Electric Vehicle Charging under Power and Balance Constraints as Dynamic Scheduling☆

Alejandro Hernández-Arauzo[a], Jorge Puente[a], Ramiro Varela[a,*], Javier Sedano[b]

[a] *University of Oviedo. Department of Computing. West Building Department, 1. Campus of Gijón. 33204 Gijón.*

[b] *Instituto Tecnológico de Castilla y León (ITCL). Calle López Bravo 70, Polg. Ind. Villalonquejar, 09001 Burgos.*

Abstract

Scheduling the charging periods for a large set of electric vehicles with the objective of satisfying the user demands may be a very hard problem due to the physical constraints of the charging stations. In this paper, we consider a problem of this family which is motivated by a real life situation where a set of users demand electric charge while their vehicles are parked. Each stall has a charging point which is connected to one of the lines of a three-phase electric feeder. There are power constraints that limit the number of vehicles that can be charging at the same time on the same line and balance constraints that limit the difference in the number of vehicles charging in every two lines. We model this problem in the framework of Dynamic Constraint Satisfaction Problem (DCSP) with Optimization, and propose a solution procedure that requires solving a sequence of CSPs over time. Each one of these CSPs requires in its turn solving three instances of a one machine sequencing problem with variable capacity. This procedure was implemented on a simulator of the charging station and evaluated on a number of instances defined from different scenarios of vehicle arrivals and energy requirements. The results of the experimental study show clearly that the proposed algorithm is effective and that it produces schedules much better than those computed by a classic dispatching rule.

☆Some preliminary results of this research work has been discussed in (Hernández-Arauzo et al., 2013)

*Corresponding author.

E-mail addresses: alex@uniovi.es (Alejandro Hernández-Arauzo), puente@uniovi.es (Jorge Puente), ramiro@uniovi.es (Ramiro Varela), javier.sedano@itcl.es (Javier Sedano)

Keywords: Scheduling electric vehicles charging; Dynamic Constraint Satisfaction Problems; Power and balance constraints; One machine sequencing; Tardiness minimization

1. Introduction

It is well known that the use of Electric Vehicles (EVs) may have a positive impact on both the economies of the countries and the environment due to promoting the use of alternative sources of energy and relieving the dependency of foreign petrol. Furthermore, the energy stored in EVs may be utilized as an ancillary service resource (Kang et al., 2013) for regulating frequency and voltage profiles as well as to compensate fluctuations in renewable energy generation (Dallinger, 2014). At the same time, the emerging fleet of EVs introduces some inconveniences such as the additional load on the power system and the time required to charge the batteries. As pointed in (EDSO, 10 April 2012), one of the challenges in EVs' technology is developing smart systems for charging control to avoid increasing peak demand. This has given rise to the design of some algorithms for charging control (Wu et al., 2012). Most of these algorithms try to fill the overnight valley in demand in order to reduce daily cycling and operational cost of power plants (Gan et al., 2007). This raises the issue of coordinating EVs charging patterns (Ma et al., 2013) that may be addressed by means of centralized or distributed strategies. In the first case, a central operator decides when and at what rate every EVs will charge; while in the second, individual EVs determine their own charging pattern on the basis of time-of-day or electricity price, for example.

In this paper we consider a real life problem that requires scheduling the charging intervals of a set of EVs that demand power while they are parked in their own spaces within a community car park. A charging station is installed in the car park so that each space has an independent charging point. A centralized control system establishes the available power at each charging point over time. If the power demand is very large during a given time period, not all the requiring vehicles can be charged simultaneously, as the contracted power is limited. So, in these situations, an appropriate scheduling policy is necessary to organize and control the charging intervals of the vehicles along the time they are in the car park (Sedano et al., 2013).

We propose modeling the problem of computing such a schedule in the framework of Dynamic Constraint Satisfaction Problems (DCSP) with Optimization. As it is usual, one problem of this class requires solving a number of CSPs over time. In order to solve each one of these CSPs, we propose an algorithm that requires solving a number of instances of a one machine scheduling problem with variable machine capacity. The scheduling algorithm is evaluated by means of simulation and compared with the Latest Starting Times (LST) rule.

The rest of the paper is organized as follows. In the next section we summarize the characteristics of the charging station that are relevant from the point of view of the scheduling algorithm. Then, we review some of the recent literature related to EVs charging. After this, we give a problem definition; firstly, we define the problem as if it were a static CSP and then we give a more realistic definition as a DCSP. We follow with a detailed description of the solving procedure. Finally, we report the results of the experimental study and give some conclusions and ideas for future research.

2. Description of the charging station

In this section we summarize the main characteristics of the electrical structure and the operation mode of the charging station. These elements are detailed in (Sedano et al., 2013). Figure 1 shows a schema of the distribution net of the charging station which is designed to be installed in a community park where each user has his/her own stall. The net is feeded by a three-phase source of electric power with a voltage between phases of 400v. Each stall has a charging point and receives power from a single-phase at 230v; it can supply energy at a constant rate (2.3Kw) in the so called mode I (Sedano et al., 2013). So, for a given contracted power, there is a maximum number of vehicles charging in each line at the same time. Also, the consumption in the three lines should be similar at any time. Otherwise, the net is imbalanced and there is current in the neutral point. In this case, there are power losses which are in direct ratio with the imbalance. Moreover, in the Spanish law (BOE, 22 September 2013) there are legal restrictions about imbalanced systems which do not allow the installation of devices that produce large imbalances without the consentient of the supplier company.

The operation of the station is controlled by a distributed system comprising a master and a number of slaves. Every two consecutive charging points in the same line are under the control of the same slave. The master access the database where the vehicles' data and the charging schedule are stored. It receives information about the state of the charging vehicles from the slaves, and transmits to the slaves starting times and durations of charging intervals. So the slaves are responsible for activating and deactivating charging points as well as registering asynchronous

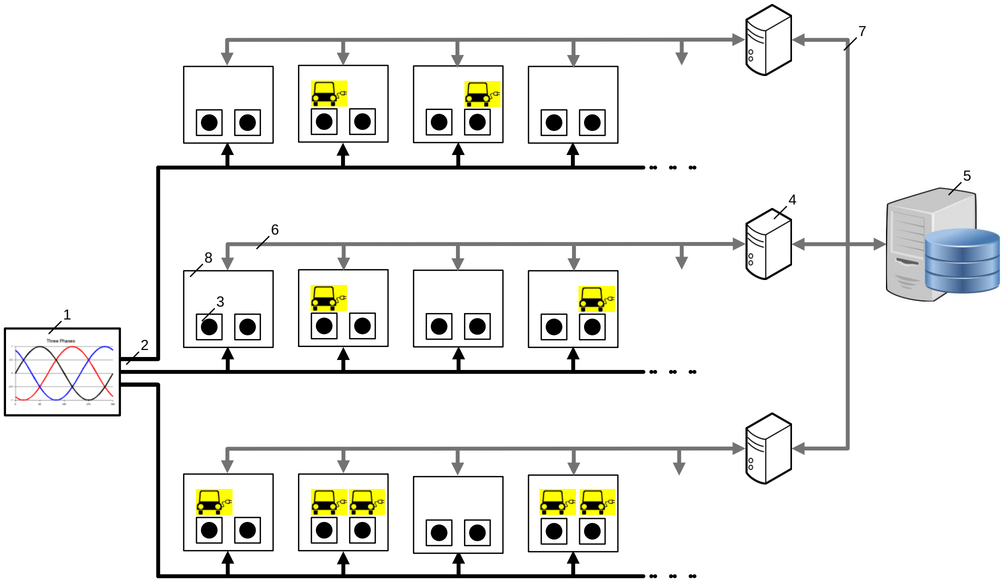


Figure 1: General structure of the distribution net of charging stations. It is formed by different parts such as: (1) power source, (2) three-phase electric power, (3) charging points, (4) masters, (5) server with database, (6) communication RS 485, (7) communication TCP/IP, (8) slaves.

events such as a new vehicle arriving to the system. When entering in the station, the user parks the vehicle in his own stall (as he cannot use the stall of another user) and connects the vehicle to the charging point. Then, he has to provide the charging time and and the time he will take the vehicle away. From these data, the control system schedules the charging periods of the vehicles.

