22nd International Conference on Renewable Energies and Power Quality (ICREPQ'24)
Bilbao (Spain), 26th to 28th June 2024

ISBN: 978-84-09-60656-6

ACO-Driven Optimization: Analysis of Constraint Implementation in the Electric Vehicle Charge Optimization Problem

O. Abarrategi[1], G. Dobric[2], M. Zarkovic[2] and A. Iturregi[1]

[1] Department of Electrical Engineering
Bilbao School of Engineering, UPV/EHU
Rafael Moreno Pitxitxi 2, 48013 Bilbao (Spain)

[2] Department of Power Systems
University of Belgrade - School of Electrical Engineering Belgrade, Serbia

Abstract. This paper presents a novel Ant Colony Optimization (ACO) approach to optimize electric vehicle (EV) charging schedules, specifically focusing on minimizing tardiness. Addressing real-world constraints such as power limitations and load balancing, the proposed ACO algorithm effectively explores the solution space. Inspired by ant foraging behaviour, the method strategically utilizes pheromone trails to guide the optimization process. Validation with actual EV charging data underscores the algorithm's performance in minimizing tardiness. The study focuses on the significant impact constraints have on the optimization problem, shedding light on their role in shaping efficient charging schedules. It studies the difference outcomes of implementing the same constraints in different ways. This research contributes to the elaboration and development of new and efficient solutions that will help promote the adoption of electric vehicles.

Key words. Electric Vehicles, Ant Colony Optimization, Constraint Implementation, Tardiness Optimization.

1. Introduction

The widespread adoption of electric vehicles (EVs) represents a promising solution towards achieving sustainable transportation systems. However, the integration of EVs into the existing infrastructure poses significant challenges, particularly in managing their charging requirements efficiently. The Electric Vehicle Charge Optimization Problem emerges as a critical research area aimed at optimizing the charging schedules of a parquet of EVs while considering various constraints such as maximum power and power imbalance[1][2]. Thus, this problem can be address as a tardiness minimisation problem. The tardiness minimization problem is a type of scheduling problem where the objective is to minimize the total tardiness of jobs or tasks in a given schedule. Tardiness refers to the amount of time by which a job or task is completed later than its due date. In other words, it measures how late a job is finished beyond its specified deadline.

In [3], they propose the development of hybrid metaheuristics, inspired by the proven effectiveness of such combinations in addressing numerous scheduling problems. Specifically, they introduce a GRASP-like approach and a memetic algorithm, both tailored for utilization within the Variable Neighborhood Search framework. To solve the optimization problem, [4] adopt three metaheuristic algorithms, including particle swarm optimization (PSO), Salp swarm algorithm (SSA), and arithmetic optimization algorithm (AOA). [5] defined the objective functions for charging cost minimization, load variance minimization, and power loss minimization. The multi-objective problem was solved by the Whale Optimization Algorithm (WOA).

[6] uses Ant Colony Optimization (ACO), a metaheuristic that has gained prominence as a powerful metaheuristic approach for solving combinatorial optimization problems inspired by the foraging behaviour of ants. Its ability to explore complex solution spaces and adaptively search for optimal solutions makes it an attractive candidate for addressing the above mentioned problem. However, the successful application of ACO in solving the problem critically depends on the effective implementation of constraints within the optimization framework. ACO was used by [7] in an off-line electric vehicle (EV) scheduling problem for cloud-based parking operators, that a-priori accept parking reservations for EVs requesting charging services during their stay.

This paper presents a comprehensive analysis of constraint implementation in the context of the EV charging optimization using Ant Colony Optimization. Through rigorous experimentation and analysis, it provides an insight into the challenges and opportunities associated with the different ways of incorporating constraints into the optimization process and offer recommendations for enhancing the effectiveness.

