Electric Vehicle Routing Problem

Andrew Struthers

June 2023

Honor code: We pledge that we have neither given nor received help from anyone other than the instructor or the TAs for all work components included here. – Andrew

Introduction

Electric vehicles (EVs) have gained significant attention in recent years as a sustainable solution to reduce greenhouse gas emissions and dependence on fossil fuels. Efficient utilization of EVs involves not only developing advanced battery technologies but also optimizing their routing to maximize their range and minimize energy consumption. The Electric Vehicle Routing Problem (EVRP) addresses the challenge of determining the most efficient routes for a fleet of EVs to serve a set of customer locations while considering their charging requirements.

This report presents the use of genetic algorithms (GAs) to optimize the EVRP. Genetic algorithms are a class of heuristic search algorithms inspired by the process of natural selection and genetics. They offer an effective and flexible approach to solving optimization problems, making them particularly suitable for addressing complex and dynamic routing problems like EVRP.

The primary objectives of this study are to describe the EVRP, establish benchmarking criteria, and evaluate the performance of genetic algorithms in comparison to other state-of-the-art methods. By applying GAs to the EVRP, we aim to achieve optimal routing solutions that consider both the customers' demands and the limited range of EVs.

The primary source of information I used for this project came from "The Electric Vehicle Routing Problem and its Variations: A Literature Review", published in "Computers & Industrial Engineering" by Ilker Kucukoglu, Reginald Dewil, and Dirk Cattrysse[1]. This source compiles 136 published papers that consider the routing of battery electric vehicles. We will also be using the same dataset as the one discussed. We will be focusing on the most basic version of the EVRP. The commonly used datasets can be found here

Problem Description

Overview

The Vehicle Routing Problem (VRP) is a classical optimization problem that deals with determining optimal routes for a fleet of vehicles to serve a set of customer locations. The objective of the VRP is to minimize the total distance traveled or the total cost while satisfying various constraints, such as vehicle capacity limits and time windows for customer visits.

As an extension of the VRP, the Capacitated Vehicle Routing Problem (CVRP) incorporates the additional constraint of limited vehicle capacity. Each customer location has a demand associated with it, and the vehicles must not exceed their capacity while making deliveries. The CVRP has been extensively studied and has various practical applications, such as goods distribution and waste collection.

With the rise of electric vehicles and the need for sustainable transportation solutions, the Electric Vehicle Routing Problem (EVRP) has emerged as a specific variant of the VRP. The EVRP addresses the unique challenges associated with EVs, such as limited driving range and the requirement for charging infrastructure. The objective of the EVRP is to optimize the routes of electric vehicles to minimize energy consumption or maximize the range covered while considering charging station locations and customer demands.

Key Challenges

The EVRP inherits challenges from its parent problems, the VRP and CVRP, while introducing additional complexities:

1. Limited Range: Unlike conventional vehicles, EVs have a limited driving range due to their reliance on battery power. The EVRP requires careful route planning to ensure that EVs can complete their routes without running out of energy. This constraint necessitates efficient utilization of available charging stations and strategic placement of the routes to minimize range anxiety.

2. Charging Infrastructure: The availability and location of charging stations play a critical role in the EVRP. EVs need access to charging stations to recharge their batteries and extend their driving range. Optimizing the charging station usage, including factors like charging time and station capacities, becomes an integral part of the EVRP.

3. Customer Demand: Similar to the VRP and CVRP, the EVRP must consider the varying demands of customers at different locations. Efficient routing strategies should aim to minimize the total distance traveled and ensure timely delivery while considering customer preferences and requirements.

4. Time windows: Most common interpretations of the VRP, CVRP, and EVRP have customers with a set serviceable time window. To successfully meet the demand of each customer, the vehicle must arrive within a time window, and the model must simulate travel time to build an accurate time table.

5. Dynamic Factors: Real-world scenarios introduce dynamic factors, such as traffic congestion, changing customer demands, and unpredictable charging station availability. These factors further compound the complexity of the EVRP and require adaptive algorithms capable of handling dynamic environments.

In this report, we will be focusing on the most basic version of the EVRP, only taking into account limited vehicle range, charging infrastructure, and customer demand.

Objectives of the EVRP Optimization

The primary objective of optimizing the EVRP is to develop efficient routing strategies that maximize the utilization of EVs while minimizing energy consumption and total travel distance. The optimization process should consider the constraints imposed by the limited range of EVs, the charging station infrastructure, and the varying customer demand.

