

Electric Vehicle Charging Scheduling Using Ant Colony System

Michalis Mavrovouniotis, Georgios Ellinas, Marios Polycarpou
KIOS Research and Innovation Center of Excellence,
Department of Electrical and Computer Engineering,
University of Cyprus.
email: {mavrovouniotis.michalis,gellinas,mpolycar}@ucy.ac.cy

Abstract—In this work we consider the scheduling problem for charging a fleet of electric vehicles (EVs) within a station such that the total tardiness of the problem is minimized. The generation of a feasible and efficient schedule is a difficult task due to the physical and power constraints of the charging station, i.e., the maximum contracted power and the maximum power imbalance between the lines of the electric feeder. The ant colony optimization (ACO) metaheuristic is applied to coordinate the charging process of the EVs within the charging station by generating efficient schedules. The behaviour and performance of ACO is analyzed and compared against state-of-the-art approaches on a benchmark set inspired by real-world scenarios. The experimental results show that the application of ACO is highly effective and outperforms other approaches.

Index Terms—Electric vehicles, scheduling, ant colony optimization

I. INTRODUCTION

Nowadays, there is a growing interest of using electric vehicles (EVs) instead of fuel-based vehicles due to their positive impact on the environment in reducing CO₂ emissions. However, their sustainable deployment requires new technologies (e.g., improving their battery capacity [1] and recharging batteries faster [2]), as well as new infrastructures to accommodate the charging of large fleets of EVs [3].

In this paper, we address a challenging scheduling problem, which is motivated by a charging station designed to be installed in a parking area where it can be used by a fleet of EVs both as a parking slot but also as a power source to recharge their batteries [4]. The aim of the problem is to coordinate the charging process of the EVs parked in the parking area by generating efficient schedules. In fact, several studies have shown that when the EVs' charging process is not properly coordinated in a charging station, several problems may occur, such as increase in the peak load period, decrease in service quality, degradation of the voltage profile, overload of circuits, and increase in energy losses [5], [6].

It is well known that EVs' bottleneck is their batteries that often require very high charging times and their capacity is relatively low for EVs to operate in large mileages. Therefore, deploying charging stations that can properly coordinate the charging process of EVs may contribute in addressing some of these issues. Apart from the EV issues regarding their battery technology, there are additional issues with the charging stations (e.g., they often have restrictions based on their

power and physical constraints). Particularly, the distributed power within the charging station must be balanced because of the economical and electro-technical reasons mentioned previously [7]. These constraints must be encompassed in the scheduling problem.

The aforementioned scheduling problem is \mathcal{NP} -hard [8], and, thus, the use of efficient scheduling algorithms is required to generate quickly schedules that maximize the utilization of the charging station resources while satisfying the charging demands of the EVs such that the total tardiness is minimized (in other words the total delay in charging EVs on time). A number of methodologies have been used to solve this problem including problem decomposition approaches [4], simple dispatching rules (e.g., first come first serve [4] and latest starting time [8]) and metaheuristics (e.g., artificial bee colony [9] and genetic algorithm [10]). A comprehensive review of optimization techniques on relevant scheduling problems can be found in [11]. Among the approaches, metaheuristics have showed the best performance due to their ability to efficiently provide the optimal or near to the optimal solutions in real-world sized instances.

In this paper, we design and investigate the application of another metaheuristic, that is, ant colony optimization (ACO) [12], that has shown good performance in different scheduling problems [13]–[15] but also in similar EV charging scheduling problems [16]. Hence, it would be interesting to investigate the application of ACO in the aforementioned scheduling problem, which is much more challenging and realistic, and compare it with existing applications. The rest of the paper is organized as follows. Section II reviews existing charging scheduling models. Section III describes in detail the scheduling problem of the EV charging station considered in this work. Section IV describes the ACO application for the scheduling problem. The experimental results and analysis are presented in Section V. Finally, Section VI concludes this paper.

II. BACKGROUND

The coordination of EV charging within a station raises challenging scheduling problems. In the last few years, several charging scheduling models have been proposed (a comprehensive review of such charging approaches can be found in [17]). In general, two main architectures are used to coordinate EV charging in existing scheduling models: 1) decentralized

control architectures, and 2) centralized control architectures. The former type of architectures offers great flexibility to EV owners, allowing them to decide the period to start charging their EVs. In the latter type of architectures, all the decisions are taken by the central control system of the charging station using information provided by the EV owners. Although the centralized architecture is not very flexible, it is more secure than the decentralized architecture. This is because with the centralized architecture the overall system can be controlled in such a way to prevent overloading of the power grid and to minimize power losses.

Depending on the characteristics and goals of each EV charging station different models and scheduling problems arise. For example, a number of objective functions have been considered in existing scheduling models such as minimizing the total tardiness [9], [10], total cost [18], [19], energy losses [4], [8], grid congestion [20] or optimizing more than one objectives simultaneously [21]. Also some models consider variable charging rates of the EVs [20], variable charging power of the station [22], or variable electricity prices [23]. Other models, allow the control system of the charging station to decide the assignment of EVs to the charging (or parking) slots [6] whereas in other models the assignment is agreed in advanced (e.g., each EV owner reserves a specific slot in the charging station) [8].