In this paper we consider a simplified model of the charging station which make the following assumptions: the user never takes the vehicle away before the declared due date and the battery does not get completely charged before the charging time indicated by the user. Even though they are unrealistic assumptions, the model may be adapted to dealing with these situations by introducing new asynchronous events.

In principle, each time a new vehicle requires charging, the current schedule may get unfeasible and so a new schedule should be built. However, in order to avoid the system to collapse if many of such events are produced in a very short period of time, due to the fact that a new schedule could not be obtained in the short time interval from one event to the next, new schedules are computed at most at time intervals of length ΔT (typically 120s). In order to do that, the protocol is the following: every ΔT time units a supervisor program, running on the server, checks for the events produced in the last interval. If at least one event was produced that may make the current schedule unfeasible, then the scheduler is executed to obtain a new feasible schedule which is applied from this time onwards.

3. Literature review on scheduling of electric vehicle charging

In this section we review some recent works dealing with scheduling the charge periods of electric vehicles. Our purpose is to demonstrate that this is a hot topic of research and that the problem has been considered from different points of view. In all cases, the proposed scheduling algorithms are evaluated by simulations.

A scenario where a number of EVs have to be scheduled over T time slots satisfying due date constraints is considered in (Gan et al., 2007). Each EV may be charged at different rates in different time slots and the objectives are minimizing the peak demands and filling the valleys of power availability. To solve this problem, the algorithm ODC (optimal decentralized charging) is proposed in (Gan et al., 2011). The reduction of peak power consumption is also considered in (Kim et al., 2012), where feasible schedules are built based on the activation time, the deadline, and the power load profile of each charging task. In (Kang et al., 2013), a real-time scheduling EV charging system is proposed that extends the valley-filling charging strategy so that it tries to optimize total load variance and owners' preferences. Based on EVs requirements, the scheduler establishes a feasible charging schedule in accordance with the power constraints and sends activation/deactivation signals to the charging stations.

In (Lopes et al., 2009), some charging strategies are evaluated using a three phase power flow. The load imbalance between phases is an objective to be minimized. This makes a difference with considering the imbalance as a hard constraint, as we do in this paper, which makes more difficult to find feasible schedules.

In (Wu et al., 2012) a framework is proposed that operates in two steps. Firstly, a loading scheduling algorithm determines the purchase of energy for the day-ahead and then a dispatch algorithm distributes energy to the vehicles. A vehicle charging facility with N charging docks is described in (Tran et al., 2013). Based on different information availabilities about arrivals, charging times and deadlines, a scheduling algorithm decides which dock the vehicle is assigned. The vehicles in a dock are then sequentially scheduled in a First Come First Served basis.

In (Ma et al., 2013), the authors develop a strategy to coordinate charging of large populations of autonomous EVs using concepts from non-cooperative games. The objective is minimizing the overnight valley. The existence of a unique Nash equilibrium that almost satisfies the optimum is proved. This method is based on a decentralized control strategy that utilizes real-time marginal electricity price information and allows each EV to implement its own charging control so that it may minimize its individual charging cost.

In (Clement-Nyns et al., 2010), the impact of Plug-in Hybrid EVs (PHEVs) on the distribution grid is analyzed. The authors try to reduce power losses or voltage deviation by means of coordinate charging when these vehicles are charged at home. The relationship between feeder losses, load factor and load variance are also analyzed in (Sortomme et al., 2011), where three charging algorithms are developed with the objective of minimizing the impact of charging EVs on the distribution system.

In (Sundstrm and Binding, 2010), the authors consider two models of battery behavior, linear and quadratic, and propose an approach implemented on IBM ILOG CPLEX library to optimize the vehicle battery charging with the objective of optimizing costs, power balancing over time and energy levels.

4. Modeling frameworks

Given the characteristics of the charge scheduling problem, we propose to use the dynamic constraint satisfaction problem framework (DCSP) introduced in (Dechter and Dechter, 1988) as modeling framework. A DCSP is a sequence of CSPs, $\langle P_1, P_2, \ldots, P_n \rangle$, where each P_i, $1 < i \leq n$, is derived from P_{i-1} by adding and removing a limited number of constraints. Some variants of the DCSP framework has been proposed that capture other characteristics such as dynamic domains of the variables, state variables which are controlled by the physical system and not by the decision maker, or the uncertainty about the presence of some constraints. However, none of these characteristics appears in the version of the charging scheduling problem considered here. All of these and other frameworks are surveyed in (Verfaillie and Jussien, 2005).

There are two main types of methods to solve a DCSP: reactive and proactive. A reactive algorithm does not use knowledge of the possible changes, so it may not produce robust solutions, but at the same time it may react better to any kind of change. On the contrary, a proactive method is able to exploit any available knowledge and so it may produce robust or flexible solutions. In both cases, the algorithms may either reuse the solutions of the previous CSP or compute a new solution from scratch. Each of these options has its own advantages and drawbacks. Solution reuse may speed up the calculation of a new solution, but at the same time may prevent the algorithm from obtaining a better one. Applications of both methods can be found in the literature, for example in (Le and Pang, 2013) a reactive scheduling framework is proposed to improve energy efficiency in flexible manufacturing systems. The goal is minimizing the sum of energy cost and tardiness penalty. The dynamic nature of the problem comes from power consumption uncertainties of the jobs. Also, in (Ghezail et al., 2010) the authors consider proactive scheduling and propose a method based on graphical representations to analyze how perturbations may affect to a schedule. They consider the classic flow shop scheduling problem as case study.

From all the above, we have opted to use a reactive method with the purpose of obtaining a good solution for each CSP. In the next sections we give the formal definition of the problem as a DCSP and describe the proposed algorithms. Before this, we also give a formal definition of the problem as a static CSP, which assumes a complete knowledge in advance about vehicle arrivals. This helps to understand the subsequent dynamic definition.

5. Definition of the PI problem as a static CSP

As we have pointed, if the problem data, i.e., the arrival times of the vehicles and their charging times and due dates were known in advance, the problem could be formalized as a static CSP. Even though this is not the case for our problem, we consider here a static version of it. The main purpose is to clarify the overall problem; however, this definition could also be useful to obtain solutions which could be compared with those of the dynamic problem. In the next subsections we give the problem data, the goal, the problem constraints and the evaluation function to be optimized.

5.0.1. Problem data

In an static instance $\mathbf{P}$ of the PI problem there are 3 charging lines L_i, $1 \leq i \leq 3$, each one having n_i charging points. $N > 0$ is the maximum number of charging points that can be active at the same time in each one of the three lines. The line L_i receives a number of M_i vehicles $\{v_{i1}, \ldots, v_{iM_i}\}$ from a time 0 up to a planning horizon. Each vehicle v_{ij} is characterized by an arrival time $t_{ij} \geq 0$, a charging time p_{ij} and a time at which the user is expected to take the vehicle away, or due date, d_{ij} by which the battery of the vehicle should be charged.

There is also a parameter $\Delta \in [0,1]$ which controls the maximum imbalance among the lines.

5.0.2. Goal

The goal is to get a feasible schedule for $\mathbf{P}$, i.e., assigning starting times to the decision variables st_{ij} for each vehicle v_{ij}, $1 \leq i \leq 3$, $1 \leq j \leq M_i$, satisfying the constraints and optimizing the evaluation function.

5.0.3. Constraints

I. For all vehicle v_{ij}, $st_{ij} \geq t_{ij}$.

II. No preemption is allowed, so a vehicle v_{ij} cannot be disconnected before its charging time C_{ij} is reached, i.e., $C_{ij} = st_{ij} + p_{ij}$.

III. The number of active charging points in a line at a given time cannot exceed N, i.e.,

$$\max_{(t \geq 0; i=1,2,3)} N_i(t) \leq N \tag{1}$$

where $N_i(t)$ denotes the number of charging points of line L_i which are active during the time interval $[t, t+1)$.

IV. The maximum imbalance between any two lines L_i and L_j is controlled by the parameter Δ as

$$\max_{(t \geq 0; 1 \leq i,j \leq 3)} (|N_i(t) - N_j(t)|/N) \leq \Delta \tag{2}$$

5.0.4. Evaluation function

The evaluation function is the total tardiness defined as

$$\sum_{i=1,2,3; j=1,\ldots,M_i} \max(0, C_{ij} - d_{ij}) \tag{3}$$

which must be minimized.

