Experimental Evaluation

1. Electrical vehicles description

For the experiments, a fleet of ten real Electric Vehicles was used with a variety of different state of charge within the limits of each car. For performing experiments with a greater number of data, the same 10 vehicles were multiplied with respect to their ration in the real data but used random state of charge according to the curved used.

The initial data-set which was scaled is described as follows:

- Three Renault ZOE with 22 kWh battery capacity (Type 2 maximum charge power: 22 kW);  
- Four Renault ZOE with 41 kWh battery capacity (Type 2 maximum charge power: 22 kW);  
- Two Nissan LEAF with 24 kWh battery capacity (Type 2 maximum charge power: 7 kW);  
- One Hyundai KONA with 64 kWh battery capacity (Type 2 maximum charge power: 11 kW).

As for data source, in the beginning we used an MQTT broker for Italian pilot charging station real-time data collection project until we got the specific stable data about the Electric Vehicles such as:

vehicle: electric vehicle id  
brand: electric vehicle brand  
model: electric vehicle model

battery percentage : state of charge of the electric vehicle  
battery capacity: electric vehicle battery capacity (kWh)

From these data, the only information that varies is the battery percentage which was randomly generated after retaining all the data in a database.

Also, to be able to apply the constraint, each EV has some particular data specified by the driver such as:

* The favourite charging station
* The availability time interval of the driver
* The order of importance of the two preferences stated above

After saving all this information in a database, the format in which the data is preprocessed by the program is JSON data, and the output respects also this format.

1. Charge stations description

For charging stations, the location is not as important for our problem because we let the driver choose his favorite charging station and we did not do an automatic assignation to the closest CS. But nevertheless, we assigned to each station in the experiment a physical location from Cluj-Napoca where a real charging station with the specification from the data set exists.

Our initial data set provided also by the MQTT broker had 2 Level 2 Charging Stations, each with 2 plugs and Type 2 chargers which operate at 208-240 V and output anywhere from 3 kW to 19 kW of AC power, having the following specifications:

Charging stations:  
- Manufacturer: EMOT  
- Model: Spot Link EVO  
- Nominal Voltage: 400 V AC THREE-PHASE  
- Nominal Frequency: 50 Hz  
- Current: from 7 A to 64 A  
- Sockets: two Type 2  
- On-board Computer: Raspberry Pi 3

But for our experiments where we had to test bigger scenarios, we considered more charging stations with the same specifications.

As for the operational hours, the ones in Cluj-Napoca have no restrictions as they are in public, easy-accessible spaces and they are like a parking lot at the side of a main road. But, for our experiment, we either considered the time interval for scheduling to be 8 AM – 6PM, 11AM to 2PM or 11PM to 4PM.

1. Description of the renewable electricity production curves

In order to obtain the electric power difference that needs to be balanced by the value charged/discharged by cars, we used two different energy curves provided by The California Independent System Operator (ISO) is a nonprofit public benefit corporation that operates the high-voltage transmission grid for the state of California. It is responsible for ensuring the reliable operation of California's bulk electric power system, managing the flow of electricity across the transmission lines, and maintaining the balance between electricity supply and demand in real-time. We always use as a baseline the energy curves from a day before.

From the supply curves, we take into account from the Renewables trend,( <https://www.caiso.com/TodaysOutlook/Pages/supply.html>) the curve of the Solar energy, which is expressed in MegaWatts. The Energy in megawatts is broken down by renewable resource in 5-minute increments, but we consider only the values at exact hours (eg. 8:00AM, 9:00AM etc.) so the curves are reproduced hourly.

Also, considering that the value is in MW and represents an entire state, but our fleet has between 10 and 80 vehicles, depending on the experiment, we considered dividing this values by 1000 to have kW, as that is the measurement unit for the power in cars. But even that was too much for the actual capabilities of our cars, considering that at most a car can charge is between 22 – 61 (depending on the capacity) and that will be if they were almost discharged. Because of that, we considered dividing the now value in kW furthermore by 100 or 150, depending on the size of the test data. The solar curve has the form of a Flattened-Gaussian Distribution, as the peak of the values is during the daytime, between 8AM and 5PM, and outside this hours, the values are significantly lower. In winter time, the values are lower, but the curve preserves its shape.