III. ELECTRIC VEHICLE CHARGING SCHEDULING PROBLEM

A. EV Charging Station

Suppose that a charging station utilizing a centralized control architecture was designed to be installed in a private parking in which EVs have their own space to use as a parking slot and at the same time to charge their batteries [4]. A general structure of a charging station with a three-phase electric power source is shown in Fig. 1.

When EVs enter the charging station the control system of the station must create a feasible and efficient schedule for the charging process of the EVs, so that the total tardiness is minimized. There are two main constraints that the control system of the charging station must take into account. First, there is limited contracted power, and so there is a maximum number of charging points that can be active at the same time in any given line. Second, the power consumed by the lines must be balanced at any given time to avoid energy losses [7].

For example, in Fig. 1 the charging station uses a three-phase electric power source, hence three lines connect the charging points. If we assume that each line is limited to a maximum four active charging points at the same time, then all the EVs that are currently parked in the station can start charging, while at the same time, the power distribution in the lines is balanced assuming that the power is transferred at a constant rate.

B. Problem Formulation

The charging station consists of n charging points (or parking slots) that are connected by L lines. Each line i

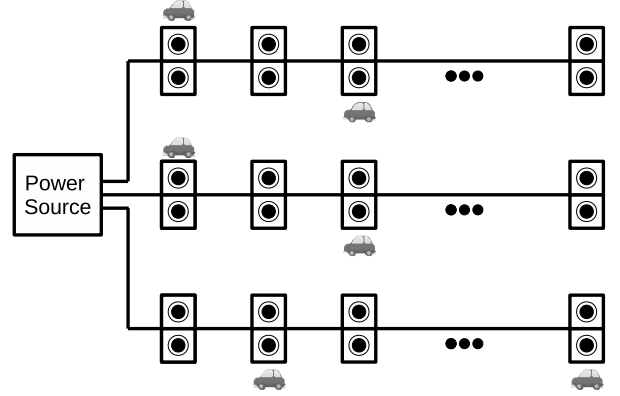


Fig. 1. A general structure of a charging station with three lines and multiple charging points (or parking slots).

connects P^i charging points where each point is also the private parking of an EV. Even though the charging station has n charging points available, they cannot be all active at the same time. In particular, the maximum number of active charging points for each line must satisfy the following constraints:

$$\sum_{j=1}^{P^i} x_j^i \leq N, \quad i = \{1, \dots, L\}, \quad (1)$$

$$\frac{|\sum_{j=1}^{P^i} x_j^i - \sum_{q=1}^{P^l} x_q^l|}{N} \leq \Delta, \quad i, l = \{1, \dots, L\}, i \neq l, \quad (2)$$

$$x_j^i = \begin{cases} 1, & \text{if charging point } j \text{ on line } i \text{ is active;} \\ 0, & \text{otherwise;} \end{cases} \quad (3)$$

where Eq. (1) ensures that each line can only have N charging points active to charge N EVs at the same time, Eq. (2) controls the maximum imbalance Δ ($\Delta \in [0, 1]$) between the lines, and Eq. (3) defines a decision binary variable.

For each EV j , using the charging station, there is an arrival time t_j ($t_j \geq 0$), a charging time p_j ($p_j > 0$), and a due date d_j ($d_j \geq t_j + p_j$) denoting the departure time of the EV. The goal is to generate a sequence of EVs to coordinate their charging, satisfying the two constraints in Eq. (1) and Eq. (2), and minimizing the total tardiness defined mathematically as follows:

$$\sum_{j=1}^n \max\{0, CT_j - d_j\}, \quad (4)$$

where CT_j is the completion time of EV j which is calculated as $CT_j = s_j + p_j$ where s_j ($s_j \geq t_j$) is the assigned starting time for the charging of EV j .

It must be noted that no preemption is allowed (i.e., an EV cannot be disconnected before the completion time CT_j is reached). Other assumptions imposed to simplify the model include: all EVs charging at the same constant rate and the contracted power in the charging station to be constant over time. Also we assume that the profiles of the EVs (i.e., t_j , p_j and d_j) are known in advance, since a static model is considered.

IV. ANT COLONY OPTIMIZATION

In the ACO metaheuristic a colony of ω (artificial) ants iteratively constructs solutions for the problem under consideration using (artificial) pheromone trails which are associated with appropriately defined solution components and heuristic information [24]. A skeleton of the ACO metaheuristic is presented in Algorithm 1. Ants modify the pheromone trails during the algorithm's execution based on the quality of the constructed solution. In this paper, we apply Ant Colony System (ACS) [12], one of the best performing ACO algorithms in minimizing the total tardiness for several scheduling problems [25]–[27], to the scheduling problem of the charging station described above.