6. Definition of the PI problem as a DCSP

The PI problem may be naturally considered as a dynamic problem due to the fact that the arrival of vehicles is not known in advance. For this reason, an instance $\mathbf{P}$ can be defined as a sequence of instances, $\mathbf{P_1}, \mathbf{P_2}, \ldots$ of a static CSP termed PII. Each $\mathbf{P_k}$ is defined (see the next Section) from the set of vehicles in the system which have not yet completed their charging periods[1].

To solve this problem, we adopted here a similar strategy to that used in (Rangsaritratsameea et al., 2004) for the dynamic Job Shop Scheduling problem where the jobs are unknown until they arrive. In that paper, the authors propose to build a new schedule at each "rescheduling point" combining all previous operations that have not started processing together with operations arriving after the previous rescheduling point.

Due to technological restrictions, we do not consider rescheduling each time a new vehicle arrives. Instead, we consider rescheduling each time the Supervisor is activated. The new schedule involves the vehicles which have arrived from the previous point together with all the vehicles in the system which have not yet started to charge.

6.1. Solving the dynamic PI problem

Algorithm 1 shows a simulation of the actual algorithm to solve a dynamic PI problem. In the simulation, the problem data and the sequence of times for the Supervisor to be executed are given to the algorithm. The algorithm iterates on this sequence of times $T_1, T_2, \ldots$. In the iteration k, i.e., at time T_k, a new instance $\mathbf{P_k}$ of PII is created if some vehicle arrived in $[T_{k-1}, T_k)$. P_k includes all the vehicles in the system that have arrived before T_k and that have yet not started to charge. This instance is solved and the new solution S defines starting times st^*_{ij} for all the vehicles in P_k. The solution S remains active at least along $[T_k, T_{k+1})$. This means that for all $st^*_{ij} \in [T_k, T_{k+1})$, st_{ij} is fixed to st^*_{ij} in the solution to the $\mathbf{P}$ instance and so the vehicle is disconnected at the completion time $C_{ij} = st_{ij} + ps_{ij}$. If no vehicle has arrived in $[T_{k-1}, T_k)$ then the current solution in this interval remains active at least in the next one $[T_k, T_{k+1})$.

[1] We use here a formulation that differs from the conventional definition given in (Dechter and Dechter, 1988) where a DCSP is a sequence $P_0, P_1, \ldots$ of static CSPs in which P_i is a restriction or a relaxation of P_{i-1}. As we will see in the next sections, we also consider a sequence of CSPs to solve a PI instance. However, each of these instances may be obtained from the previous one by adding new constraints and doing some relaxations.

7. Definition of the PII problem

The PII problem can be defined as a static CSP as follows: In an instance $\mathbf{P_k}$, we are given a set of vehicles $\{v_{i1}, \ldots, v_{il_i}, \ldots, v_{im_i}\}$ at time T_k in each line L_i, $1 \leq i \leq 3$. Each vehicle v_{ij} requires a charging time p_{ij} and has a due date d_{ij}. The vehicles $v_{i1}, \ldots, v_{il_i}$ are already active, as they started to charge at a time $t < T_k$ and have not yet finished, i.e., $C_{ij} = st_{ij} + p_{ij} > T_k$. While the vehicles $v_{il_i+1}, \ldots, v_{im_i}$ have not yet started to charge. So, in the iteration k, the capacity of the line L_i to charge new vehicles, denoted $M_i^k(t)$, is given by

$$M_i^k(t) = N - \sum_{1 \leq j \leq l_i} X_{ij}(t), \ \ t \geq T_k \tag{4}$$

where

$$X_{ij}(t) = \begin{cases} 1, & t < C_{ij} \\ 0, & t \geq C_{ij} \end{cases} \tag{5}$$

The objective is to obtain a feasible schedule for all vehicles in the system such that all of them can be sorted out, even if no new vehicles arrive after T_k. This requires assigning starting times st^*_{ij} to all vehicles unscheduled at time T_k,

Algorithm 1 Solving the PI problem.

```
Require: The data of a $\mathbf{P}$ instance of the PI problem: $t_{ij}, p_{ij}$ and $d_{ij}$ for all vehicles $v_{ij}$; and the sequence of times $T_1, T_2, \ldots$ at which the Supervisor is activated.
Ensure: A schedule $\mathbf{S}$ for $\mathbf{P}$ defined by the time each vehicle starts to charge $st_{ij}$ and the total tardiness produced by this schedule.
  $S = \emptyset$;
  for all $k = 1, 2, \ldots$ do
    if a new vehicle $v_{ij}$ has arrived at a time $t = t_{ij} \in [T_{k-1}, T_k)$ then
      Generate a new instance $\mathbf{P_k}$ of the problem PII with all vehicles $v_{ij}$ s.t. $t_{ij} < T_k$ and that have not started charging yet;
      Calculate a solution $S$ to $\mathbf{P_k}$; {A solution $S$ defines starting times $st^*_{ij} \geq T_k$ to schedule all vehicles $v_{ij}$ that are not active at $T_k$}
    end if
    Establish $S$ as the current solution along $[T_k, T_{k+1})$; i.e., for each $st^*_{ij} \in S$ such that $T_k \leq st^*_{ij} < T_{k+1}$, set $st_{ij} = st^*_{ij}$ in the final schedule $\mathbf{S}$, i.e., $v_{ij}$ will start charging at $st^*_{ij}$;
  end for
  return the schedule $\mathbf{S}$ for $\mathbf{P}$ and its total tardiness;
```

which are compatible with the starting times of the vehicles already scheduled before T_k. This means that all the constraints naturally derived from the static PI problem must be satisfied.

Also, the evaluation function will be, in principle, minimizing the total tardiness. However, as the solution to $\mathbf{P_k}$ is expected to be useful along a short time period (hopefully along a small number of iterations), we will try to maximize the number of charging vehicles at the beginning. So, we could consider a time horizon t_h at which a new event may be expected and try to maximize the charge along the interval $[T_k, T_k + t_h]$. This new objective may be expressed as maximizing

$$\int_{T_k}^{T_k+t_h} (N_1(t) + N_2(t) + N_3(t))dt \tag{6}$$

where $N_i(t); i = 1, 2, 3$ denotes the number of active vehicles in line L_i at time t.

7.0.1. What changes from $\mathbf{P_k}$ to $\mathbf{P_{k+1}}$

As we have pointed, $\mathbf{P_{k+1}}$ is obtained from $\mathbf{P_k}$ by adding new constraints and doing some relaxations. In this case, the only relaxation comes from the vehicles that started to charge along the interval $[T_k, T_{k+1})$ as these vehicles take part in $\mathbf{P_k}$, but not in $\mathbf{P_{k+1}}$. So, if no vehicle started to charge in that interval, there is no relaxation from $\mathbf{P_k}$ to $\mathbf{P_{k+1}}$. At the same time, there are three types of new constraints that may appear in $\mathbf{P_{k+1}}$ w.r.t. $\mathbf{P_k}$. The first one is due to the difference in the capacity of each line L_i; clearly $M_i^{k+1}(t) \leq M_i^k(t)$, for all t, as we can consider $M_i^k(t) = 0, t < T_k$. The second constraint is due to the starting times of the vehicles which are not scheduled before T_{k+1} so that they remain in $\mathbf{P_{k+1}}$, as $st_{ij} \geq T_k$ in $\mathbf{P_k}$ and $st_{ij} \geq T_{k+1}$ in $\mathbf{P_{k+1}}$, while the due dates d_{ij} are the same in both problems. Finally, the third constraint is given by the vehicles arriving in $[T_k, T_{k+1})$.

So, $\mathbf{P_{k+1}}$ is never a relaxation of $\mathbf{P_k}$ due to the fact that the first constraint is always added. However, if no new vehicles arrive in $[T_k, T_{k+1})$, it is clear that the solution to $\mathbf{P_k}$ can be trivially adapted to $\mathbf{P_{k+1}}$.