The EVRP variation used in this report can be defined by the following mathematical model

- Notations

- $0, N+1$
  - Depot nodes
- F
  - Set of charging stations
- F'
  - Set of dummy nodes required to allow multiple visits to a charging station in the set F
- V
  - Set of customers; $V = \{1, 2, \ldots, N\}$
- V_0, V_{N+1}
  - Set of customers and depot node; $V_0 = V \cup 0, V_{N+1} = V \cup N+1$
- V'
  - Set of customers and charging stations; $V' = V \cup F'$
- $V'_0, V'_{N+1}, V'_{0,N+1}$
  - Set of customers, charging stations, and depot node; $V'_0 = V' \cup 0, V'_{N+1} = V' \cup N+1, V'_{0,N+1} = V' \cup 0 \cup N+1$
- K
  - Set of vehicles
- d_ij
  - Traveling distance from node i to node j; $\forall i, j \in V'_{0,N+1}$
- h
  - Energy consumption rate of the vehicles per unit distance
- Q
  - Battery capacity of the vehicles

• Decision variables

- x_{ij}^k
  - Binary variable and equal to 1 if vehicle k travels from node i to node j, 0 otherwise; $\forall i, j \in V'_{0,N+1}, i \neq j, d_{ij} > 0, \forall k \in K$
- y_i^k
  - Decision variable to track the battery level of vehicle k on arriving at node i; $\forall i \in V'_0, \forall k \in K$

• Objective function

$$\text{Min}z = \sum_{i \in V'_0} \sum_{j \in V'_{N+1}} \sum_{k \in K} d_{ij} x_{ij}^k$$

- Subject to

$$\sum_{j \in V_{N+1}} \sum_{k \in K} x_{ij}^k = 1 \qquad \forall i \in V \tag{1}$$

$$\sum_{j \in V_{N+1}} \sum_{k \in K} x_{ij}^k \leq 1 \qquad \forall i \in F' \tag{2}$$

$$\sum_{j \in V'} x_{0j}^k \leq 1 \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in V_0'} x_{ij}^k = \sum_{i \in V_{N+1}'} x_{ji}^k \qquad \forall j \in V', \forall k \in K \tag{4}$$

$$y_j^k \leq y_i^k - (h \cdot d_{ij})\, x_{ij}^k + Q\left(1 - x_{ij}^k\right) \qquad \forall i \in V, \forall J \in V_{N+1}', \forall k \in K \tag{5}$$

$$y_j^k \leq Q - (h \cdot d_{ij}) x_{ij}^k \qquad \forall i \in F' \cup 0, \forall j \in V_{N+1}', \forall k \in K \tag{6}$$

$$y_o^k \leq Q \qquad \forall k \in K \tag{7}$$

The objective function of this mathematical model aims to minimize the total distance of the electric vehicles. Constraint (1) handles the connectivity of the customer nodes and constraint (2) guarantees that each charging station can be visited at most once. Constraint (3) forces each vehicle to only be used once, i.e. the same vehicle doesn't go on two routes. Constraint (4) allows continuity of the routes, meaning that the outgoing routes are equal to the incoming routes at each customer and charging station. No vehicle should ever instantly teleport from charging station 1 to customer 4 during its route. Constraint (5), (6), and (7) track the battery level of the vehicle as well as the state of the battery after recharging, assuming a full charging policy instead of a partial charging policy. In addition to these 7 constraints, there are 15 others that could be used to further complicate the model, by considering metrics such as weight of the vehicle, partial charging policies, serviceable time windows, and homogeneous vs heterogeneous fleets. Since we are considering the simplest version of the EVRP, we will only use the constraints mentioned above.

The use of genetic algorithms (GAs) offers a promising approach to tackle the EVRP. GAs leverage concepts from natural evolution, such as selection, crossover, and mutation, to iteratively improve solutions and converge towards an optimal or near-optimal solution. By applying GAs to the EVRP, we aim to leverage their ability to handle complex and dynamic optimization problems, addressing the challenges outlined above. The genome sequence that we will use will be an ordered list of the customer nodes visited in the complete tour. When evaluating the fitness of each genome sequence, we will pad the sequence with the starting location of the depot, the ending location of the depot, visits to the depot to restock, and visits to nearby charging stations when required. These pieces of information should not be encoded into the genome sequence as they are unique and only determined by the desired customer route. It would make little sense, for example, to put trips to the depot into the sequence as trips to the depot aren't necessary unless the customer route deems them necessary, nor are they what we are trying to minimize.

Implementation

Common ways to solve the EVRP include Adaptive Variable Neighborhood Search (VNS/AVNS), Adaptive Large Neighborhood Search (LNS/ALNS), Adaptive or Greedy Tabu Search (TS/ATS/GTS), or Genetic Algorithms (GA). Because of my familiarity with GAs, I opted to implement this method over other more commonly used methods. As described above, I implemented selection, crossover, and mutation to work towards an optimal or near-optimal solution. The genome sequence consisted of the ordered list of customer nodes in the route. This was done using C++ with only standard libraries. I chose to write this code in C++ for the speed benefits over other programming languages like Python, as well as C++ requiring me to fully understand how to implement not only the objective functions and math described in the previous section alongside a GA.