A. Constructing Solutions

In ACS, each ant starts with an empty set of *scheduled* EVs and then incrementally appends *unscheduled* EVs to the partial set of the scheduled EVs so far, until all EVs are scheduled. A constructed solution is basically a sequence of EVs (i.e., a permutation of EV indices). It must be noted that a charging schedule is generated based on the order in which the EVs are placed in the permutation in the same way as the scheduler algorithm proposed in [10]. Specifically, to sequentially schedule all EVs in the permutation, assigning for each EV the earliest possible starting time, such that all constraints are satisfied with respect to the previously scheduled EVs.

Ant k selects the unscheduled EV j to be added in the permutation at position i according to the following decision rule:

$$j = \begin{cases} \arg \max_{l \notin S^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta, & \text{if } q \leq q_0, \\ \mathcal{J}, & \text{otherwise,} \end{cases} \quad (5)$$

where τ_{il} is the pheromone trail associated to the assignment of EV l in position i and η_{il} is the heuristic information of assigning EV l in position i , S^k is the partial set of scheduled EVs, α is a parameter that controls the influence of the pheromone trails, β is a parameter that controls the influence of the heuristic information, q is a random variable uniformly distributed in $[0, 1]$, q_0 ($q_0 \in [0, 1]$) is a parameter that controls the exploration of the decision rule, and \mathcal{J} is a random variable (representing an EV index) selected according to the following probability distribution:

$$p_{ij}^k = \frac{[\tau_{ij}]^\alpha [\eta_{ij}]^\beta}{\sum_{l \notin S^k} [\tau_{il}]^\alpha [\eta_{il}]^\beta}, \quad \text{if } j \notin S^k, \quad (6)$$

In other words, with probability q_0 the ant makes the best decision as indicated by the pheromone trails and the heuristic information (exploitation), while with probability $(1 - q_0)$ the ant makes a random decision biased by the pheromone trails and the heuristic information (exploration).

The heuristic information η_{ij} of assigning EV j in position i in the schedule is computed by the *earliest due date* rule [28]. This dispatching rule sorts EVs in increasing order of their due dates d_j (i.e., the requested departure times of the EVs).

Algorithm 1 ACO Metaheuristic Outline

- 1: Set parameters
 - 2: Initialize pheromone trails
 - 3: Initialize heuristic information
 - 4: **while** (termination condition *not* satisfied) **do**
 - 5: *ConstructSolutions*
 - 6: *PheromoneUpdate*
 - 7: **end while**
 - 8: **OUTPUT**: the best-so-far solution
-

Hence, in our case $\eta_{ij} = 1/d_j$. In other words, the heuristic information will favour the EVs that need to leave earlier than other EVs. In the experiments in Section V the impact of using this heuristic information is further investigated.

B. Updating Pheromones

In ACS, two types of pheromone updates are applied, one at the global level and the other at the local level. In the global pheromone trail update after each algorithmic iteration only the best-so-far ant¹ is allowed to add pheromone to its solution components. In particular, the pheromone trails associated with the solution components represented by the best-so-far ant are updated as follows:

$$\tau_{ij} \leftarrow (1 - \rho)\tau_{ij} + \rho\Delta\tau_{ij}^{bs}, \quad (7)$$

where ρ ($\rho \in (0, 1]$) is the pheromone evaporation rate and $\Delta\tau_{ij}^{bs} = 1/C^{bs}$, where C^{bs} is the total tardiness value of the best-so-far solution. It must be noted that ACS terminates whenever the total tardiness of the best-so-far solution becomes zero (which is the minimum value according to Eq. (4)). Therefore, a division by zero is never allowed.

In the local pheromone trail update, ACS applies a step-by-step pheromone update rule immediately after an ant has added a new solution component (i.e., an EV j in position i). In particular, the pheromone trails associated with the aforementioned solution components are updated as follows:

$$\tau_{ij} \leftarrow (1 - \xi)\tau_{ij} + \xi\tau_0, \quad (8)$$

where ξ ($\xi \in (0, 1]$) is a parameter that controls the influence of the local pheromone update and τ_0 is the initial pheromone value. Note that a good value for τ_0 was found to be $1/rC^{edd}$, where r is the total number of EVs and C^{edd} is the total tardiness value of the solution generated by the earliest due date rule [26]. All pheromone trails are uniformly initialized at the start of the execution. The effect of the local updating rule is to make the decision of assigning EV j in position i less desirable for the other ants to favour assignments of other EVs on that position. In this way, the exploration of the search is promoted. In the experiments in Section V these two pheromone update types (i.e., local and global) are further investigated.

¹A special ant that represents the global best solution and may not necessarily belong in the current constructing colony.

V. EXPERIMENTAL STUDIES

A. Experimental Setup

In the experimental studies, we consider the real-world benchmark set proposed in [8], representing a charging station of three lines (as in Fig. 1), each one connecting 60 charging points. In total the charging station consists of 180 charging points (also parking slots). Specifically, the L and n parameters in Section III are defined as follows: $L = 3$ and $n = 180$. The data of the benchmark set consists of the profiles of 180 EVs (and hence the r parameter is set to $r = 180$), including their arrival times, due dates and demands of the EVs (based on the behaviour of real users). The benchmark instances are accessible at <http://www.di.uniovi.es/iscop>.