7.1. Solving the PII problem

The PII problem is really hard to solve because of the constraint derived from constraint (IV) of the static PI problem, which for the instance $\mathbf{P_k}$ may be expressed as

$$\max_{(t \geq T_k; 1 \leq i,j \leq 3)} (|N_i(t) - N_j(t)|/N) \leq \Delta \tag{7}$$

It is not easy to build a schedule satisfying this constraint and at the same time maximizing expression (6). To solve this problem, we propose to use a dispatching rule and a more sophisticated algorithm based on problem decomposition.

7.1.1. Solving PII with dispatching rule

We propose to use the following dispatching rule, termed LST (Latest Starting Time), to solve each $\mathbf{P_k}$ instance. The unscheduled vehicles at time T_k in the system are sorted in accordance with their latest starting times given by $d_{ij} - p_{ij}$ from low to high values; then these vehicles are scheduled in this order and each one is given the earliest starting time such that the scheduled vehicles satisfy all the constraints. In this way, the vehicles that arrived before T_k but have not yet started to charge by T_{k+1} may be assigned a different starting time in the next schedule.

This rule can be easily implemented, but the restriction that the balance constraint must be satisfied after scheduling each vehicle is very strong and may clearly prevent the algorithm from reaching near optimal solutions.

7.1.2. Solving PII by problem decomposition

We propose here a more sophisticated method to maintain the balance constraints which does not require keeping these constraints after each operation is scheduled. The method relies on the following ideas. First of all, we establish profiles of maximum charge, $N_i^{max}(t), i = 1, 2, 3$. Then, to solve $\mathbf{P_k}$, we try to obtain a schedule for each line L_i, so that $N_i(t)$ is as close as possible to $N_i^{max}(t)$ for $t \geq T_k$ while it satisfies the constraint

$$N_i(t) \leq N_i^{max}(t), \;\; t \geq T_k \tag{8}$$

If the combination of the solutions to the three lines does not give rise to a feasible solution to $\mathbf{P_k}$, the profiles $N_i^{max}(t)$ must be adjusted and new schedules have to be computed for one or more lines.

The problem of calculating a schedule for a line subject to a maximum load is denoted PIII herein and the instance of this problem which consist in scheduling the vehicles in the line L_i, subject to the profile $N_i^{max}(t)$ at time T_k, is denoted $\mathbf{P_{ki}}$. So, our proposed method starts from some initial profiles and then these profiles are updated as long as the solutions obtained to the $\mathbf{P_{ki}}$ instances, $1 \leq i \leq 3$, do not make up a solution to the $\mathbf{P_k}$ instance.

Algorithm 2 describes the calculation of a solution to a $\mathbf{P_k}$ instance. The algorithm starts from trivial profiles $N_i^{max}(t)$ and then iterates until a solution is reached. In each iteration, it solves the three $\mathbf{P_{ki}}$ instances subject to the profiles

Algorithm 2 Solving the PII problem.

```
Require: The data of an instance $\mathbf{P_k}$ of the PII problem: $p_{ij}$ and $d_{ij}$ for all unscheduled vehicles $v_{ij}$ that arrived by $T_k$; and the values $st_{ij}$ and $p_{ij}$ for all vehicles scheduled before $T_k$ such that $st_{ij} + p_{ij} \geq T_k$.
Ensure: A schedule $S$ for $\mathbf{P_k}$ defined by the time each vehicle starts to charge $st^*_{ij}$ and the total tardiness produced by this schedule.
  Set the initial profiles to $N_i^{max}(t) = N, t \geq T_k, 1 \leq i \leq 3$;
  while $\mathbf{P_k}$ remains unsolved do
    Solve the instances $\mathbf{P_{ki}}$ under the current profiles $N_i^{max}(t)$; {The three PIII instances get solved with charge profiles $N_i(t)$}
    Let $t' \geq T_k$ be the earliest time such that an imbalance exists, i.e., $N_i(t') - N_j(t') > \Delta \times N$ for some $1 \leq i, j \leq 3$;
    if there exists such a time $t'$ then
      Adjust the profile of maximum load for the line $L_i$ so that $N_i^{max}(t) \leq N_j(t') + \Delta \times N, t \geq t'$;
    else
      The solutions to the $\mathbf{P_{ki}}$ instances, $1 \leq i \leq 3$, make up a solution $S$ to $\mathbf{P_k}$;
    end if
  end while
  return the schedule $S$ for $\mathbf{P_k}$ and its total tardiness;
```

$N_i^{max}(t)$. If these solutions make up a solution for $\mathbf{P_k}$, the algorithm finishes; otherwise, some profile is adjusted from the earliest time t' at which an imbalance is detected onwards. In this way, the profiles are maintained as large as possible at the beginning and so, hopefully, the evaluation function given in expression (6) is maximized. As it can be expected, the adjustment of the profiles is the most controversial operation in this algorithm. We consider this issue in Section 9.

8. Definition of the PIII problem

In an instance $\mathbf{P_{ki}}$ of the PIII problem we are given the set of vehicles $\{v_{i1}, \dots, v_{il_i}, \dots, v_{im_i}\}$ at time T_k in the line L_i. Additionally, we are given a maximum charge profile for the line L_i, $N_i^{max}(t), t \geq T_k$.

The objective is to obtain a schedule for the vehicles, i.e., starting times $st^*_{ij} \geq T_k$ for the inactive vehicles $v_{il_i+1}, \dots, v_{im_i}$, such that the following two constraints, derived from the PII instance, are satisfied:

i. $st^*_{ij} \geq T_k$, for each inactive vehicle.

ii. $N_i(t) \leq N_i^{max}(t)$, for all $t \geq T_k$.

The evaluation function is the total tardiness, defined as

$$\sum_{j=l_i+1,\ldots,m_i} \max(0, C_{ij} - d_{ij}) \tag{9}$$

which must be minimized.

8.1. Solving the PIII problem

The $\mathbf{P_{ki}}$ problem can be viewed as that of scheduling a number of $m_i - l_i$ jobs, all of them available at time T_k, on a machine whose capacity varies along the time and the objective is minimizing the total tardiness. The processing time of the jobs are the charging times of the vehicles $v_{il_i+1}, \ldots, v_{im_i}$, respectively. Each job can use only one slot of the machine at a time. In other words, the machine is a cumulative resource with variable capacity. Cumulative scheduling has been largely considered in the literature, mainly in the context of the Resource Constrained Project Scheduling Problem (RCPSP). However, to the best of our knowledge, cumulative resources with time dependent capacity have not been considered yet.

In our case, the capacity of the machine is defined by the profile $N_i^{max}(t)$ and the vehicles already scheduled $v_{i1}, \ldots, v_{il_i}$, which complete charging at times $C_{ij} \geq T_k$. So, the capacity of the machine may be expected to be increasing at the beginning, as long as the scheduled vehicles complete charging, and decreasing at the end, as the profiles $N_i^{max}(t)$ are non increasing along time. To be concrete, the capacity of the machine at time t for the line L_i, $Cap_i^k(t)$, is calculated as

$$Cap_i^k(t) = \min(M_i^k(t), N_i^{max}(t)), \;\; t \geq T_k \tag{10}$$

We denote this problem as $(1, Cap(t)||\sum T_i)$ following the conventional notation $(\alpha|\beta|\gamma)$ proposed in (Graham et al., 1979). Figure 2 shown an instance of this problem.

8.1.1. Solving the $(1, Cap(t)||\sum T_i)$ problem

In the simple case where the capacity $Cap(t)$ is non decreasing, the problem is equivalent to the problem of identical parallel machines with variable availability denoted $(P, NC_{inc}||\sum T_i)$ following the notation used in (Schmidt, 2000), where P is the number of parallel machines and N_{inc} denotes that the availability of machines is non decreasing along the time. Scheduling problems with machine availability appear in many situations, for example when maintenance periods are

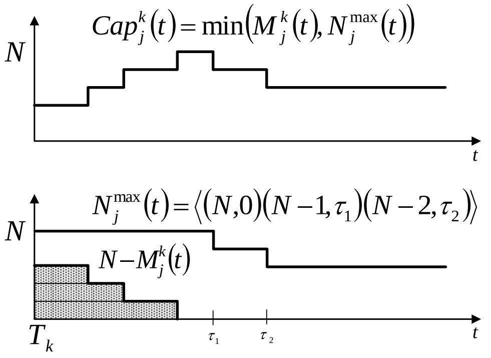

$Cap_j^k(t) = \min(M_j^k(t), N_j^{\max}(t))$
N
t
$N_j^{\max}(t) = \langle (N,0)(N-1,\tau_1)(N-2,\tau_2) \rangle$
N
$N - M_j^k(t)$
T_k
τ_1
τ_2
t

Figure 2: An instance of the $(1, Cap(t)|| \sum T_i)$ obtained at a time T_k where three vehicles are charging in line j and the profile of maximum load has been adjusted from τ_1 to $N-1$ and from τ_2 to $N-2$.

considered, with different profiles of machine availability. This kind of problems is surveyed in (Ma et al., 2010).

