

Electric Vehicles Energy Efficient Routing Using Ant Colony Optimization

Rami Abousleiman
Gotion, Inc., Fremont, CA

Osamah Rawashdeh
Oakland University, Rochester, MI

Romi Boimer
Amazon Web Services, Palo Alto, CA

ABSTRACT

Growing concerns about the environment, energy dependency, and the unstable fuel prices have increased the sales of electric vehicles. Energy-efficient routing for electric vehicles requires novel algorithmic challenges because traditional routing algorithms are designed for fossil-fueled vehicles. Negative edge costs, battery power and capacity limits, vehicle parameters that are only available at query time, alongside the uncertainty make the task of electric vehicle routing a challenging problem. In this paper, we present a solution to the energy-efficient routing problem for electric vehicles using ant colony optimization. Simulation and real-world test results demonstrate savings in the energy consumption of electric vehicles when driven on the generated routes. Real-world test results revealed more than 9% improvements in the energy consumption of the electric vehicle when driven on the recommended route rather than the routes proposed by Google Maps and MapQuest.

CITATION: Abousleiman, R., Rawashdeh, O., and Boimer, R., "Electric Vehicles Energy Efficient Routing Using Ant Colony Optimization," *SAE Int. J. Alt. Power*, 6(1):2017, doi:10.4271/2017-01-9075.

INTRODUCTION

Increased government regulations are pushing all major automakers to introduce Electric Vehicles (EVs) as part of their current and future fleets. This will subsequently increase the number of consumer ownership of such vehicles. Propelled by efficient electric motors and powered by high-energy density Lithium-Ion (Li-Ion) batteries, these vehicles are capable of reaching 240 miles of range per charge [1]. Furthermore, EVs have the ability to increase the cruising range by about 20% in a typical urban driving setting through regenerative energy [2]. This percentage can be higher if the EV is driven in hilly areas. Currently, EVs present the best near-term solution to reduce carbon emissions. It is estimated that the U.S. transportation sector accounts for two-thirds of the petroleum use. Thus, shifting consumers towards EVs will reduce total CO₂ emissions by almost half [3]. In Europe 73% of all consumed oil is used in transportation and 25% of CO₂ emissions are from road transportation [4]. All of these reasons are helping the advancement of the EV market, but as with every new technology, there are always some barriers to overcome before the product can be mass sold to the consumers.

Due to the high cost of battery packs, most of the EVs that are released to the market, or are planned to be released in the next couple years will have a limited battery size. This will lead to a driving range of around 100 miles. This relatively low range, along with the scarcity of the charging stations, price factor, and the long

charging times are the major factors that keep consumers away from purchasing EVs. Researchers and engineers from both Original Equipment Manufacturers (OEMs) and academia have been working on increasing the driving efficiency by optimizing the usage of the limited onboard available energy. One of the significant research topics is to optimize the power usage through energy-efficient route planning. Traditional routing algorithms are optimized for fossil-fueled vehicles. These algorithms, including the ones used by Google Maps and MapQuest, use Dijkstra's Algorithm or modified forms of it. Dijkstra's Algorithm solves the shortest path problem with static nonnegative nodes [5].

The EV routing problem cannot be solved using Dijkstra or Dijkstra-like algorithms because EVs are capable of energy recovery through recuperation or regenerative energy thus creating possible negative nodes [6]. This implies that the cost of a path is not the integration of the costs of its edges. Another reason is that edge costs are hard to be pre-computed and thus are only available at query time. Finally, the onboard energy source (Li-Ion high-voltage battery pack) has a pre-known limited capacity and any route calculation should adhere to the minimum and maximum power/energy limits of the battery [7]. For example, consider the simplified routing problem shown in Figure 1. The goal is to start at node 'S' and reach node 'D', apparently there are 2 routes, one through node 'a' and the other is through node 'b'. From observation, both routes have the same cost

of +1. Now assume that a fully charged EV is starting at node 'S' with a battery capacity of +4. Now the route through node 'a' is a more favorable solution as the EV will reach the destination node 'D' with a battery capacity of +4. On the other hand, if node 'b' is selected, the same EV will reach the destination node 'D' with a battery capacity of +2. All these factors prompt researchers to find new and EV suited energy-efficient routing algorithms rather than the more obvious, currently used, and the better understood distance-optimized or time-optimized, route selection. The work presented in this paper will analyze the problem of energy-efficient routing for EVs. The implemented solution will be based on a developed novel method using Ant Colony Optimization (ACO). The presented algorithm will consider the regenerative capabilities of the EV and study the different EV specific parameters that affect the energy-efficient routing solution.