Two types of instances exist in the benchmark set (30 different instances for each type) that differ in the distribution of the EVs on the lines. In Type 1 instances 60 EVs arrive at each line during the day (in a 24-hour period) demanding charging, while in Type 2 instances 108 EVs arrive in line 1, 54 in line 2 and 18 in line 3. Different values of the maximum number parameter for EVs that can charge at the same time in a line, i.e., $N \in \{20, 30, 40\}$ defined in Eq. (1), and different values of the maximum power imbalance parameter, i.e., $\Delta \in \{0.2, 0.4, 0.6, 0.8\}$ defined in Eq. (2), are considered for each problem instance, resulting in 720 different test cases (12 different test cases for each instance).

ACO performs 30 independent runs for each test case because of its stochastic nature and the averaged total tardiness (in hours) is recorded. The colony size of ACO was set to $\omega = 200$ and the remaining parameters are further investigated in the next section.

B. ACO Parameter Settings

In order to achieve a good balance between the exploration and exploitation for ACO on this scheduling problem, several parameters, e.g., α , β , ρ , ξ and q_0 , require further investigation. The parameter tuning was performed on 24 test cases (i.e., the first instance of the benchmark set for each different test case in Table I).

1) *Effect of Pheromone trails:* The value of the α parameter is set either to 0, indicating no pheromone bias, and to 1, indicating pheromone bias. The experimental results regarding the total tardiness with different α values are given in Fig. 2. It can be observed that when the solutions are constructed without any guidance from the pheromone trails (i.e., when $\alpha = 0$) the total tardiness increases. This is because the solutions are constructed without utilizing the experience learned so far, which is stored in the pheromone trails. The pheromone trails have the effect of increasing the probabilities of the promising positions in the schedule for an EV (which is the solution component associated with a pheromone value).

2) *Effect of Heuristic Information:* The value of the β parameter is varied as follows: $\beta \in \{0, 2, 5, 10\}$. The experimental results regarding the total tardiness with different β values are given in Fig. 3. It can be observed that when heuristic information is utilized (i.e., when $\beta > 0$) the total

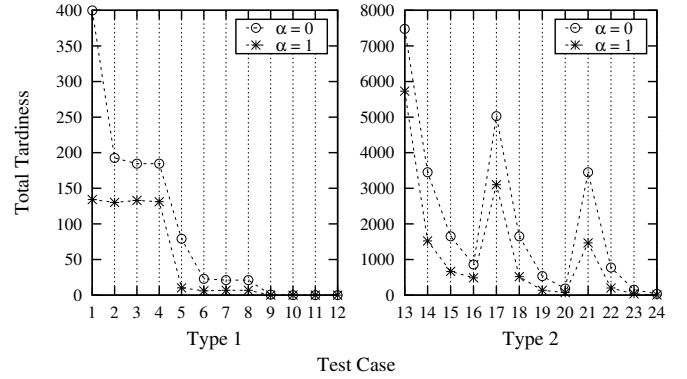


Fig. 2. Total tardiness (averaged over 30 runs) of ACO with different α values on the first problem instance of the benchmark set for Type 1 (left) and Type 2 (right) test cases.

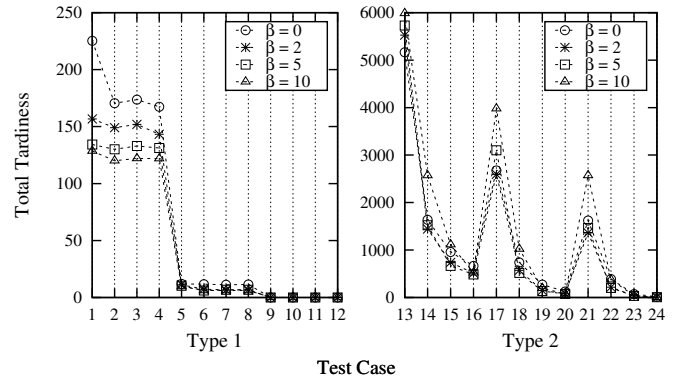


Fig. 3. Total tardiness (averaged over 30 runs) of ACO with different β values on the first problem instance of the benchmark set for Type 1 (left) and Type 2 (right) test cases.

tardiness improves in almost all cases (except in test case 13). This shows that the heuristic information (based on the earliest due date of the EV) is beneficial in the construction of solutions because it favours EVs that must leave earlier than other EVs. In many Type 2 test cases, that promote the imbalance between the lines, when $\beta = 10$ the total tardiness is worse than when $\beta = 0$. This is because of the power and imbalance constraints in Eq. (1) and Eq. (2) that affect the validity of the information provided to the search by the earliest due date rule. On the contrary, in Type 1 test cases these constraints are less likely to be violated, hence, a stronger influence by the heuristic information (e.g., when $\beta = 10$) can guide the search in the promising areas of the search space. It must be noted that selecting higher values of β (not presented here) did not have significant improvements in Type 1 test cases, but significantly increased the total tardiness in most Type 2 test cases.