2. Electric Vehicle Charge Optimization

This study addresses the scheduling challenge of charging a fleet of electric vehicles (EVs) at a station to minimize total tardiness, as expressed in (1). In scheduling problems like this one, tardiness represents the amount of time by which a job finishes after its due date. This is typically seen as a penalty or cost incurred due to late completion. If a job finishes early, it is not considered tardy. Therefore, tardiness is inherently non-negative, as can be seen in (1). Creating a feasible and efficient schedule proves challenging due to physical and power constraints at the charging station, including the maximum contracted power and power imbalance limits between electric feeder lines. Thus, the focus of this paper is to obtain the sequence in which EV's have to begin their charge, in order to minimize tardiness. When tardiness is zero, no vehicle is left partially charged.

$$\sum_{j=1}^{n} max\{0, \mathrm{CT}_j - \mathrm{d}_j\}$$
(1)

Where:
CT_j: charge completion time of vehicle j
d_j : due date of vehicle j

$$CT_j = s_j + p_j \quad (2)$$

s_j: assigned starting time for vehicle j
p_j: charging time for vehicle j

After defining the objective function, it becomes necessary to outline the constraints that will govern the optimization process.

The first constraint is the maximal power available for the charge of the vehicles. The layout of the electric car charging station features three charging lines, each equipped with multiple points that also serve as private parking spaces, allowing simultaneous car battery charging during parking. The power grid receives energy from a three-phase source with a 400 V voltage between phases. Each charging point (P_i) connects to a single-phase, supplying energy at 230 V and 5 kW. This setup allows a maximum number of vehicles to charge concurrently within a line, provided it stays within the contracted power limit, as expressed in (3).

The second constraint consists on maintaining a balanced consumption across the three lines, which is crucial to prevent grid imbalances, as described in equations (4) and (5). Grid imbalances could lead to higher energy losses and decreased transmission efficiency. Thus, in this respect, it is necessary to comply with Spanish regulations (BOE, 2013), as large imbalances without supplier consent may result in penalties for the customer [7].

$$\sum_{j=1}^{Pi} x_j^i \leq N, \mathrm{i} = \{1, \ldots, \mathrm{L}\} \quad (3)$$

$$\frac{\left|\sum_{j=1}^{Pi} x_j^i - \sum_{q=1}^{Pi} x_q^l\right|}{N} \leq \Delta, \mathrm{i}, \mathrm{l} = \{1, \ldots, \mathrm{L}\}, \mathrm{i} \neq \mathrm{l} \quad (4)$$

$$x_j^i = \begin{cases} 1, \text{if charging point j on line i is active;} \\ 0, \text{otherwise;} \end{cases}$$
(5)

Where,
x_j: is the state of the charging point j
t_j: the arrival time of vehicle j
N: number of active charging points
i: number of lines
j: number of vehicles

In summary, with "i" representing the number of lines, each line connects to "Pi" charging points, and the variable "N" stores the count of active charging points. The station's design harmonizes efficiency, regulatory compliance and user convenience.

3. Benchmark definition

Upon a car's arrival at the parking facility, crucial data needs to be gathered, including the arrival time, remaining charging time for reaching a full 100% charge (factoring in the 5 kW charging power and 50 kWh battery capacity), and the due date for the car to vacate the premises.

The Benchmark set used in this study was proposed by [1] and used by [6]. The three parameters shown above follow the following normal or uniform distributions as shown in the tables below (Table I, Table II, Table III). The first column of the tables shows the percentage of cars arriving at the car park and the second column shows the distribution that each group of cars follows. It can be seen that Table II does not directly show the remaining charging time to charge the battery of the cars arriving at the car park to 100% (N(C,D)). In order to calculate this data, the equations (5) and (6) have been used.

$$\boldsymbol{c} = \frac{(\mathbf{100}-\boldsymbol{A})*\mathbf{0.01}*\mathbf{60}*\boldsymbol{BatteriesCapacity}}{\boldsymbol{ChargingPower}}$$
(6)

$$\boldsymbol{D} = \frac{\boldsymbol{B}*\mathbf{0.01}*\mathbf{60}*\boldsymbol{BatteriesCapacity}}{\boldsymbol{ChargingPower}}$$
(7)

Where,
C: time it takes to charge the batteries
A: current battery charge level (%)
D: time it takes to discharge the batteries
B: current battery charge level (%)

Table I. – Vehicle Arrival Time

% VEHICLES	ARRIVAL TIME (MINUTES)
10	U (0,1440)
20	N (510,15)
10	N (720,15)
50	N (1170,15)
10	N (1350,15)