The Demand curve(<https://www.caiso.com/TodaysOutlook/Pages/index.html>) is provided by the same agency and in terms of measurement units the same mechanism was applied as for the demand curve. The difference is that the shape is not that specific and may vary more from day to day, but it is close to a bimodal distribution, with its peaks in the morning and in the afternoon-evening.

1. Data Preprocessing

The preprocessing for experiments consisted of mostly changing the state of charge for each car and adapting the preferences of the driver and of course, scaling the data. The ration between the cars was preserved as in the initial data set. Also, from the initial MQTT API we had more fields provided, but they were not needed for our experiment, so they were not saved in the database. The state of charge was changed considering the experiments we wanted to do, considering the charging operation, the SOC varied between 15% and 60%, while considering the discharging operation the SOC varied between 99% and 65%. As for the driver’s preferences regarding the charging stations, the ratio between the preferred charging station was maintained nearly equal.

The time availability interval was more flexible and were random time intervals in the available schedule, with some of them being available the entire time and other just one or two hours, but really a lot of variations in between.

Another aspect was the order preference of these two constraints. We used individual penalty for breaking this constraints, but we kept unity among the penalty. Meaning that the driver could choose the preference order (eg charging station over time interval) but this was reflected among the penalties. For example, we give a penalty of 5 to the most important constraint of each driver and a penalty of 3 to the second one. The ration between the preferences was also maintained nearly equal.

1. Experimental setup

For the experiments, the energy curves for either supply or demand situation considered were the ones from 28th of May 2023.

The algorithm starts from an initial set of random solutions which cannot be kept the same each time for the same data (same energy curve and same electric vehicle’s specific values) because how the algorithm does its initial repartition in a solution is that it takes a random vehicle from the ones that were not already assigned and check if there is the possibility of assigning it at the desired Charging Station within the time interval in witch the driver is available, and if not, the vehicle is assigned at a random free position.

The number of iterations also can vary from run to run, as if no improvement of the Fitness Score was done in the last 10 iteration the algorithm stops. What I kept constant almost for every run except Scenario 1 and 5 is the number of the initial solution set size. For the bigger scenarios, the sample size was kept 50, but for the smaller scenarios 10 was enough.

1. Scenarios

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Scenario** | **Period of Time** | **Number of Cars** | **Number of charging station** | **Current SoC** | **Starting Time** | **Number of Charging Stations** | **Driver's Constraints** | **Renewable Energy Curve** |
| Scenario 1 | 3 hours | 10 | 2 station each 2 plugs | Varies between 21% - 63% | 11 | 2 | Each has a favorite charging station and an availability interval | Production energy curve |
| Scenario 2 | 5 hours | 20 | 2 station each 2 plugs | Varies 21% - 63% | 11 | 2 | Each has a favorite charging station and an availability interval | Production energy curve |
| Scenario 3 | 10 hours | 40 | 2 station each 2 plugs | Varies 21% - 63% | 8 | 2 | Each has a favorite charging station and an availability interval | Production energy curve |
| Scenario 4 | 10 hours | 80 | 4 station 2 plugs | Varies 21% - 63% | 8 | 4 | Each has a favorite charging station and an availability interval | Production energy curve |
| Scenario 5 | 3 hours | 10 | 2 station each 2 plugs | Varies (e.g., 10% - 50%) | 11 | 2 | Each has a favorite charging station and an availability interval | Consumption energy curve |
| Scenario 6 | 5 hours | 20 | 2 station each 2 plugs | Varies (e.g., 10% - 50%) | 11 | 2 | Each has a favorite charging station and an availability interval | Consumption energy curve |
| Scenario 7 | 10 hours | 40 | 2 station each 2 plugs | Varies (e.g., 10% - 50%) | 8 | 2 | Each has a favorite charging station and an availability interval | Consumption energy curve |
| Scenario 8 | 10 hours | 80 | 4 station 2 plugs | Varies (e.g., 10% - 50%) | 8 | 4 | Each has a favorite charging station and an availability interval | Consumption energy curve |

1. Evaluation Metrics Used
2. **Fitness value** that measures the quality of the solution found by the algorithm. Plots the change in the fitness score over iterations to visually analyze the convergence behavior. A converging algorithm will show a decreasing trend in the fitness score over time, with diminishing improvements as it progresses.

