(repr(local))
        req = urllib.request.urlopen(local)
        html0 = req.read().decode('utf-8').replace(' ','')

pattern=re.compile(r'</strong><br/>(.*?).uperchargers,available(.*?),upto(.*
?)kW</p>',re.S)
        items=re.findall(pattern,html0)
        if items:
            item=items[0]
            print([item[0],item[1],item[2]])
            detail.append([item[0],item[1],item[2]])
        else:
```

```
            detail.append([0,0,0])
        i=i+1
        print(i)

#check_connected
#coding:utf-8
import numpy as np
import pandas as pd
def getDistance(x1,y1,x2,y2):
    R=6367
    X1=  R * np.cos(y1) * np.cos(x1)
    Y1 = R * np.cos(y1) * np.sin(x1)
    Z1 = R *  np.sin(y1)
    X2 = R * np.cos(y2) * np.cos(x2)
    Y2 = R * np.cos(y2) * np.sin(x2)
    Z2 = R * np.sin(y2)
    cos0=1-((X1-X2)**2+(Y1-Y2)**2+(Z1-Z2)**2)/(2*R**2)
    return R*np.arccos(cos0)

des=pd.read_csv('Destination_charge(final).csv')[['x','y']]
sup=pd.read_csv('tesla_supercharger(final).csv')[['x','y']].astype(float)
data=pd.concat([des,sup],ignore_index=True)
data=data[data['x']!=0]
data=data[data['y']!=0].values

mindistances=[]
for i in range(len(data)):
    mindistance = 100000
    print(i)
    for j in range(len(data)):
        if(i!=j):
            x1=data[i,0]
            y1=data[i,1]
            x2=data[j,0]
            y2=data[j,1]
            distance=getDistance(x1,y1,x2,y2)
            if(distance<mindistance):
                mindistance=distance
    mindistances.append(mindistance)
mindistances=pd.DataFrame(mindistances)
mindistances.to_csv('min.csv')
```

#solve U.S.A

```
# coding:utf-8
import pandas as pd
from scipy.spatial import Voronoi, voronoi_plot_2d, ConvexHull, KDTree
import matplotlib.pyplot as plt
import numpy as np
from math import factorial as fac

M=1/float(24*7*2)
mu = 2

def calculateT(c,lambda0,pho):
    p0=1/float((sum([1 / fac(k) * (lambda0 / mu) ** k  for k in range(c)])+
1 / fac(c) * 1 / (1 - pho) * (lambda0 / mu) ** c))
    #L = (lambda0/mu) ** c * pho / (fac(c) * (1 - pho) ** 2) * p0
    l=1
    for i in range(1,c):
        l=(lambda0/mu)/i
    L=l*pho*p0/(1-pho)**2
    t_current = L / lambda0
    return t_current

def getDistance(x1,y1,x2,y2):
    R=6367
    X1=  R * np.cos(y1) * np.cos(x1)
    Y1 = R * np.cos(y1) * np.sin(x1)
    Z1 = R *  np.sin(y1)
    X2 = R * np.cos(y2) * np.cos(x2)
    Y2 = R * np.cos(y2) * np.sin(x2)
    Z2 = R * np.sin(y2)
    cos0=1-((X1-X2)**2+(Y1-Y2)**2+(Z1-Z2)**2)/(2*R**2)
    return R*np.arccos(cos0)

data = pd.read_csv('tesla_supercharger(fill_up).csv')[['number', 'x', 'y']]
cityLocation = pd.read_csv('carownership(final).csv')[['x', 'y']]
city = pd.read_csv('carownership(final).csv')
cityTree = KDTree(cityLocation)

destination=pd.read_csv('Destination_charge(final).csv')[['x','y','amouts']]
Cars=dict()
for i in range(len(destination)):
    point=destination.iloc[i]
    nearestCity = cityTree.query([point.x,point.y], p=2)

    city.iloc[nearestCity[1],3]=city.iloc[nearestCity[1]].cars-point.amouts
```

```
#building tree
points = []
for i in range(len(data)):
    points.append(data.ix[i, ['x', 'y']].values)