3) *Effect of Decision Rule:* The value of the q_0 parameter is varied as follows: $q_0 \in \{0.0, 0.5, 0.9, 0.95, 1.0\}$. The experimental results regarding the total tardiness with different q_0 values are given in Fig. 4. It can be observed that as the q_0 value increases the total tardiness improves. This is because

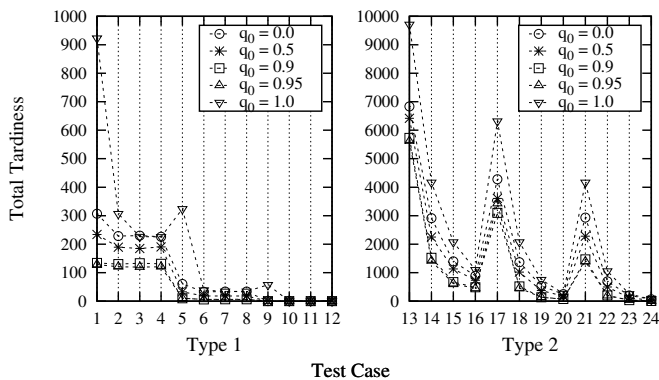


Fig. 4. Total tardiness (averaged over 30 runs) of ACO with different q_0 values on the first problem instance of the benchmark set for Type 1 (left) and Type 2 (right) test cases.

the pheromone trail policy described in Section IV promotes exploration. For example, with the local pheromone update rule it is less likely for the ants to construct similar solutions. Therefore, a decision rule that further promotes exploration (i.e., when $q_0 = 0.0$) will not be beneficial. In particular, it can be observed that when $q_0 = 0.9$ or $q_0 = 0.95$ the total tardiness is better than when $q_0 = 0.5$ and $q_0 = 0.0$ in most test cases. However, an extreme value of q_0 (i.e., $q_0 = 1.0$) is not beneficial because it leads the search to no exploration at all.

4) *Effect of Pheromone Evaporation:* The value of the ρ parameter is varied as follows: $\rho \in \{0.0, 0.1, 0.2, 0.4, 0.6\}$. The experimental results regarding the total tardiness with different ρ values are given in Fig. 5. It is interesting to observe that when no pheromone evaporation is used (i.e., when $\rho = 0.0$) the total tardiness increases. This is because the pheromone trails representing the best-so-far solution are never reduced. Hence, there is a potential risk of continuously increasing these trails and getting trapped in a (possibly poor) local optimum from the initial stages of the search, resulting in premature convergence. This can be easily observed in Fig. 5, when $\rho > 0.0$ the total tardiness significantly improves. On the contrary, the total tardiness does not show any significant improvement when the evaporation rate is higher than 0.2. It must be noted that higher values of ρ (i.e., $\rho > 0.6$) have been investigated with no significant improvements, and, thus, are not included in Fig. 5.

5) *Effect of Local Pheromone Update:* The value of the ξ parameter is varied as follows: $\xi \in \{0.0, 0.1, 0.2, 0.4\}$. The experimental results regarding the total tardiness with different ξ values are given in Fig. 6. It can be observed that when local pheromone update is not used (i.e., when $\xi = 0.0$) the total tardiness is the worst especially in Type 2 test cases in which the imbalance constraint is promoted. For this type of test cases, it is easier for ACO to make several undesirable assignments because of the imbalance constraint. Hence, using the local pheromone update these assignments will be quickly ignored as their pheromone trails will be reduced, while the

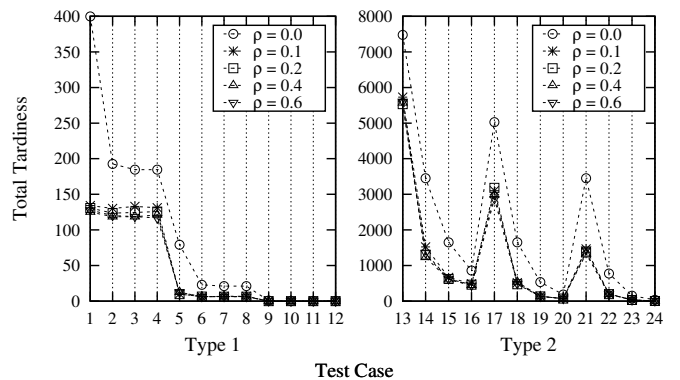


Fig. 5. Total tardiness (averaged over 30 runs) of ACO with different ρ values on the first problem instance of the benchmark set for Type 1 (left) and Type 2 (right) test cases.