The rest of this paper is organized as follows: the next section will briefly outline the problem statement followed by a related work section. EV configuration, setup, and the foundations of regenerative braking section will follow. ACO will then be discussed with thoughts about varying some of the parameters. The implemented algorithm will be presented afterwards. Simulation and real-world results will then be discussed and analyzed. The paper is ended with a future work section and a conclusion.

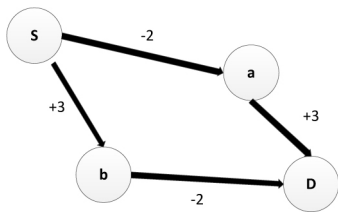


Figure 1. A Simple Vehicle Routing Problem Example

PROBLEM STATEMENT

Drivers usually have significantly large number of routes to pick from when given a start-to-destination routing problem. They are generally interested in routes that will get them to the destination point with the least amount of time and vehicle energy consumption. Evidently, this selection leads to a route with less traffic and shorter traveled distance. Although this selection might not be a global minimum solution, drivers are usually 'OK' with a solution that is close to the global minimum. Furthermore, a solution that gives the least amount of time and the least vehicle energy consumption might not even be viable. A compromise between the two is rendered as an acceptable solution.

In fossil-fueled vehicles, route selection algorithms are targeted towards the selection of a solution with the least traveled time or the shortest traveled distance. This will in most cases lead to fuel-efficient routing because the less the traveled time or the shorter the traveled distance, the less time the engine is run. The less time the engine is run the less fuel it will consume. That is not necessarily true for EVs because these vehicles are capable of regenerating energy. This fact forces the search for novel algorithmic methods that can solve the energy efficient routing problem for EVs. This paper targets the selection of the most energy-efficient route for an EV while

traveling from point 'S' to point 'D' on a given predefined map. The problem of the energy-efficient route selection will be solved based on 6 selection parameters:

1. Elevation changes (braking/regenerative braking)
2. Battery capacity (Max/Min limits)
3. Road travel cost (length and speed limit)
4. Traffic information
5. Ambient temperature
6. Auxiliary loads

Li-Ion batteries installed in EVs need to be operated in a specific temperature range. OEMs tend to cool or heat Li-Ion batteries depending on the ambient temperature [8]. Keeping the battery's temperature within the room temperature range will ensure that its State of Health (SOH) remains high. This fact is of a major concern to OEMs because they want to be able to claim a 10 year warranty on the powertrain components including the battery. To be able to do so, most OEMs install a liquid cooling/heating system [9]. Such systems have power rating of up to 6.6 kw. This substantial power consumption will significantly affect the routing problem formulation and thus it will be considered in this paper. Another implication of the ambient temperature is its effect on the auxiliary loads of the vehicle. Passengers tend to keep the cabin temperature within a comfortable range. This can be an added load on the vehicle's battery. In a conventional fossil fuel vehicle, cabin heat is free as it's already being generated (mostly as wasted energy) by the engine. Cabin heat if needed can be used as soon as the engine gets warm. Air conditioning power on the other hand, is not free on both EVs and conventional cars. On an EV, the air-conditioning system is electric and it's directly powered by the Li-Ion battery, while on a conventional car the compressor is belt-connected to the engine. The cabin temperature control system can also consume up to 6.6 kw of power and again it is included in the routing problem formulation.

Figure 2 shows a simple map on which the algorithm's details will be illustrated. Figure 2 is used for illustrative purposes only and by no means used to prove the convergence of the solution. The vehicle starts at point 'S' and has to reach point 'D'. The numbers between the nodes [l, s] represent the length ($\times 10^3$ meters) and the speed limit ($\times 10^3$ meters per hour) of the section, respectively. It is assumed that there is a traffic light at each node, which means a possibility for a complete stop. This possibility will be calculated randomly with a probability factor of 0.5, calculated at run-time. The elevation at each node is pre-known and it's assumed to have the values listed in Table 1. Point 'S' is used as a reference point for the elevation and thus it has a value of 0. The arrows detail the direction of the flow as some segments are bidirectional while others are not.