points = np.array(points)
vor = Voronoi(points, qhull_options='Qbb Qc Qx')
# voronoi_plot_2d(vor)
# plt.show()
point_region = vor.point_region
points = vor.points
regions = vor.regions
vertices = vor.vertices



t_opt_s=[]
c_opt_s=[]
delta_c=[]
segs=[]
for i, point in enumerate(points):
    if(data.iloc[i, 0]):
        print('round %d'%i)
        region = regions[point_region[i]]
        if -1 in region:
            region = [i if x == -1 else x for x in region]
            hullPre = [list(vertices[j]) for j in region]
            # print('Hull:%s'%hullPre)
            hull = ConvexHull(hullPre)
            volume = hull.volume * 2
        else:
            hullPre = [list(vertices[j]) for j in region]
            # print('Hull:%s' % hullPre)
            hull = ConvexHull(hullPre,qhull_options="Qc")
            volume =hull.volume
        # print('point:%s'%point)

        nearestCity = cityTree.query(point, p=2)
        nearestNum = nearestCity[1]
        distance =
getDistance(point[0]/180,point[1]/180,city.iloc[nearestNum,
5]/180,city.iloc[nearestNum, 6]/180)
        c = data.iloc[i, 0]
```

```
volume=min(volume*np.cos(np.pi*point[0]/180)*np.pi**2*(6367/1.603)**2/180**2
,20000)

        cars = city.iloc[nearestNum, 3]
        area = city.iloc[nearestNum, 2]
        t_opt = 0
        density = cars / (area * (1 - np.e ** (-1))) * np.e ** (-np.pi / (70
* area) * distance ** 2) / (70 * 3)
        lambda0 = density * volume * M

        pho = lambda0 / (c * mu)
        t_current = calculateT(c, lambda0,pho)
        seg=1
        while (pho>1 or t_current > 0.5):
            seg=seg+1
            t_current = calculateT(c, lambda0/seg,pho)
            pho = lambda0 / (c * mu*seg)

        segs.append(seg)
        t_opt_s.append(t_current)
        c_opt_s.append(c)
        delta_c.append(c*seg-data.iloc[i, 0])
t_opt_s=pd.DataFrame(t_opt_s).to_csv('t_opt.csv')
c_opt_s=pd.DataFrame(c_opt_s).to_csv('c_opt_s.csv')
delta_c=pd.DataFrame(delta_c).to_csv('delta_c.csv')
```

#check if in urban, sub urban or rural

```
#coding:utf-8
import pandas as pd
from scipy.spatial import KDTree
import numpy as np
from math import sqrt

def getDistance(x1,y1,x2,y2):
    R=6367
    X1=  R * np.cos(y1) * np.cos(x1)
    Y1 = R * np.cos(y1) * np.sin(x1)
    Z1 = R *  np.sin(y1)
    X2 = R * np.cos(y2) * np.cos(x2)
    Y2 = R * np.cos(y2) * np.sin(x2)
    Z2 = R * np.sin(y2)
    cos0=1-((X1-X2)**2+(Y1-Y2)**2+(Z1-Z2)**2)/(2*R**2)
    return R*np.arccos(cos0)
```

```
super_charger=pd.read_csv('tesla_supercharger(opt).csv')
city_location=pd.read_csv('carownership(final).csv')[['x','y']]
city=pd.read_csv('carownership(final).csv')
city_x=city.iloc[:,5]
city_y=city.iloc[:,6]
super_charger_x=super_charger.iloc[:,6]
super_charger_y=super_charger.iloc[:,7]


city_tree=KDTree(city_location)
label=[]
for i in range(len(super_charger)):
    charger=super_charger.iloc[i,[6,7]]
    nearestCity = city_tree.query(charger, p=2)
    nearestNum = nearestCity[1]
    area=sqrt(city.iloc[nearestNum, 2]*2.56*70/np.pi)
    r=sqrt(area)
    city_area=area=sqrt(city.iloc[nearestNum, 2]*2.56/np.pi)
    r0=sqrt(city_area)
    distance = getDistance(charger[0] / 180, charger[1] / 180,
city.iloc[nearestNum, 5] / 180,city.iloc[nearestNum, 6] / 180)
    print(r,distance)
    if(distance<r0):
        label.append('urban')
        print('urban')
    elif(distance<r):
        label.append('suburban')
        print('suburban')
    else:
        label.append('rural')
        print('rural')