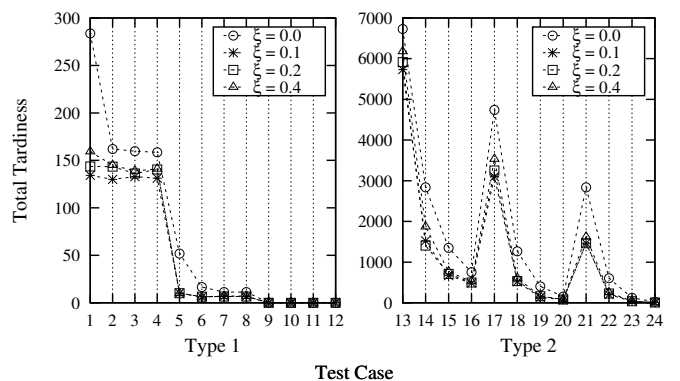


Fig. 6. Total tardiness (averaged over 30 runs) of ACO with different ξ values on the first problem instance of the benchmark set for Type 1 (left) and Type 2 (right) test cases.

ants are selecting components during the construction of their solutions. It must be noted that higher values of ξ (i.e., $\xi > 0.4$) have been investigated with no significant improvements, and, thus, are not included in Fig. 6.

C. Comparison with Other Approaches

In this section we compare the proposed ACO (with its parameters set to $\alpha = 1$, $\beta = 5$, $\rho = 0.4$, $\xi = 0.1$ and $q_0 = 0.95$)² with a problem decomposition (PD) approach [4], three dispatching rules: 1) first come first serve (FCFS) [4], 2) latest starting time (LST) [8], and 3) earliest due time (EDT) [28], and a state-of-the-art genetic algorithm (GA) [10], which is a metaheuristic method, on the full benchmark set. FCFS, EDT, and LST approaches sort in ascending order the EVs based on their arrival time t_j , their due date d_j and the rule given by $(d_j - p_j)$, respectively. Then, the schedule will be generated according to the ordering given by the corresponding dispatching rule. The PD approach decomposes the problem and aims to generate a schedule for each line separately. On

²Note that different parameter settings may result in much better performance in individual test cases. However, these parameter settings were found to achieve reasonable performance over most test cases.

TABLE I
COMPARISON OF ACO WITH OTHER APPROACHES. EACH VALUE IS THE SUM OF THE (AVERAGED OVER 30 RUNS FOR GA AND ACO) TOTAL TARDINESS OF THE 30 INSTANCES OF EACH TEST CASE. THE VALUES ARE GIVEN IN HOURS. THE BEST VALUES FOR EACH TEST CASE ARE INDICATED IN BOLD.

Test Case				FCFS [4]	LST [8]	EDT [28]	PD [4]	GA [10]	ACO
Number	Type	N	Δ						
1	1	20	0.2	21807	19326	21667	7720	5442	5391
2			0.4	8909	6801	7410	4240	2680	2592
3			0.6	7104	5476	5563	3849	2300	2261
4			0.8	7012	5332	5406	3807	2239	2214
5		30	0.2	8892	5822	7571	1782	997	906
6			0.4	1538	454	799	490	92	82
7			0.6	1045	168	494	458	50	44
8			0.8	1004	158	487	458	49	41
9		40	0.2	3587	1928	2847	646	364	264
10			0.4	82	26	51	28	0	0
11			0.6	11	0	1	9	0	0
12			0.8	11	0	1	9	0	0
13	2	20	0.2	228816	226647	232109	127614	124380	123882
14			0.4	102396	100355	96408	46254	45263	44852
15			0.6	49899	48867	45988	23008	21206	20914
16			0.8	27581	27421	25703	14808	13031	12856
17		30	0.2	149857	148926	146673	72460	71129	70168
18			0.4	49941	48844	45872	21427	20630	20324
19			0.6	19340	18385	17405	8079	7188	7108
20			0.8	9518	8201	7688	4501	3607	3577
21		40	0.2	102396	100355	96408	46096	45216	44864
22			0.4	25814	25019	23819	10932	10011	9822
23			0.6	9078	7711	7127	3520	2917	2873
24			0.8	3765	2286	2365	1659	923	882

the contrary, the GA generates a schedule for the entire system using search operators specifically designed for the scheduling problem described in Section III. Similar to the proposed ACO approach, the GA approach performs 30 independent runs for each instance and the average total tardiness (in hours) is recorded. The two metaheuristic approaches (i.e., ACO and GA) use the same number of individuals³ (i.e., 200) and the same termination conditions (i.e., when the best solution is not improved for 25 consecutive iterations, or if zero total tardiness is reached). It must be noted that the parameters of the GA are also optimized in [10] under the same test cases as ACO was optimized in Section V-B and they are set as follows: crossover probability 0.8, mutation probability 0.1 and the size of the tournament selection was set to 8. Table I shows the results of the experiments. The values reported are the sum of the total tardiness of the 30 instances of each test case of the algorithms.

³This is a general term that is used to represent “ants” in ACO and “chromosomes” in GA.

From Table I, it can be observed that all algorithms obtain a significantly lower total tardiness in Type 1 test cases (i.e., number 1–12) than Type 2 test cases (i.e., number 13–24). As described above, in Type 1 test cases, EVs are uniformly distributed whereas in Type 2 test cases they are not. Consequently, this feature imposes more challenges to the algorithms when solving Type 2 test cases because they have to control in many situations the power imbalance among the lines when building the schedule. Also, it can be observed that as the N and Δ parameters increase the total tardiness of all algorithms decreases. In fact, for several Type 1 test cases the total tardiness is approaching 0, which is the minimum value according to the objective function in Eq. (4). This is because the values of the constraints in Eq. (1) and Eq. (2) are large enough, hence, they are less likely to be violated causing an increase in the total tardiness. In other words, the more times these constraints are violated the higher the total tardiness.