In (Koulamas, 1994), the $(P|| \sum T_i)$ problem, in which all the machines are continuously available, is proved to be at least binary NP-hard. An efficient simulated annealing algorithm for this problem is proposed in (Sang-Oh Shim and Kim, 2007). In this algorithm, the starting solution is obtained by means of the apparently tardiness rule. This rule was adapted for similar problems in (Kaplan and Rabadi, 2012), to deal with ready times and due date constraints. In this paper, we propose to adapt this rule to solve the $(1, Cap(t)|| \sum T_i)$ problem as follows: let $\Gamma(\alpha)$ be the earliest starting time for an unscheduled job in the partial schedule α built so far. Then for all unscheduled jobs that can start at $\Gamma(\alpha)$ a selection probability is calculated as

$$\Pi_j = \frac{1}{p_j} \exp\left[\frac{-max(0, d_j - \Gamma(\alpha) - p_j)}{g\overline{p}}\right] \tag{11}$$

where $\overline{p}$ is the average processing time of the jobs and g is a look-ahead parameter to be fixed empirically. These probabilities may be applied deterministically, i.e., the job j with the largest probability is selected to be scheduled next, or probabilistically. In principle, we will consider the first option in the experimental study.

9. Profiles of maximum load

As we have pointed, the balance among the lines is the most critical issue of the whole charge scheduling algorithm. In order to deal with it, we propose to use the following model for the profiles of maximum load. A profile $N_i^{max}(t)$ is given by a stepwise non increasing function of the form:

$$N_i^{max}(t) = \begin{cases} \delta_j & \tau_j \leq t < \tau_{j+1}, 1 \leq j < k \\ \delta_k & \tau_k \leq t \end{cases} \tag{12}$$

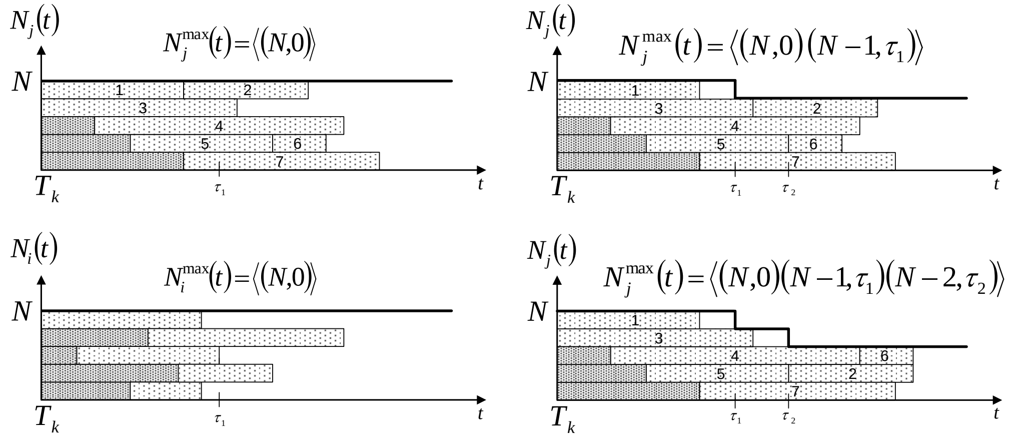
where $N \geq \delta_1 > \cdots > \delta_k$ and $\tau_1 < \cdots < \tau_k$, $k \geq 1$. We represent this profile as a sequence of tuples as: $\langle(\delta_1, \tau_1), (\delta_2, \tau_2), \ldots, (\delta_k, \tau_k)\rangle$.

In Algorithm 2, the initial profiles are $N_i^{max}(t) = \langle(N, 0)\rangle$ for all three lines. Then, these profiles are adjusted as long as new imbalances are found after the solutions of the three PIII instances. In particular, when an imbalance of the form $N_i(t') - N_j(t') > \Delta \times N$ is detected, then the profile $N_i^{max}(t)$ is modified in such a way that a new element $(\delta, \tau) = (\Delta \times N + N_j(t'), t')$ is inserted and all tuples (δ_j, τ_j) with $\delta_j > \delta$ and $\tau_j > \tau$ are removed from $N_i^{max}(t)$.

This is a very simple model which helps to keep the load in the three lines as large as possible at the beginning, hopefully along the interval $[T, T + t_h]$. However, it may have some inconvenience as well. For example, a new imbalance may be produced at a time just after to t'. To avoid this drawback, we could adjust the new tuple as $(\delta - \delta_H, \tau - \tau_H)$, where $\delta_H \geq 0$ and $\tau_H \geq 0$ are parameters to be established empirically. Also, the next imbalance may be at a time lower than t' due to the non-preemption constraint. In any case, the value of $N_i^{max}(t)$, for each time t, is non increasing along the subsequent adjustments. This guarantees that Algorithm 2 terminates after a finite number of steps. Figure 3 shows a situation where the profile of the line L_j undergoes two adjustments before reaching a compatible solution with that of line L_i.

10. Experimental study

As it was done in the models summarized in Section 3, we evaluated the scheduling algorithm by simulation. To do that, we have firstly defined a set of instances of the PI problem and then we implemented a simulator to run the Algorithm 1, termed EVS (Electric Vehicle Scheduling) herein. In the next subsection, we give the details of the benchmark defined and summarize the results



$N_j(t)$
$N_j^{\max}(t)=\langle(N,0)\rangle$
N
1
2
3
4
5
6
7
T_k
τ_1
t
$N_i(t)$
$N_i^{\max}(t)=\langle(N,0)\rangle$
N
T_k
τ_1
t
$N_j(t)$
$N_j^{\max}(t)=\langle(N,0)(N-1,\tau_1)\rangle$
N
1
3
2
4
5
6
7
T_k
τ_1
τ_2
t
$N_j(t)$
$N_j^{\max}(t)=\langle(N,0)(N-1,\tau_1)(N-2,\tau_2)\rangle$
N
1
3
4
6
5
2
7
T_k
τ_1
τ_2
t

(a) Initial schedules in lines L_i and L_j. There is an imbalance at time τ_1.

(b) Schedules in line L_j after two adjustments of $N_j^{max}(t)$ at times τ_1 and τ_2.

Figure 3: Illustration of adjustments of the profile of maximum load of the line L_j to get a compatible schedule with that of line L_i. Δ=0.2 and N=5, so the maximum difference in the number of active EVs in the two lines is 2.

of the experimental study[1]. Then we evaluate the proposed EVS algorithm and compare it with the algorithm derived form the dispatching rule described LST. Finally, we include a subsection were we discuss the results and some aspects of the proposed model.

10.1. Definition of the benchmark set

We consider that the charging station is installed in a car park with 180 spaces distributed in the three lines[2]. We have generated some benchmarks considering a time horizon of one day and different profiles of arriving times, battery load, due dates and distribution among the lines. In particular, we have defined the following three benchmark sets corresponding to three plausible situations. The first one represents a profile that may be expected in a normal weekly day with vehicles arriving on the whole day through, but with arrival peaks at some times. In this

[1] The instances and more details of the experimental study are available at http://www.di.uniovi.es/iscop (Repository).

[2] These vehicles should naturally be distributed uniformly among the three lines. However, for the purpose of generating scenarios with different characteristics, we will consider different distributions.