Table II. – Vehicle Initial Battery Charge Percentage

% VEHICLES	INITIAL CHARGE (%)
10	N (80,10) -> N(A,B)
30	N (50,15)
30	N (35,7.5)
30	N (12,6)

Table III. – Vehicle Due Date

% VEHICLES	DUE DATE (MINUTES)
10	N (240,120)
30	N (360,120)
30	N (480,120)
30	N (660,120)

4. Ant Colony System (ACS) Optimization

In order to solve the previously stated problem, this paper proposes ACS. Ant Colony System optimization is a metaheuristic algorithm inspired by the foraging behaviour of ants [8]. In this case, the algorithm utilizes a population of artificial ants that construct solutions to the given problem.

On the one hand, the State Transition Rule showed in (7) governs how artificial ants make decisions during the construction of solutions. If $q \leq q_0$, it favours exploitation, else it favours biased exploration.

$$\begin{cases} argmax\,[\tau_{ij}]^{\alpha} * [n_{ij}]^{\beta} & \text{if } q \leq q0 \\ P_{ij}^{k} = \frac{[\tau_{ij}]^{\alpha} * [n_{ij}]^{\beta}}{\sum_1^N [\tau_{ij}]^{\alpha} * [n_{ij}]^{\beta}} & \text{otherwise} \end{cases}$$

(8)

Where, τ_{ij} is the amount of pheromone on the edges and η_{ij} represents the fitness, a heuristic value, in this case the inverse of the due date, as shown in (8) and α and β are parameters controlling the relative importance of pheromone versus heuristic information respectively.

$$\eta = \frac{1}{Due\ Date} \tag{9}$$

On the other hand, the Global Pheromone Evaporation Rule establishes the velocity at which pheromone evaporates and accumulates, and it occurs after each ant completes a solution as expressed in equation (9).

It involves evaporating existing pheromones to simulate the natural decay process. The update is typically expressed as:

$$\tau_{ij} = (1 - \rho)\tau_{ij} + \rho * \Delta\,\tau\,\text{ij} \tag{10}$$

Where p is the pheromone evaporation rate ($\rho \in (0, 1]$) and $\Delta\,\tau\,\text{ij} = 1/C$, where C is the total tardiness for the best solution.

In summary, Ant System optimization employs a probabilistic state transition rule for ant movement and a global pheromone update equation to guide the search for optimal solutions while mimicking the foraging behaviour of real ants and therefore it is suited for the constrained optimisation problem described in section 2 that this paper aims to solve.

5. Constraint Implementation

Constraint implementation is of paramount importance in the efficiency with which solutions will be constructed in the search space of the ACS, for this particular problem of the EV, charging tardiness minimization. As it has been stated, constraints regarding this particular problem are defined in equations (3), (4) and (5). However, adapting ACS to handle constraints can be done following different strategies, and the choice depends on the specific requirements of the optimization problem.

This paper proposes two different methodologies to implement constraints in the above mentioned problem. The first approach is to implement a heuristic methodology that discard solutions that concur in constraint violations. During solution construction, it is essential to enforce adherence to constraints. Repair mechanisms can be applied if a solution violates constraints, or local search algorithms can be integrated for effective refinement.

The second approach is to extend the objective function with penalty terms, incorporating the cost of constraint violations. This ensures a balance between optimization goals and constraint satisfaction. Alternatively, the pheromone update mechanism can be refined to consider both the objective function value and the degree of constraint satisfaction. This adjustment influences pheromone levels based on the overall quality of solutions.

Considering all this, this paper focuses on comparing a heuristic constraint implementation with repair mechanisms vs. including penalties for constraint violation in the objective function calculation.