label=pd.Series(label).to_csv('label.csv')
```

#solve Dublin step1

```
#coding:utf-8

import pandas as pd
from gurobipy import *
map=pd.read_csv('DublinSolve2.csv')
a=dict()
m=Model('DublinSolve2')
for i in range(len(map)):
```

```
    a[i]=m.addVar(lb=1,vtype=GRB.INTEGER,name='Number_%d'%i)
m.update()

m.setObjective(quicksum((a[i]-map.iloc[i])*(a[i]-map.iloc[i]) for i in
range(len(map))),GRB.MINIMIZE)
m.addConstr(quicksum(a[i] for i in range(len(map)))==247)
m.optimize()
print('obj:%d' % m.objVal)
for v in m.getVars():
    print('%s:%d' % (v.varName, v.x))


#semi-greedy solve Durblin step2

#coding:utf-8
import numpy as np
import pandas as pd

data=pd.read_csv('DublinSolve2.csv')
data['current']=0
data2=pd.DataFrame(data.sort_values(by='Result',ascending=False).values,colu
mns=['Demand','Result','Location','current'])
percent=0.9
toBuild=int(247*percent)

for i in range(min(toBuild,49)):
    data2.iloc[i,3]=1
    toBuild=toBuild-1

k=0
while(toBuild>0):
    data2.iloc[k, 3]=int(min(data2.iloc[k,1]-1,toBuild)+data2.iloc[k, 3])
    toBuild=toBuild-(data2.iloc[k, 3]-1)
    k=k+1

data2['current']=data2['current'].astype('int')
data2['x']=np.floor(data2.Location/8)
data2['y']=data2.Location%8
data2['x']=data2['x'].astype('int')
data2['y']=data2['y'].astype('int')

map0=np.zeros((8, 8))
print(map0)
for i in range(len(data2)):
    x=data2.iloc[i,4]
```

```
    y=data2.iloc[i,5]
    z=data2.iloc[i,3]
    map0[x, y]=z
    print(x,y)
    print(z)
    print(map0[x, y])

import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(map0, fmt="d", cmap='YlGnBu', ax=ax)
plt.show()
```

#dynamic solve Durblin step 3

```
#coding:utf-8
import pandas as pd
import numpy as np
from math import sqrt
import copy
import math
step=1#how fast we choose
theta=1#adjusting factor
length=8#the amount of grid each side
direction_x=[-1,-1,-1,0,0,0,1,1,1]
direction_y=[-1, 0, 1, -1, 0, 1, -1, 0, 1]
percentage=1

def getNearby(x,y):
    nearbyValue=0
    for i in range(len(direction_x)):
        x_new=x+direction_x[i]
        y_new= y + direction_y[i]
        if((x_new<=length) and (x_new>=1) and (y_new<=length) and
(y_new>=1)):

nearbyValue=nearbyValue+1/(1+theta*sqrt(direction_x[i]**2+direction_y[i]**2)
/value[x_new,y_new])
    return nearbyValue

def update():
    for i in range(1,length+1):
        for j in range(1,length+1):
            temp[i, j] = value[i, j]*getNearby(i, j)
```

```
    return temp

#prepare data
metadata=pd.read_csv('DublinSolve2.csv')

#set Location
metadata['x']=np.floor(metadata.Location/8)
metadata['y']=metadata.Location%8
metadata['x']=metadata['x'].astype('int')
metadata['y']=metadata['y'].astype('int')

map=np.zeros((length+2,length+2))
value=np.zeros((length+2,length+2))
value_temp=np.zeros((length+2,length+2))
for i in range(len(metadata)):
    x=metadata.iloc[i,3]+1
    y=metadata.iloc[i,4]+1
    map[x,y]=metadata.iloc[i,1]