For selection, I chose to use tournament selection where the top parent out of n parents got selected. Crossover needed to be implemented in such a way where the uniqueness of the values in the genome sequence was kept, so I used Single Point Crossover. A random number of elements were selected from the first parent, then the rest of the elements were filled with unique elements from the second parent. This allowed the genome sequence to evolve without getting multiple of the same customers. For mutation, two random elements of the genome sequence swapped places. I used a population size of 100, with 1000 max generations, a tournament size of 10, and a mutation rate of 0.15. I played with these variables, and I found that the search spaces of the datasets used were small enough that 1000 generations was a nice balance of execution time and finding the optimal result. If larger datasets were used, I would suggest increasing the population size and tournament size before increasing the max generations.

Data and Benchmarking

Throughout the 136 reviewed articles published in the space of EVRP, the most widely used dataset consists of small and large sized instances with either 5, 10, or 100 customers. Customers and charging stations can be distributed by either random distribution, clustered distribution, or a mix of both random and clustered distributions.

StringID	Type	x	y	demand
D0	d	40.0	50.0	0.0
S0	f	40.0	50.0	0.0
S1	f	77.0	52.0	0.0
S3	f	57.0	82.0	0.0
S16	f	48.0	8.0	0.0
S20	f	93.0	43.0	0.0
C98	c	58.0	75.0	20.0
C78	c	88.0	35.0	20.0
C4	c	42.0	68.0	10.0
C13	c	22.0	75.0	30.0
C95	c	62.0	80.0	30.0
C100	c	55.0	85.0	20.0
C54	c	42.0	10.0	40.0
C27	c	23.0	52.0	10.0
C89	c	63.0	58.0	10.0
C96	c	60.0	80.0	10.0

Figure 1: Sample EVRP Instance. Nodes with type d represent the depot, type f represents charging stations, and type c represents the customers

Each dataset follows the same constraints: one charging station must be located at the depot, and the remaining charging stations are distributed randomly with the assumption that each customer could be reached from the depot by visiting at most two charging stations. The literature review reports two solvers, CPLEX and GUROBI, and their respective benchmarking data for obtaining the optimal solution on each dataset. I benchmarked my code using a handful datasets.

Dataset	CPLEX	GUROBI	TS/VNS	My Solution
C101_5	0.98	0.29	0.21	64.3
R103_10	4.8	1382.0	0.72	71.2
RC204_15	7200	7200	15.57	88.4

Each solution was restricted to a 2 hour runtime, so the two 7200s runtimes indicate that the solvers didn't converge and no solution was found. In both cases, my solution was close to the optimal but was not fully optimal. I believe this is due to not having a larger population size or enough generations. The RC204_15 dataset, for example, had an optimal overall tour distance of 384.86, and my code found a route with distance 395.528. I also didn't implement the time window constraints, even though the datasets would have supported this. Not including the time window constraints could also add to the discrepancy between my solutions and the solutions found in the literature review.

Results

By comparing some of the runtimes with those found in the literature review, it is clear that my GA solution is generally much slower than other methods. However, if we look at the average solve time for each of the solvers on all of the small size instances (5, 10, and 15 customers), we can see that my code performs better in general than the two numeric solvers used. In comparison to the TS and VNS algorithms, my GA solution is much slower.

	CPLEX	GUROBI	TS/VNS
Average	1216.97	1286.64	5.03

In general, my code was generating similar tours as the optimal routes found in the literature review, albeit without the time window constraints. This is a promising result, as this problem is very complicated. I am quite happy to be able to implement some computationally intelligent system to solve the EVRP with moderate success relative to other solutions.

Discussion

In the future, I would change a few things about how I solved this problem. First, I would have an early exit condition in my genetic algorithm code. As far as I am aware, the optimal solution could have been found on an earlier generation, but my code continues executing for n generations regardless of the fitness of the solution. Second, I tried to implement VNS to solve this problem, as this was the most common and fastest way to solve this problem. I had difficulties implementing that solution, however, because I was unfamiliar with both the problem and the implementation. I stuck with GA because of my familiarity with the process for programming a GA, so that I could focus entirely on solving the problem. In the future I would like to focus more on other implementations that could be better, and not limit myself to only what I already know. Third, there are many more constraints possible with this dataset, mainly the time window portion. It was difficult to do a direct comparison between my results and the results presented in the literature review because all of their code solved the EVRP with time window constraints, making it technically the EVRPTW. I went with the simplest version, due to the complexity of just implementing the basic version of the EVRP with as few constraints as possible. In the future I would like to expand this to at least include time windows.

References

[1] Ilker Kucukoglu, Reginald Dewil, and Dirk Cattrysse. The electric vehicle routing problem and its variations: A literature review. *Computers & Industrial Engineering*, 161:107650, 2021.