Table 1: Summary of percentages and probability distributions used to generate the arrival time for each percentage of instances in the three scenarios considered. $U(x, y)$ denotes uniform distribution in the interval $[x, y]$ and $N(x, y)$ denotes normal distribution with mean x and standard deviation y. x and y are given in minutes.

scenario 1		scenarios 2 and 3	
%vehicles	Arrival time	%vehicles	Arrival time
10	$U(0, 1440)$	20	$U(0, 1440)$
20	$N(510, 15)$	80	$N(870, 30)$
10	$N(720, 15)$		
50	$N(1170, 15)$		
10	$N(1350, 15)$		

setting, we consider that 10% of the vehicles arrive uniformly along the day, 20% arrive around 8:30, 10% around 12:00, 50% around 19:30 and the remaining 10% around 22:30. In order to simulate these conditions, we have used the probability distributions given in the first two columns of Table 1 (scenario 1). Also, we have considered two more special situations where most of the vehicles arrive almost at the same time; in this case, we consider two due date profiles. Table 1 (scenarios 2 and 3) shows the probability distributions used to generate the arrival times of these instances.

Regarding the charge of the batteries at the time the vehicles arrive to the car park, in the three scenarios we consider that 10% of them are at about 80% of their capacity, 30% at about 50%, 30% at about 35% and the remaining 30% at about 12%; the charging times are then calculated assuming that all the vehicles require charging at 100% of their capacity, 23 Kwh, and that the charging rate is 2.3 Kw. Two different due date profiles are considered; in scenarios 1 and 2 they are about 4, 6, 8 and 11 hours respectively after the arrival time, while in scenario 3 they are about 2, 5, 6 and 9 hours respectively, so representing much more tight due dates. The percentages and distributions used to generate all these values are given in Table 2.

In the three cases, we consider two different profiles of distribution of vehicles along the lines. In the first one (type 1) the vehicles are uniformly distributed, while in the second (type 2) the distribution is 10%, 30% and 60% respectively on lines 1, 2 and 3. Also, we consider four values for the parameter Δ, 0.2, 0.4, 0.6

Table 2: Summary of percentages and probability distributions used to generate the initial charges and due dates for all the instances. Times are expressed in minutes.

%vehicles	Initial charge (in percent)	Due date Sc. 1,2	Due date Sc. 3
10	$N(80, 10)$	$N(240, 120)$	$N(120, 60)$
30	$N(50, 15)$	$N(360, 120)$	$N(300, 60)$
30	$N(35, 7.5)$	$N(480, 120)$	$N(360, 60)$
30	$N(12, 6)$	$N(660, 120)$	$N(540, 60)$

and 0.8, which establishes the maximum imbalance among every two lines; and three different values for parameter N, 20, 30 and 40, which defines the maximum number of vehicles that can be charging at the same time in the same line. So, combining all these options we have 72 classes of instances in all; 30 instances of each class were generated.

For all the instances, the generated data are consistent, i.e., $t_{ij} + p_{ij} \leq d_{ij}$. To ensure this, the charging time is set as $p_{ij} = d_{ij} - t_{ij}$ if the randomly generated value were larger than this. It may be expected that, for a given Δ, instances of type 1 will be easier to solve than instances of type 2 with the same or lower Δ, due to the fact that for the first instances the imbalance constraint seems to be easier to fulfill than for the second. This means that the number of adjustments of the maximum profiles may be much larger for the first instances than for the seconds. Also, it seems reasonable to expect that the lower the value of N, the larger the tardiness of the solutions.

It is also worth noting that solving one PI instance from a 24 hours scenario requires solving 720 PII instances, as ΔT is set to 2 minutes in the experiments, which in its turn requires solving at least 2160 PIII instances, being this number too much optimistic due to the required profile of maximum load adjustments.

10.2. Evaluation of the proposed algorithm

To evaluate EVS, we start considering different values for the lookahead parameter g, see exp.(11). Table 3 summarizes the values of the total tardiness obtained for all instances with four values of g, 0.25, 0.50, 0.75 and 1.00. In all cases, the parameters δ_H and τ_H are set to 0. As it was expected, the instances of the scenario 3 produce the largest tardiness as they represent situations where the due dates are more tight w.r.t. the arrival times and most of the vehicles arrive around the same time. Also, the lowest tardiness is obtained for the instances of

Table 3: Summary of results from EVS with four values of parameter g, the values are averaged for each scenario. The reported values have to be multiplied by E+06 to represent tardiness in minutes.

	Value of parameter g			
scenario	0.25	0.50	0.75	1.00
1	2.024	2.033	2.038	2.045
2	3.081	3.096	3.099	3.104
3	3.256	3.258	3.257	3.258
Average	2.787	2.795	2.798	2.803

scenario 1 where the vehicles arrive more uniformly along the day and the due dates are not so tight.

Regarding the parameter g, it is clear that it has not a strong influence on the tardiness obtained. However, there is a significant difference in favor of value 0.25. Table 4 shows the average ranking computed by the Friedman test for all the instances. This test yielded p-value of 8.97E-11.

Table 4: Average rankings of the EVS algorithm with four values of parameter g (Friedman).

Value of parameter g	Ranking
0.25	1.28
0.50	2.19
0.75	2.76
1.00	3.76

It is also worth analyzing the number of adjustments required to reach a solution to a PII instance, as it may have an important impact on the time required by EVS to reach a solution. Figure 4 shows the mean number of adjustments that EVS, with $g = 0.25$, requires to solve the instances in each subset. As we can observe, this number decreases as long as Δ and N augment, strongly in the first case and moderately in the second. So, these results confirm the hypothesis that larger values of these parameters make the instances easier to solve. In any case, the average number of adjustments is really low and so it can be expected not to