A. Heuristic implementation

After the virtual ants select a new component for the solution, it is verified whether it complies with constraints (3), (4) and (5) or not. If it does, it will be added to the final solution, if not it will be discarded and the next best selected in its place.

```
selectedComponent =
virtualAnts.selectNewComponent()

    if
isValidComponent(selectedComponent):

finalSolution.add(selectedComponent)
    else:

discardComponent(selectedComponent)

function isValidComponent(component):
    return satisfiesConstraints(component,
constraint1, constraint2)
```

B. *Penalty implemention*

In this case, the penalties will be included in the calculation of the fitness. This paper proposes equation (11) to do so. Hence, all penalties are accounted for and included in the equation. Each penalty counter is multiplied by a weighting factor. The weighting factors provide more parameters that are possible to alter to adapt fitness values and help improve the search process. Thus the resulting equation for fitness calculation would be:

$$\boldsymbol{\eta} = \frac{\mathbf{1}}{\boldsymbol{Due\ Date} + \boldsymbol{\lambda p}\mathbf{Cp} + \boldsymbol{\lambda b}\mathbf{Cb}} \tag{11}$$

Where,
C_p: power limit violations and
C_b: power imbalances and
λ_p: weighting factor for power limit violations
λ_b: weighting factor for power imbalances

6. Results

A. *Heuristic implementation*

In this case, the algorithm showed convergence problems from the beginning. Many of the trials resulted in tardiness values well over 1000min. The best results obtained are shown in Table IV and Fig.1.
Fig.1 shows that in the three cases the algorithm oscillates among the same values, regardless of the parameterization. The best results are shown for cases 1 and 3. This is because the first one has a bigger number of iterations and ants. Hence, it is able to explore the search space better. In case three, the value q_0 is lower and this favours exploration, so it is easier that the algorithm randomly comes to a better solution. But this not due to an efficient search process, and when executing again with the same q0 value, higher tardiness values were obtained. Thus, in this case too, this better tardiness value of case three can be considered an outlier.

The reason why the results are not good can be appreciated in Fig.2. The pheromone distribution through the search space is completely homogeneous. Note that the elements in the diagonal of the pheromone matrix have to be zero and are not part of the search space. However, all the rest of the elements in the search space are not accumulating much pheromone, as many solutions are being discard due to the hard constraints implemented. This has a negative effect on the solution construction, because exploitation is not helping conduct the search. Parameterization that strongly favours exploration can improve the obtained results, but the randomness in the results makes it an undesirable method

Table IV. – Parameterization and results for heuristic CP

It	Ant No	q_0	τ_0	A	β	ρ	N	Tardiness (min.)
200	5	0.6	0.5	1	2	0.3	30	652
50	3	0.75	0.5	2	4	0.8	30	833.8
50	4	0.4	0.5	1	5	0.8	30	440.5
50	3	0.4	0.5	1	5	0.3	30	900