Static factors such as the speed limit, elevation changes, length of the nodes, and the beginning battery capacity are all known in advance. Dynamic factors include traffic information, instantaneous EV power consumption, ambient temperature, the instantaneous high voltage battery power availability, and the change in the battery capacity [10]. These dynamic factors causes the exclusion of the preprocessing techniques such as Johnson's Algorithm [11] to be viable methods to solve the EV routing problem.

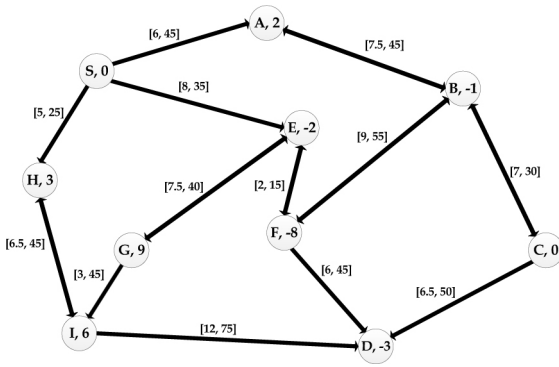


Figure 2. Map to be Used for the EV Route Selection Problem

Table 1. Elevation at each node

Node	Elevation
S	0
A	+2
B	-1
C	0
D	-3
E	-2
F	-8
G	+9
H	+3
I	+6

RELATED WORK

The study of the EV energy-efficient routing problem is not very common due to the fact that Global Positioning Systems (GPS) and electronic maps were not available at economically feasible prices until the early 2000s. Furthermore, OEMs only started to invest in EVs in the past few years. Because of these reasons, the number of studies of the energy-efficient routing of EVs remains very minimal. On the other side, numerous literatures handled energy-efficient routing for conventional vehicles, studied route planning for hybrid electric vehicles [13], tried to solve the Traveling Salesman Problem (TSP), and the Vehicle Routing Problem (VRP).

In previous works [14], [15], and [16] we demonstrated a simplified simulated work to solve the energy efficient routing for EV using different metaheuristic methods. In those works we presented the basic foundation for the simplified algorithms without getting much into details and without proving the implementation with a real-world example. The implemented metaheuristic methods were respectively based on ACO, Particle Swarm Optimization, and Tabu Search.

Other works in the field listed in their chronological order will be listed below. Artmeier *et al.* [17], [18] studied energy-efficient routing for an EV from a graph theory context. They developed a generic shortest path algorithm to solve the problem. The authors analyzed the shortest path problem and adapted the method to fit the specific EV characteristics. Furthermore, they identified hard constraints which declare that the battery cannot be discharged below its limits and soft constraints which enforce that the battery cannot store more energy than its maximum capacity. The developed algorithm had a worst case complexity of $O(n^3)$. The developed algorithm was not proved on an actual vehicle furthermore it did not consider factors such as traffic delays, auxiliary loads, etc... Sachenbacher *et al.* [6]

presented an A* search that considers regenerative energy and the battery parameters which yields a consistent heuristic function of $O(n^2)$ complexity problem. Furthermore, the authors developed the routing algorithm claiming that it expands each node once at most. The presented algorithm adjusted for the battery limits by modifying the weight function that the authors presented. The weight function is generated by the potential energy and the propulsion system inefficiencies. Furthermore, the proposed paper did not consider traffic cost, time delay in longer paths, and it assumed that the map is a perfect environment with the vehicle traveling from the source to the destination with no interruptions. Also the implemented algorithm was never tested on an actual vehicle and it grows at a n^2 rate which can be computationally demanding on embedded computers, especially with the abundant number of route nodes that are usually made available to the drivers. Eisner *et al.* [19] studied optimal routing for an EV. Although they were able to achieve fast processing times, their solution required a preprocessing phase of the desired map. This big disadvantage questions the usefulness of the algorithm in real-time scenarios. The implemented algorithm was also not tested on an actual vehicle and the implemented preprocessing phase meant that dynamic factors such as the ones listed in the previous section were not considered.