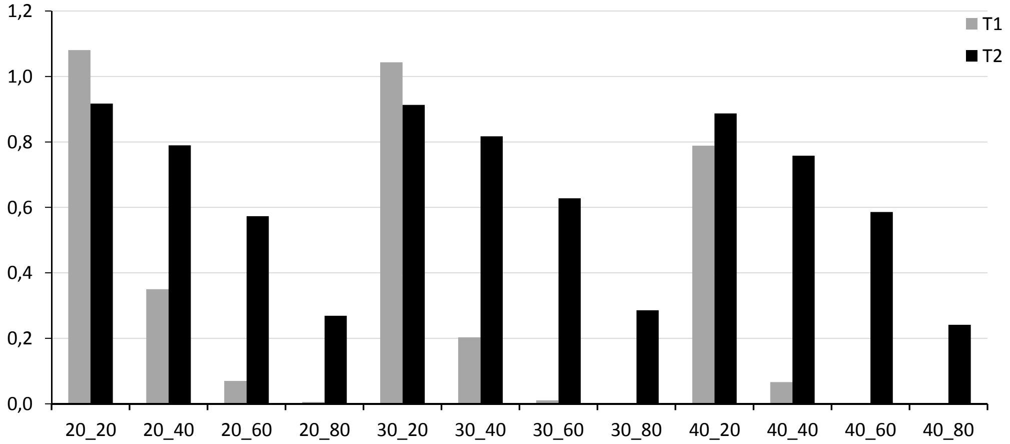


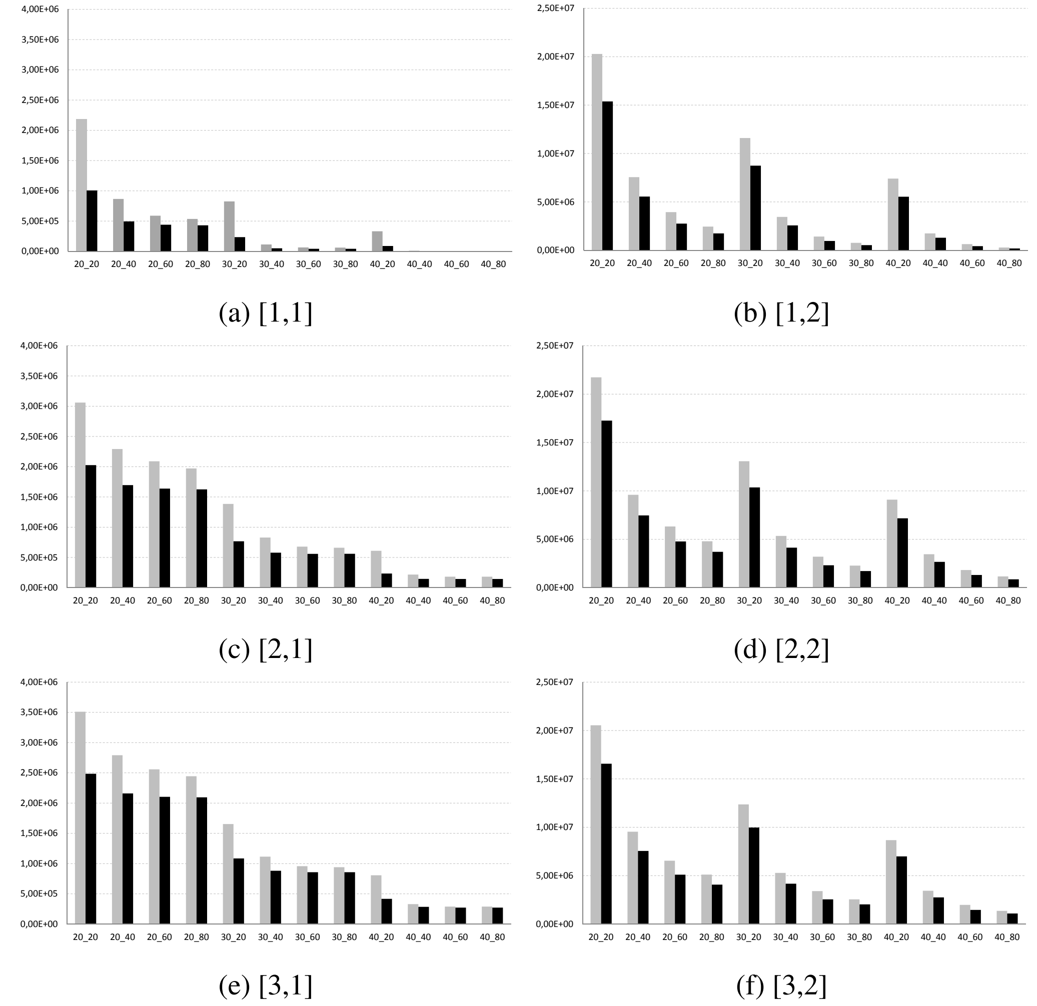
Figure 4: Number of adjustments required by EVS to reach feasible solutions to the instances of the PII problem along the time horizon. x-axis represents the number of adjustments and y-axis represents the subsets $T_i\_N\_\Delta{\times}100$, i denotes the type (1,2). The results are averaged for the 30 instances in each subset.

have a great influence on the time taken by the scheduler. We will come on this issue in the next section.

10.3. Comparison of EVS with LST

All the instances were also solved with the algorithm LST. For none of the instances this rule reached a solution better than that obtained by EVS. Figure 5 shows the comparison of the results obtained with EVS ($g = 0.25$) with those from the LST rule, for each one of the 72 subsets. As we can observe, for all of them the tardiness obtained with LST is larger than that obtained with EVS, the average value from LST being about $33,4\%$ larger than from EVS.

In Figure 5, we can also appreciate the differences in the tardiness of the instances in each one of the 72 subsets. In general, instances of type 2 show more tardiness than instances of type 1, the tardiness growing up quickly from scenario 1 to scenario 2, and moderately from scenario 2 to scenario 3, as it was also shown in Table 3. This is reasonable as in scenario 1 the arrivals are more uniformly distributed along the day, while in scenarios 2 and 3 almost all the vehicles arrive around the same time. Also in scenario 3 the due dates are more tight than they are in scenario 2. At the same time, the tardiness decreases as long as parameters N and Δ increase for all subsets. However, for instances of type 2, the parameter Δ



(a) [1,1]

(b) [1,2]

(c) [2,1]

(d) [2,2]

(e) [3,1]

(f) [3,2]

Figure 5: Results from LST and EVS ($g = 0.25$) averaged for the six subsets defined by [scenario,type], scenario=1,2,3; type=1,2. The x-axis represent total tardiness in seconds, with different scales for each type, and the y-axis represent combinations of $N\_\Delta{\times}100$. Grey bars show LST values and black bars EVS values.

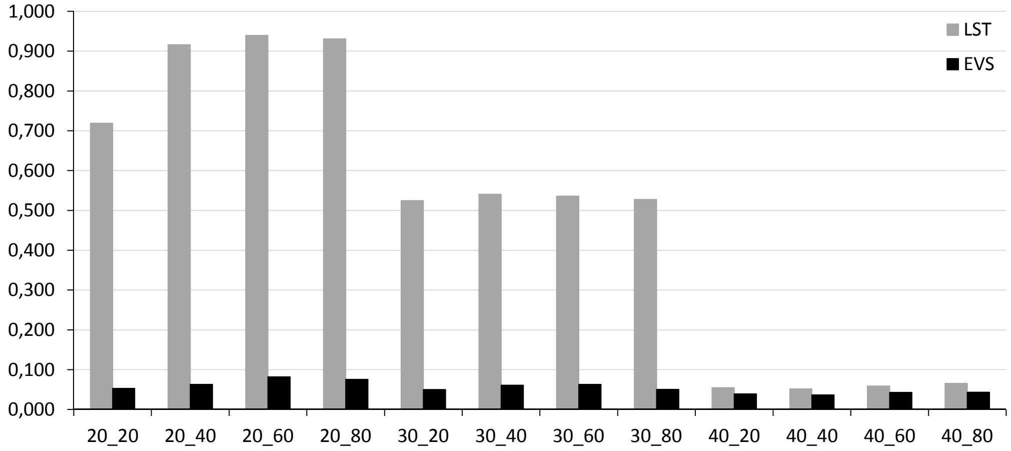


Figure 6: Time taken by EVS and LST to reach feasible solutions to the instances of the PII problem along the time horizon. x-axis represents the time taken in seconds and y-axis represents the subsets $N\_\Delta{\times}100$. The results are averaged for the 60 instances in each subset (30 of type 1 and 30 of type 2).

has stronger influence on the tardiness than it has for type 1. This is quite natural due to the fact that instances of type 2 have a non uniform distribution of vehicles on the three lines and so a large value of Δ allows a large imbalance among the lines and then more vehicles can be charging at the same time in the most loaded lines. Special mention deserves the high similarity of graphics on Figures 5(d) and 5(f). Remember that they represent results from instances generated with the same parameters, with the exception of those that control the due dates. So the charging times and the arrival times of the vehicles are similar, but there are differences of about 90 minutes in the average due dates. So, in principle it is reasonable to expect that the average tardiness for the instances [scenario,type] = [3,2] is at most $90 \times 60 \times 180 = 9.72E$+05 seconds larger than they are for instances [2,2]. This difference being negligible for tardiness of around $1.00E$+07 seconds that produce the hardest instances of type 2. So, this fact together with the reasonable differences in the charging and arriving times makes the results almost indistinguishably.

Regarding the time taken for the algorithms to reach feasible schedules, it is important to remark that it must be much lower than the period of time between two rescheduling points (120s). Figure 6 shows the time taken by LST and EVS across all the instances of types 1 and 2. The results are averaged for each subset defined by the same values of N and Δ. As we can see, in all cases the average time is lower than 1s. For EVS the time is practically independent of the subset of instances and it is always lower than 0.1s. However, for LST the time is strongly dependent on the value of N, it is about 0.9s for $N = 20$, 0.6s for $N = 40$ and 0.3s for $N = 60$. These differences are due to the fact that the algorithm LST has to find the first time at which one operation can be scheduled, i.e., it satisfies all the constraints, and this operation requires a search along intervals of candidate scheduling points. Being the length of these intervals in inverse ratio with the number of vehicles that can be charging in parallel established by N, as these vehicles are more spread along time.

10.4. Discussion of the results

The results of this study shown that the proposed EVS algorithm is effective and more efficient than a dispatching rule such as LST. In our opinion, the performance of EVS relies on how it deals with the imbalance constraints. Instead of keeping this constraint after each operation is scheduled, as it is done by the algorithm derived from the LST rule, we define profiles of maximum load in the three lines and then adapt the schedules to these profiles. Even though these profiles may require a number of adjustments, the time taken by the algorithm is quite reasonable as it is negligible w.r.t. the time between two consecutive executions of the scheduler.