Furthermore, from Table I it can be observed that FCFS, LST, and EDT are not performing as good as GA and ACO. This is natural because no optimization is performed within

the three aforementioned dispatching rules. Also, it can be observed that both GA and ACO significantly outperform PD in all test cases as it was expected. This is because PD evaluates schedules for each line separately, and, thus, may overlook some situations of addressing the power imbalance without necessarily violating the constraint. As a result, the charging of the EVs will be unnecessarily delayed because the imbalance constraint is violated causing an increase in the total tardiness. On the other hand, this situation is less likely to occur in ACO and GA. These results validate that metaheuristics are suitable for the scheduling problem arising in the EV charging station.

Also, it is interesting to observe that ACO outperforms the GA approach in all test cases. The advantage of ACO against GA lies on its structural difference. In particular, ACO can be seen as an iterative constructive heuristic that generates new solutions in every iteration on the basis of learned data (i.e., the pheromone trails). Therefore, the risk of beginning with poor initial solutions (as with GA) or getting trapped in a (possibly poor) local solution later on is limited with ACO.

VI. CONCLUSIONS

EVs and their technologies have received increased interest due to their positive impact on the environment. In this work, we consider a scheduling problem arising in a charging station to address the coordination of the charging process for a fleet of EVs. The generated schedules need to satisfy the physical and power constraints of the charging station. These constraints make the scheduling problem harder to solve using simple dispatching rules (e.g., first come first serve [4] or latest starting time [8]). In this paper, we apply ACO to generate feasible schedules for EVs to minimize the total tardiness of the scheduling problem. Experimental results using a benchmark set with various real-world inspired scenarios showed that the ACO approach is more suitable for the scheduling problem, compared to simple dispatching rules and other approaches, mainly because of its intrinsic characteristics.

For future work, we plan to extend the model considered in this work with additional realistic constraints (e.g., variable charging rates) and integrate a local search operator to further improve the performance of ACO. In fact, the performance of other metaheuristics has been significantly enhanced with a local search in [29], [30] on the described scheduling problem. Recall that in Table I the metaheuristic approaches are not utilizing any local search operator. Therefore, for future work it would be interesting to compare the performance of the existing metaheuristics with local search against the proposed ACO with local search. Also, another interesting direction is to solve the problem dynamically, assuming that the information of the EVs is not known a priori as in [4], [10].

ACKNOWLEDGMENT

This work has been supported by the European Union's Horizon 2020 research and innovation programme under grant agreement No. 739551 (KIOS CoE) and from the Government

of the Republic of Cyprus through the Directorate General for European Programmes, Coordination and Development.