Sweda and Klabjan [20] studied the problem of finding the minimum cost path of an EV, when the EV has a target destination but not enough battery capacity or charge to get there. Two methods were studied, the backward recursion approach and the approximate dynamic programming algorithm. However, the authors in this paper did not consider the regenerative energy ability of the EV. Salehinejad *et al.* [21] studied route selection using ACO and they also considered the availability of traffic information by using an online/offline mode. But the developed algorithm was implemented on a conventional fossil fuel vehicle. There are a significant number of works that discussed ACO and its different applications. On the other hand, (as of date and to the understanding of the authors) none studied the EV energy-efficient routing problem. ACO applications that mostly resemble the work presented in this paper include the TSP and the VRP [22]-[32]. The work presented in this paper uses ACO and applies it to the EV example in order to find the most energy-efficient route between 2 points on a given map. In the next section, the EV system of equations representing the energy flow will be described. These equations will ultimately be used in the ACO algorithm to help select the most energy-efficient route.

ELECTRIC VEHICLE

A. EV Power Flow Equations

Fossil-fuel powered vehicle's energy consumption is calculated by the fuel volume that is consumed per unit distance, such as liters per km (l/km). For an electric vehicle, the energy consumption is an integration of the power output at the battery terminals assuming ideal components. The output power (in watts) is shown in (1), where M is the vehicle mass in Kg, g is the gravitational acceleration in m/s^2 , f_r is the tire rolling resistance coefficient, ρ_a is the air mass density in kg/m^3 , C_D is the aerodynamic drag coefficient, A_f is the frontal area of the vehicle in m^2 , K is the vehicle rotational inertia factor, i is the road grade, and dv/dt is the vehicle

acceleration in m/s^2 . Equation (2) shows the regenerative braking power at the battery terminals, where α ($0 < \alpha < 1$) is the percentage of the total brake energy that can be regenerated by the electric motor(s) [33]. The integrated energy shown in (3) is the net energy consumption at the battery terminals. The goal of this paper is to minimize (3), given that the electric vehicle has to drive from one point to another on a given map. Other constraints will be described and added later in upcoming sections of the paper.

$$P_{out} = V \left(Mg(f_r + i) + \frac{1}{2} \rho_a C_D A_f V^2 + M \delta \frac{dv}{dt} \right) \quad (1)$$

$$P_{in} = \alpha V \left(Mg(f_r + i) + \frac{1}{2} \rho_a C_D A_f V^2 + M \delta \frac{dv}{dt} \right) \quad (2)$$

$$Energy_{PROPULSION} = \int P_{out} dt + \int P_{in} dt \quad (3)$$

B. Regenerative Braking

Regenerative braking is the process by which some or most of the vehicle's kinetic energy is converted into another form of energy. Electric motors are capable of dual directional energy flow. This major characteristic permits the energy recovery process [34].

In a conventional fossil fuel powered vehicle, the driver feels mild braking when he/she lets off the throttle. This is caused by engine braking. In an electric vehicle regenerative braking is purposely applied whenever coasting is in progress to simulate the mild braking that people are used to. This fact will regenerate energy back into the vehicle's battery. Another mode where regenerative braking is used is when the foot is applied on the brake pedal. A higher amount of regeneration will be allowed when compared to the mild engine braking feel described earlier.

There are many benefits for using regenerative braking. For example, regenerative braking helps in energy consumption, wear reduction, lower emissions, and better fuel consumption [35]. Regenerative braking is one of the most efficient ways to improve the driving range of an EV [36], [37]. Studies have shown that in urban driving (such as the New York City Cycle) about one third to one half of the energy required for operating the vehicle is consumed during braking. Thus it is very vital for energy conservation if some or most of the braking energy can be retrieved.

C. EVs Assumptions

In order to estimate the energy usage of the EV, some parameters need to be defined. Table 2 shows the parameters that were used in the simulation and analysis. The values in Table 2 are used in (1), (2), and (3) to estimate the energy usage while driving from one node to the other. A typical acceleration rate is 0 to 60 mph (miles per hour) in 15 seconds. That's equivalent to 0 to 26.8224 m/s (meters per second). Using that assumption and because $a = dv/dt$, this means that the acceleration rate is 1.78 m/s^2 . This rate will be used to accelerate