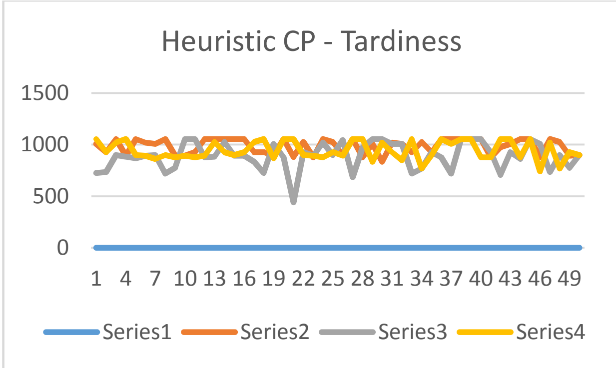

Fig. 1. Tardiness results for the Heuristic method

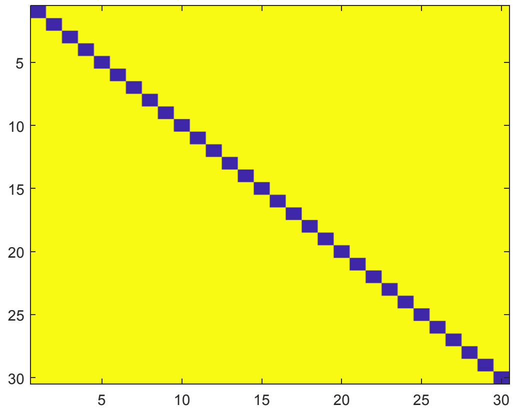

Fig.2. Pheromone distribution for the heuristic methodology

B. *Penalty implemention*

This case of constraint programming yields far better results. The algorithm generally converges, obtains solutions that do not violate constraints and if the right parameterization is chosen, it obtains very good solutions. However, as it is often the case with ACO, it tends to get stuck in suboptimal solutions. The best solution is obtained with q0 within the range [0.4,0.6], that favours the balance between an explorative and an exploitative search. The ratio β: α, shows a better construction when the fitness is favoured. However, a ratio 2:1 was more successful in obtaining the best result than the ratio 5:1. Therefore, it is crucial to maintain a careful balance between exploration and exploitation. The penalization constraint are equal and have high values in Table V and Fig.3. In Table V. and Fig.5 penalization constraints have lower values and a bigger weight is given to maximum power constraints violations. Equal and higher constraints have shown slightly better results. Fig.4. shows the pheromone distribution for the best solution. It can be seen that this heterogeneous distribution avoids stagnation and yields better solutions.

Table V. – Parameterization and results for penalty CP

It	Ant No	q0	τ0	α	β	ρ	N	λp	λb	Tar. (´)
50	5	0,6	0,5	1	2	0,3	30	500	500	335,7
50	3	0,4	0,5	1	5	0,8	30	500	500	362,34
50	3	0.75	0.5	1	5	0.8	30	500	500	517,2

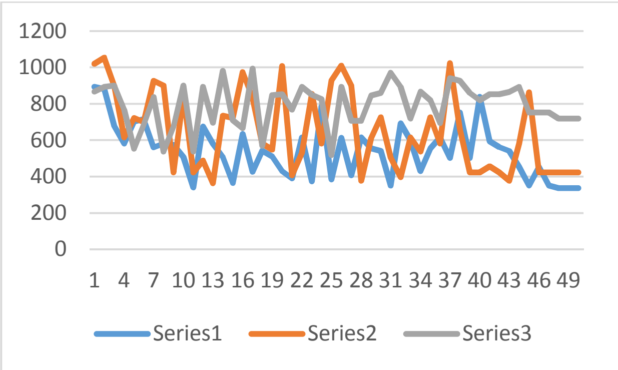

Fig.3. Tardiness results for the penalty method

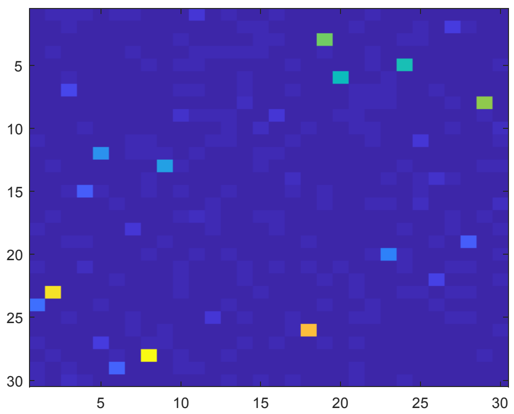

Fig.4. Pheromone distribution for the penalty methodology

Table VI. – Parameterization and results for penalty CP II

It	Ant No	q0	τ0	α	β	ρ	N	λp	λb	Tar. (´)
50	5	0.3	0,5	1	2	0,8	30	200	100	340.9
50	5	0.8	0,5	1	2	0,8	30	200	100	340.9
50	5	0.6	0.5	1	5	0.8	30	200	100	346
50	5	0.6	0.5	1	5	0.4	30	200	100	584.74

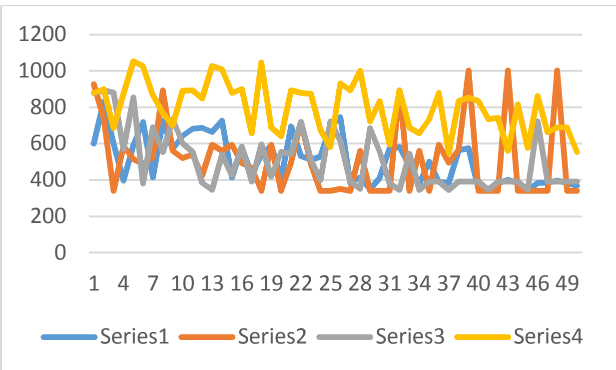

Fig.5. Tardiness results for the penalty method II

7. Conclusion

This paper focuses on how constraints may significantly affect the outcome of an optimisation algorithm, in this case an ACO algorithm used for tardiness minimisation in the EV charging problem.