REFERENCES

- [1] D. Kosmanos, L. Maglaras, M. Mavrouniotis, S. Moschogiannis, A. Argyriou, A. Maglaras, and H. Janicke, "Route optimization of electric vehicles based on dynamic wireless charging," *IEEE Access*, vol. 6, pp. 42551–42565, 2018.
- [2] A. Felipe, M. T. Ortuño, G. Righini, and G. Tirado, "A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges," *Transportation Research Part E: Logistics and Transportation Review*, vol. 71, pp. 111–128, 2014.
- [3] J. García-Villalobos, I. Zamora, J. S. Martn, F. Asensio, and V. Aperribay, "Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 717–731, 2014.
- [4] A. Hernandez-Arauzo, J. Puente, and M. A. Gonzalez, "Dynamic scheduling of electric vehicle charging under limited power and phase balance constraints," in *Proceedings of the 7th Scheduling and Planning Applications Workshop (SPARK 2013)*, 2013, pp. 1–8.
- [5] K. Clement-Nyns, E. Haesen, and J. Driesen, "The impact of charging plug-in hybrid electric vehicles on a residential distribution grid," *IEEE Transactions on Power Systems*, vol. 25, no. 1, pp. 371–380, Feb 2010.
- [6] N. B. Arias, J. F. Franco, M. Lavorato, and R. Romero, "Metaheuristic optimization algorithms for the optimal coordination of plug-in electric vehicle charging in distribution systems with distributed generation," *Electric Power Systems Research*, vol. 142, pp. 351–361, 2017.
- [7] J. A. P. Lopes, F. J. Soares, P. M. Almeida, and M. M. D. Silva, "Smart charging strategies for electric vehicles: Enhancing grid performance and maximizing the use of variable renewable energy resources," in *In Proceedings of the EVS24 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium*, 2009, pp. 392–396.
- [8] A. Hernández-Arauzo, J. Puente, R. Varela, and J. Sedano, "Electric vehicle charging under power and balance constraints as dynamic scheduling," *Computers & Industrial Engineering*, vol. 85, pp. 306–315, 2015.
- [9] J. García-Álvarez, M. A. González, C. R. Vela, and R. Varela, "Electric vehicle charging scheduling using an artificial bee colony algorithm," in *Natural and Artificial Computation for Biomedicine and Neuroscience*, J. M. Ferrández Vicente, J. R. Álvarez-Sánchez, F. de la Paz López, J. Toledo Moreo, and H. Adeli, Eds. Cham: Springer International Publishing, 2017, pp. 115–124.
- [10] J. García-Álvarez, M. A. González, and C. R. Vela, "A genetic algorithm for scheduling electric vehicle charging," in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '15. New York, NY, USA: ACM, 2015, pp. 393–400.
- [11] I. Rahman, P. M. Vasant, B. S. M. Singh, M. Abdullah-Al-Wadud, and N. Adnan, "Review of recent trends in optimization techniques for plug-in hybrid, and electric vehicle charging infrastructures," *Renewable and Sustainable Energy Reviews*, vol. 58, pp. 1039–1047, 2016.
- [12] M. Dorigo and L. M. Gambardella, "Ant colony system: A cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.
- [13] A. Colnani, M. Dorigo, M. Maniezzo, and M. Trubian, "Ant system for job-shop scheduling," *JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 34, no. 1, pp. 39–53, 1994.
- [14] T. Stützle, "An ant approach to the flow shop problem," in *In Proceedings of the 6th European Congress on Intelligent Techniques & Soft Computing (EUFIT'98)*. Verlag, 1997, pp. 1560–1564.
- [15] M. Dorigo, G. D. Caro, and L. M. Gambardella, "Ant algorithms for discrete optimization," *Artificial Life*, vol. 5, no. 2, pp. 137–172, 1999.
- [16] S. Xu, D. Feng, Z. Yan, L. Zhang, N. Li, L. Jing, and J. Wang, "Ant-based swarm algorithm for charging coordination of electric vehicles," *International Journal of Distributed Sensor Networks*, vol. 9, no. 5, pp. 1–13, 2013.
- [17] J. Garca-Villalobos, I. Zamora, J. S. Martn, F. Asensio, and V. Aperribay, "Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 717–731, 2014.
- [18] Y. He, B. Venkatesh, and L. Guan, "Optimal scheduling for charging and discharging of electric vehicles," *IEEE Transactions on Smart Grid*, vol. 3, no. 3, pp. 1095–1105, 2012.

- [19] H. Wu, G. K.-H. Pang, K. L. Choy, and H. Y. Lam, "Dynamic resource allocation for parking lot electric vehicle recharging using heuristic fuzzy particle swarm optimization algorithm," *Applied Soft Computing*, vol. 71, pp. 538–552, 2018.
- [20] L. Gan, U. Topcu, and S. H. Low, "Optimal decentralized protocol for electric vehicle charging," *IEEE Transactions on Power Systems*, vol. 28, no. 2, pp. 940–951, 2013.
- [21] A. Janjic, L. Velimirovic, M. Stankovic, and A. Petrusic, "Commercial electric vehicle fleet scheduling for secondary frequency control," *Electric Power Systems Research*, vol. 147, pp. 31–41, 2017.
- [22] J. Han, J. Park, and K. Lee, "Optimal scheduling for electric vehicle charging under variable maximum charging power," *Energies*, vol. 10, no. 7, 2017.
- [23] W. Su and M. Chow, "Performance evaluation of an eda-based large-scale plug-in hybrid electric vehicle charging algorithm," *IEEE Transactions on Smart Grid*, vol. 3, no. 1, pp. 308–315, 2012.
- [24] M. Dorigo and T. Stützle, *Ant colony optimization*. Cambridge, MA: MIT Press, 2004.
- [25] A. Bauer, B. Bullnheimer, R. F. Hartl, and C. Strauss, "Minimizing total tardiness on a single machine using ant colony optimization," *Central European Journal for Operations Research and Economics*, vol. 8, no. 2, pp. 125–141, 2000.
- [26] M. den Besten, T. Stützle, and M. Dorigo, "Ant colony optimization for the total weighted tardiness problem," in *Parallel Problem Solving from Nature PPSN VI*, M. Schoenauer, K. Deb, G. Rudolph, X. Yao, E. Lutton, J. J. Merelo, and H.-P. Schwefel, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 611–620.
- [27] D. Merkle and M. Middendorf, "An ant algorithm with a new pheromone evaluation rule for total tardiness problems," in *Real-World Applications of Evolutionary Computing*, S. Cagnoni, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 290–299.
- [28] M. Azizoglu and O. Kirca, "Tardiness minimization on parallel machines," *International Journal of Production Economics*, vol. 55, no. 2, pp. 163–168, 1998.
- [29] J. García-Álvarez, M. n. González, C. Rodríguez Vela, and R. Varela, "Electric vehicle charging scheduling by an enhanced artificial bee colony algorithm," *Energies*, vol. 11, no. 10, pp. 115–124, 2018.
- [30] J. García-Álvarez, M. A. González, and C. Rodríguez Vela, "Meta-heuristics for solving a real-world electric vehicle charging scheduling problem," *Applied Soft Computing*, vol. 65, pp. 292–306, 2018.