and decelerate the artificial ants throughout the simulations. α as reported in (2) is assumed to have a value of 0.5. This means that only 50% of the regenerative energy can be recovered due to hardware efficiencies and limitations, and due to the fact that the brake regeneration capabilities are not available at low speeds [38], [33]. In a typical real-world scenario, factors including the driver's driving style, limits of the powertrain components, and ambient temperature affect the regenerative braking factors. Dealing with all these variations and assumptions along with the energy efficient routing will be a tedious task. Furthermore, the regenerative braking element can be affected by numerous stochastic factors making it even harder to calculate with exact precision. For these reasons, our assumptions are based on a standard driving style as well as typical powertrain component parameters. Future work may include the implementation of a dynamic regenerative braking estimation and embedding that estimation in the ACO algorithm that is generated in this paper.

Table 2. Vehicle parameters used in the simulations

Description	Symbol	Value	Unit
Aerodynamic Drag Coefficient	C_D	0.25	—
Frontal Area	A_f	2.25	m^2
Rolling Resistant Coefficient	f_r	0.005	—
Vehicle Mass	M	1,500	kg
Gravitational Acceleration	g	9.8	m/s^2
Air Mass Density	ρ_a	1.275	kg/m^3
Rotational Inertia Factor (mass factor)	δ	1	—
Regenerative Braking Factor	α	0.5	—

BACKGROUND

ACO is a branch of swarm intelligence constituted of metaheuristic optimizations and is generally used to find solutions to hard combinatorial optimization problems. ACO was first introduced by Marco Dorigo in his PhD thesis in 1992 [39]. ACO is inspired by ants and their foraging behavior [40]. Ants deposit chemicals, called pheromones, on the ground. These chemical traces are then used by the entire ant colony to help navigate either towards their nest or towards a food source. Ants' navigational behavior is based on autocatalysis (exploitation of positive feedback). This behavior helps the ants to find the shortest path between a food source and the nest [41].

Deneubourg *et al.* investigated the ant's behavior [42] and their work was the benchmark for the different ACO algorithms that were proposed in the early 1990's. Deneubourg's experiment, also known as the "double bridge," features a nest of Argentine ants that were put by the edge of 2 different bridges. Each ant has the option of selecting either one of the bridges' side. Figure 3 shows the bridge geometry. The 2 bridges are identical in terms of length and structure. Using Monte Carlo simulations, the authors were able to define the ants' selection of either side of the bridge. After N ants crossed the bridge there are M pheromone units on bridge side A and $N - M$ on bridge side B . The next ant arriving at the bridge has to make a selection between A or B . The selection is based on the amount of pheromone present at the branches. If $M > N - M$, the next ant will most likely select branch A . After this selection, the ant then adds to the already existing pheromone of the chosen branch. Due to random fluctuations

and after some time, one side of the bridge will surmount the other and a majority of the ants will most likely make the same selection. Figure 4, on the other hand, shows a different 2-branch bridge, but this time the 2 branches are not identical. One branch is shorter than the other. It is obvious that the shorter branch *A* is the optimal selection to conserve both time and energy. At the start of the test and as soon as the ants are given the choice to pick a branch, the short branch *A* gets selected by around 50% of the ants due to chance. Ants that select branch *A* reach the food and head back to the nest faster, thus increasing the amount of pheromone. The amount of pheromone on branch *A* increases more rapidly than branch *B* causing most ants to ultimately select it. The probability of selecting branch *A* is formulated by Goss in 1989, as follows [43]:

$$P_A = \frac{(m_1 + k)^h}{(m_1 + k)^h + (m_2 + k)^h}$$

m_1 and m_2 represent the number of ants that have crossed branches *A* and *B* respectively. The values of k and h are fitted experimentally. The next section will discuss in more details the deviations and the implications of changing k and h . Finally the probability of selecting branch *B* is given by $P_B = 1 - P_A$