|  |  |  |
| --- | --- | --- |
| Scenario | Initial Fitness Score | Best Fitness Score |
| 1 | 108.8624 | 55.26235 |
| 2 | 131.716 | 65.916 |
| 3 | 285.712 | 133.112 |
| 4 | 283.468 | 61.668 |

Evolution of fitness score among iterations for each scenario:

A picture containing text, line, diagram, font

Description automatically generated

Figure 1 - Fitness Evolution in each scenario

1. **Rate of convergence** which measure of how quickly the metaheuristic find the optimal solution. It quantifies the average improvement per iteration or generation. The rate of convergence can be computed using the following steps:

convergenceRate =

where: [t, t+k] is the interval for each the convergence rate is computed, fitness(t) is the fitness value at the t iteration, fitness(t+k) is the fitness value at the t+k iteration, k is the number of iterations in the convergence interval for each the rate of converge is computed and | | represents the absolute value because in our problem the goal is to minimize the fitness score.

Convergence Rate indicates how much the fitness value changes, on average, over that interval. A higher rate of convergence indicates faster progress towards the optimal solution, while a lower rate suggests slower improvement.

We will consider that t is the first iteration and t+k is the last iteration for each scenario.

So, we will calculate the convergence rate until the solution is found.

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | T | K | Convergence Rate |
| 1 | 1 | 43 | 2.531 |
| 2 | 1 | 50 | 1.316 |
| 3 | 1 | 25 | 6.104 |
| 4 | 1 | 26 | 8.53 |

1. **Diversity** to assess the ability of the algorithm to explore different regions of the search space. To assess diversity, we have used Euclidean distance. This metric measures the dissimilarity or distance between individuals in the population.
2. **Run-time Efficiency** which evaluates the efficiency of the algorithm in term of time required to find a solution. It is measured using the algorithm's execution time.

|  |  |
| --- | --- |
| Scenario | Execution Time For Algorithm |
| 1 | 15ms |
| 2 | 34 ms |
| 3 | 78 ms |
| 4 | 167 ms |

1. **Constraint Violation** which measures the extent to which the individuals generated by the algorithm violate the problem's constraints. In our case where we have inequality constraints, we can determine the constraints violation by summing up the violations for each individual constraint.

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Favorite Charging Station Constraint Violation | Time Availability Constraint Violation | Total constraint Violation |
| 1 | 0 | 0 | 0 |
| 2 | 0 | 3 | 3 |
| 3 | 3 | 19 | 22 |
| 4 | 13 | 11 | 24 |

1. **Robustness**: Robustness measures the algorithm's ability to consistently find good solutions across different scenarios. It reflects the algorithm's stability and generalizability. Robustness can be evaluated by running the algorithm on multiple scenarios and analyzing the standard deviation of the fitness values obtained. The standard deviation values for different scenarios are put in a table and analyzed.
2. **Scalability**: Scalability refers to how well the algorithm performs as the problem size increases. This metric measures the algorithm's ability to handle larger problem instances without a significant decrease in performance. When evaluating scalability, we consider how the algorithm's execution time changes as the problem size or complexity grows. An algorithm is considered scalable if it can handle larger problem instances without a substantial increase in the time required. We generate a plot to show the execution times for various scenarios with different levels of complexity, and then perform an analysis of this plot.

A picture containing text, line, plot, font

Description automatically generated

Figure 2- Evolution of Execution Time For WOA

1. **Comparison with** **Baseline**: we compare to see how faithfully the EV scheduling curve reproduces the energy curve. To measure the similarity between the two curve we used

the Person coefficient:

A picture containing text, font, white, line

Description automatically generated

The resulting Pearson correlation coefficient will fall within the range of -1 to 1, indicating the strength and direction of the linear relationship between the two variables. A coefficient closer to 1 or -1 indicates a stronger linear relationship, while a coefficient closer to 0 suggests a weak or no linear relationship.

|  |  |
| --- | --- |
| Scenario | Comparison with Baseline (Pearson Coefficient) |
| 1 | -0.47684 |
| 2 | -0.803 |
| 3 | 0.427 |
| 4 | 0.944 |