Constraint programming is a paradigm for solving combinatorial problems that involve constraints over optimisation problems. It is particularly effective for problems with complex constraints, such as scheduling. This paper has proposed and implemented two different methodologies to program constraints

In the first case, a heuristic methodology has been included in order to exclude all solutions involving constraints violation. It has been seen, that in this case ACO struggles to find feasible solutions. In some cases, the algorithm was not able to converge. In all cases, the search space exploration was very limited and pheromone accumulated in the same paths from the beginning. Therefore, the results that were achieved were not good, with tardiness over 1000 minutes in many cases. However, all solutions were strictly complying with the established constraints.

In the second case, penalties were used for constraint violation. Different weighting factors were used for imbalance constraint violations and maximal power constraint violation. The first can be considered as a soft constraint, as it imbalances may sometimes occur in LV networks. However, the second one is a hard constraint, and that is why higher weighting factors were used. In this case, there were not convergence issues and the obtained results were significantly better than in the second. This is because exploration and exploitation were better balanced and the search was better directed.

Thus, we can conclude that implementing constraints through penalties in the fitness function calculation is more effective than taking a stricter or harder approach.

The right methodology for constraint programming along with a good parameterisation effort has a big influence in performance of ACO and should be considered carefully and tailored for the singularities of the optimisation problem at hand.

Acknowledgement

This work has been supported by the Basque Government (GISEL research group under grant number “OT1522-

22”) and has received funding from the European Union’s HORIZON-WIDERA-2021-ACCESS-03 under grant agreement No 101079200 - SUNRISE). Views and opinions expressed are however those of the authors only and do not necessarily reflect those of the Basque Government, European Union or European Research Executive Agency. Neither the European Union nor the granting authority can be held responsible for them

References

[1] A. Hernández-Arauzo, J. Puente, R. Varela, J. Sedano, “Electric vehicle charging under power and balance constraints as dynamic scheduling.” Computers & Industrial Engineering (2015) Vol. 85, pp. 306-315, https://doi.org/10.1016/j.cie.2015.04.002.

[2] S. Ge, J. Yan and H. Liu, "Ordered Charging Optimization of Electric Vehicles Based on Charging Load Spatial Transfer," 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019, pp. 1-6, doi: 10.1109/ICEMS.2019.8922218.

[3] J. García-Álvarez, M. A. González, C. R. Vela, “Metaheuristics for solving a real-world electric vehicle charging scheduling problem”, Applied Soft Computing (2018), Vol. 65, pp. 292-306, https://doi.org/10.1016/j.asoc.2018.01.010.

[4] P. Antarasee, S. Premrudeepreechacharn, A. Siritaratiwat, S. Khunkitti, “Optimal Design of Electric Vehicle Fast-Charging Station’s Structure Using Metaheuristic Algorithms”, Sustainability(2023),Vol15, https://doi.org/10.3390/su15010771

[5] K. Adetunji, I. Hofsajer, L. Cheng, "A Coordinated Charging Model for Electric Vehicles in a Smart Grid using Whale Optimization Algorithm" 2020 IEEE 23rd International Conference on Information Fusion (FUSION), Rustenburg, South Africa, 2020, pp. 1-7,
doi: 10.23919/FUSION45008.2020.9190284.

[6] M. Mavrovouniotis, G. Ellinas and M. Polycarpou, "Electric Vehicle Charging Scheduling Using Ant Colony System", IEEE Congress on Evolutionary Computation (CEC) (2019), pp. 2581-2588, https://doi: 10.1109/CEC.2019.8789989.

[7] T. Panayiotou, M. Mavrovouniotis, G. Ellinas, "On the Fair-Efficient Charging Scheduling of Electric Vehicles in Parking Structures", 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), Indianapolis, IN, USA, 2021, pp. 1627-1634,
doi: 10.1109/ITSC48978.2021.9565024.

[8] M. Dorigo, L.M. Gambardella, “Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem”, IEEE Transactions on Evolutionary Computation (1997), Vol. 1, No. 1, pp.53-66

[9] BOE (22 September 2013). Low Voltage Electrotechnical Regulation (TBR). Royal decree 842/2002, of 2 August 2002. Official Gazette of Spain (BOE). http://www.boe.es/