As it was pointed in Section 2, we have considered here a simplified model of the charge scheduling problem. However, the proposed modeling and solving methods can be adapted to dealing with a number of characteristics of the real situations that have been skipped here for the sake of simplicity of presentation. For example, the users may pick up the vehicle before the declared due date d_{ij}, or the battery may get fully charged before the expected charging time p_{ij}. In both cases an imbalance may be produced in the system. To deal with these situations, some new asynchronous events have to be added to the model. Also, the charging time of the vehicles may be reduced in situations of saturation in order to reduce the tardiness of the vehicles. Furthermore, if the tardiness for some vehicle is too large in situations of very high demand, the vehicle may be discarded from the schedule and so not served. The proposed model can also be adapted to variable power, what would allow the operator to adjust the amount of contracted power to the consumption valleys of the grid, for example. Furthermore, the algorithm may be adapted to situations where the vehicles can be charged at non constant rate. As it is pointed in (Sedano et al., 2013), this is technically possible under certain restrictions and offers much more flexibility to organize the charging of vehicles over time. However, this adaptation is not trivial and will require reformulating some aspects the PII problem and adapting the scheduling algorithms accordingly.

11. Conclusions

We have seen that scheduling the charging of electric vehicles may be formulated as a Dynamic Constraint Satisfaction Problem (DCSP) with Optimization. In this paper, we have given a formal definition for one problem of this family. This problem was motivated by a real environment in which a number of vehicles require charge from an electric system installed in a garage where each vehicle has a preassigned space. This problem is hard to solve due to the imbalance constraints among the three lines of the three-phase electric feeder. We have proposed an effective algorithm that reduces the calculation of a solution for the dynamic scheduling problem to solving a number of instances of the one machine sequencing problem with variable capacity, denoted $(1, Cap(t)||\sum T_i)$, which is unedited in the literature and was here solved by a dispatching rule, so it is still open for further research. A formal analysis of the $(1, Cap(t)||\sum T_i)$ problem may lead to the development of new solution methods, which may contribute to improve the performance of the whole scheduling algorithm. As future work, we will analyze the use of some exact, greedy and evolutionary approaches as it was recently done in (Kodaganallur et al., 2013), for a version of the one machine sequencing problem with setup times and quadratic penalty completion times, or in (Ji et al., 2013) for a more complex version with common due-window, deteriorating rate of the machine and job durations that depend on the amount of resource allocated.

12. Acknowledgments

This research has been supported by the Spanish Government under research project TIN2010-20976-C02-02, by the ITCL under contract FUO-EM-075-13 an by the Principality of Asturias under contract FC-13-COF13-035.

13. References

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/`.

Clement-Nyns, K., Haesen, E., Driesen, J., 2010. The impact of charging plug-in hybrid electric vehicles on a residential distribution grid. IEEE Transactions on Power Systems 25, 371–380.

Dallinger, D., 2014. Plug-in electric vehicles integrating fluctuating renewable electricity. Kassel university press GmbH.

Dechter, R., Dechter, A., 1988. Belief maintenance in dynamic constraint networks, in: Proceedings of the Seventh Annual Conference of the American Association of Artificial Intelligence, pp. 37–42.

EDSO, 10 April 2012. Position paper on electric Vehicles Charging Infrastructure. European Distribution system Operators for Smart Grids (EDSO). `http://www.edsoforsmartgrids.eu`.

Gan, L., Topcu, U., Low, S., 2007. Optimal decentralized protocols for electric vehicle charging. IEEE Transaction on Power Systtems 6, 1–10.

Gan, L., Topcu, U., Low, S., 2011. Optimal decentralized protocol for electric vehicle charging, in: IEEE Conference on Decision and Control and European Control Conference, CDC-ECE, pp. 5798–5804.

Ghezail, F., Pierreval, H., Hajri-Gabouj, S., 2010. Analysis of robustness in proactive scheduling: A graphical approach. Computers & Industrial Engineering 58, 193 – 198.

Graham, R., Lawler, E., Lenstra, J., Kan, A., 1979. Optimization and approximation in deterministic sequencing and scheduling: a survey. Annals of Discrete Mathematics 5, 287 – 326.

Hernández-Arauzo, A., Puente, J., González, M.A., Varela, R., Sedano, J., 2013. Dynamic scheduling of electric vehicle charging under limited power and phase balance constraints, in: Proceedings of SPARK'13. ICAPS'13 Scheduling and Planning Applications workshop, ICAPS council. pp. 1–8.

Ji, M., Ge, J., Chen, K., Cheng, T., 2013. Single-machine due-window assignment and scheduling with resource allocation, aging effect, and a deteriorating rate-modifying activity. Computers & Industrial Engineering 66, 952 – 961.

Kang, J., Duncan, S.J., Mavris, D.N., 2013. Real-time scheduling techniques for electric vehicle charging in support of frequency regulation. Procedia Computer Science 16, 767 – 775.

Kaplan, S., Rabadi, G., 2012. Exact and heuristic algorithms for the aerial refueling parallel machine scheduling problem with due date-to-deadline window and ready times. Computers & Industrial Engineering 62, 276–285.

Kim, H.J., Lee, J., Park, G.L., 2012. Constraint-based charging scheduler design for electric vehicles, in: Proceedings of the 4th Asian conference on Intelligent Information and Database Systems - Volume Part III, Springer-Verlag, Berlin, Heidelberg. pp. 266–275.

Koulamas, C., 1994. The total tardiness problem: Review and extensions. Operations Research 42, 1025–1041.

Le, C.V., Pang, C.K., 2013. Fast reactive scheduling to minimize tardiness penalty and energy cost under power consumption uncertainties. Computers & Industrial Engineering 66, 406 – 417.

Lopes, J., Soares, F., Almeida, P.M., Moreira da Silva, M., 2009. Smart charging strategies for electric vehicles: Enhancing grid performance and maximizing the use of variable renewable energy resources, in: EVS24 International Battery, Hybrid and Fuel Cell Electric Vehicle Symposium, 2009, pp. 392 –396.

Ma, Y., Chu, C., Zuo, C., 2010. A survey of scheduling with deterministic machine availability constraints. Computers & Industrial Engineering 58, 199–211.

Ma, Z., Callaway, D., Hiskens, I., 2013. Decentralized charging control of large populations of plug-in electric vehicles. IEEE Transactions on Control Systems Technology 21, 67–78.

Rangsaritratsameea, R., Ferrell, W.G., Kurzb, M.B., 2004. Dynamic rescheduling that simultaneously considers efficiency and stability. Computers & Industrial Engineering 46, 1–15.

Sang-Oh Shim, S.O., Kim, Y.D., 2007. Scheduling on parallel identical machines to minimize total tardiness. European Journal of Operational Research 177, 135–146.

Schmidt, G., 2000. Scheduling with limited machine availability. European Journal of Operational Research 121, 1–15.

Sedano, J., Portal, M., Hernández-Arauzo, A., Villar, J.R., Puente, J., Varela, R., 2013. Intelligent system for electric vehicle charging: Design and operation. Technical Report. Instituto Tecnológico de Castilla y León ITCL 88, 640–647.

Sortomme, E., Hindi, M., MacPherson, S., Venkata, S., 2011. Coordinated charging of plug-in hybrid electric vehicles to minimize distribution system losses. Smart Grid, IEEE Transactions on 2, 198–205.

Sundstrm, O., Binding, C., 2010. Optimization methods to plan the charging of electric vehicle fleets, in: Proc. Int'l Conf. on Control, Communication and Power Engineering. CCPE 2010, pp. 323–328.

Tran, T.T., Dogru, M.K., Ozen, U., Beck, C., 2013. Scheduling a multi-cable electric vehicle charging facility, in: Proceedings of SPARK'13. ICAPS'13 Scheduling and Planning Applications woRKshop, ICAPS council. pp. 20–26.

Verfaillie, G., Jussien, N., 2005. Constraint solving in uncertain and dynamic environments: A survey. Constraints 10, 253–281.

Wu, D., Aliprantis, D., Ying, L., 2012. Load scheduling and dispatch for aggregators of plug-in electric vehicles. IEEE Transactions on Smart Grid 3, 368–376.