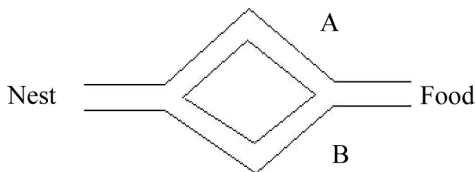


Figure 3. Identical Bridge Branches

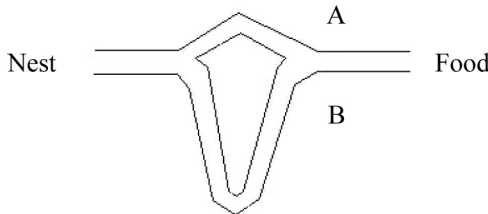


Figure 4. Bridge Branches with Different Lengths

ANT COLONY OPTIMIZATION

A. Studying the Effects of Changing the 'h' and 'k' Terms

The probability of any given ant selecting branch n is given by (4). Where $h \in (\mathbb{N}_{>0})$ is a non-zero positive natural number and $k \in (\mathbb{R}_{>0})$ is included in the set of positive real numbers. m and n are included in the set of natural numbers $\{0, 1, 2, 3, \dots\}$. m_n represents at any given time the amount of ants that selected branch n . N is the total number of branches (at any given node) that is available for a specific ant to pick from, where $N \in \mathbb{N}_{>1}$ and $n \leq N$. Assuming that the drivers are only interested in energy-efficient routing the *DesirabilityFactor* will be ignored and will be set to 1.

$$P_n = \frac{(m_n + k)^h}{(m_1 + k)^h + (m_2 + k)^h + \dots + (m_n + k)^h + \dots + (m_N + k)^h} \times \text{DesirabilityFactor} \quad (4)$$

Rewriting (4) yields the following:

$$P_n \times \{(m_1 + k)^h + (m_2 + k)^h + \dots + (m_n + k)^h + \dots + (m_N + k)^h\} = (m_n + k)^h$$

Divide by $(m_n + k)^h$ we get

$$\frac{P_n}{(m_n + k)^h} \times \{(m_1 + k)^h + (m_2 + k)^h + \dots + (m_n + k)^h + \dots + (m_N + k)^h\} = 1$$

Divide by P_n we get

$$\frac{1}{(m_n + k)^h} \times \{(m_1 + k)^h + (m_2 + k)^h + \dots + (m_n + k)^h + \dots + (m_N + k)^h\} = \frac{1}{P_n}$$

Rewriting the equation yields

$$\frac{(m_1 + k)^h}{(m_n + k)^h} + \frac{(m_2 + k)^h}{(m_n + k)^h} + \dots + \frac{(m_{n-1} + k)^h}{(m_n + k)^h} + \frac{(m_{n+1} + k)^h}{(m_n + k)^h} + \dots + \frac{(m_N + k)^h}{(m_n + k)^h} = \frac{1}{P_n} - 1$$

Adding the terms and multiplying both equation sides by $()^{-1}$ yields

$$\left[\frac{(m_n + k)}{(m_1 + m_2 + \dots + m_{n-1} + m_n + m_{n+1} + \dots + m_N + (N-1) \times k)} \right]^h = \frac{P_n}{1 - P_n} \quad (5)$$

$$\text{Let } Z = \frac{P_n}{1 - P_n}$$

As P_n approaches 1, Z approaches ∞ , P_n has a range from 0 to 1 and therefore Z has a range from 0 to ∞ . Furthermore, Z is monotonically increasing function that follows the increase of P_n . Considering (5) it is observed that the effect of 'h' is to always favor the stronger node. An example of this would be when the numerator is greater than the denominator. This will lead a $Z^{1/h}$ value greater than zero. This condition occurs when a given node n has had a higher ant count than the rest of the nodes combined + $(N - 2) \times k$ (The effect of the change of 'k' will be discussed in the next section). The 'h' term in this case will yield a stronger probabilistic selection chance for m_n . Furthermore the higher the value of 'h' the stronger the probabilistic selection chance is. On the other hand, if the numerator value of (5) is less than the denominator, this will lead to a $Z^{1/h}$ value between 0 and 1. The 'h' term in this case will yield a weaker probabilistic selection chance for m_n . Thus, selecting high values for 'h' will evidently cause the algorithm to get stuck in a local minimum solution. Furthermore, the higher the value of 'h', the higher the computational complexity of the mathematical operation will be. For these 2 reasons, we selected the 'h' value to be 1.

The effect of changing 'k' is very minimal in terms of the computational complexity because it's only used in the addition term. It is worth mentioning that when a given node n has had the same ant count as the rest of nodes combined + $(N - 2) \times k$ then the term 'k' has no effect on the probabilistic selection and node n 's selection probability is 0.5. The effect of changing 'k' changes with the change