#around
for i in range(length+2):
    for j in range(length+2):
        if(map[i,j]==0):
            map[i,j]=0.0000001
        value[i, j] = map[i,j]


#initial
print('initial')
temp=copy.copy(value)



stations=int(sum(metadata['Result'])*percentage)
print(stations)
#optimize
while(stations>0):
    temp = update()
    select=[-1,-1]
    select_value=-1
    for i in range(1,length+1):
        for j in range(1,length+1):
            if(temp[i,j]>select_value):
```

```
                select_value=temp[i,j]
                select=[i,j]

    #update
    value[select[0],select[1]]=value[select[0],select[1]]-1
    stations = stations - step
value=value[1:length+1,1:length+1]
map=map[1:length+1,1:length+1]
station_num=map-value

import seaborn as sns
import matplotlib.pyplot as plt
sns.set()
f, ax = plt.subplots(figsize=(9, 6))
sns.heatmap(station_num, fmt="d",cmap='YlGnBu', ax=ax)
plt.show()
```

#develop model

```
#coding:utf-8

from math import e
import seaborn as sns
sns.set_style('darkgrid')
#sns.despine()
#sns.set(color_codes=True)
year=150
def func(x):
    if x<0:
        return e**(g*x)
    else:
        return 1
a=-3*50/float(2*year**3)
x_1=[a*x**2+(-2)*a*year*x for x in range(1,year)]

x_2=[]
T=3/float(2)*(year-50)
a=1/T**2
for x in range(1,year):
    if(x<=T):
        x_2.append(a*x**2)
    else:
        x_2.append(1)

x_3=[1/float(100)*x for x in range(1,year)]
```

```
x_5=[]
halfyear=year/2
B=(50/float(halfyear)-4/float(3))/(2/float(3)*halfyear**2)
A=(B*halfyear**2+1)/halfyear**2
for x in range(1,year):
    if(x<=halfyear):
        x_5.append(A*x*x)
    else:
        x_5.append(B*(x-year)**2+1)

import matplotlib.pyplot as plt
fig=plt.figure()
ax={}
number=0
for k in [0.02,0.03,0.05]:
    for g in [0.003,0.005,0.015]:
        c_1 = [0.01]
        c_2 = [0.01]
        c_3 = [0.01]
        c_4 = [0.01]
        c_5 = [0.01]



        for i in range(1,year):
            c_1.append(k*(1-x_1[i-1])*(1-c_1[i-1])+func(x_1[i-1]-c_1[i-
1])*c_1[i-1])
            c_2.append(k * (1 - x_2[i-1]) * (1 - c_2[i-1]) + func(x_2[i-1] -
c_2[i-1]) * c_2[i-1])
            c_3.append(k * (1 - x_3[i-1]) * (1 - c_3[i-1]) + func(x_3[i-1] -
c_3[i-1]) * c_3[i-1])
            c_5.append(k * (1 - x_5[i-1]) * (1 - c_5[i-1]) + func(x_5[i-1] -
c_5[i-1]) * c_5[i-1])


        for i in range(1,year):
            c_4.append(k * (1 - c_4[i-1]) * (1 - c_4[i-1]) + func(c_4[i-1] -
c_4[i-1]) * c_4[i-1])


        ax[number]=fig.add_subplot(331+number)
        plt.title('k=%.3f,g=%.3f'%(k,g))
        plt.plot(c_1,label='downshift',alpha=0.8)
```

```
        plt.plot(c_2,label='upshift',alpha=0.8)
        plt.plot(c_3,label='constant_speed',alpha=0.8)
        plt.plot(c_4,label='dynamic',alpha=0.8)
        plt.plot(c_5,label='S',alpha=0.8)
        if(number>=6):
            plt.xlabel('t')
        plt.ylabel('c')
        number=number+1

ax[8].legend(loc='lower center',shadow=True,bbox_to_anchor=(1.2,
1.4),borderaxespad = 0.)
plt.show()
```