Proceedings of the 7th
Scheduling and Planning Applications woRKshop

SPARK 2013

Rome, Italy - June 11, 2013

Edited By:

Luis Castillo Vidal, Steve Chien, Riccardo Rasconi

Organizing Commitee

Luis Castillo Vidal
IActive, Spain

Steve Chien
Jet Propulsion Laboratory, USA

Riccardo Rasconi
ISTC-CNR, Italy

Program committee

Laura Barbulescu, Carnegie Mellon University, USA
Mark Boddy, Adventium, USA
Luis Castillo, IActive, Spain
Steve Chien, Artificial Intelligence Group, Jet Propulsion Laboratory, USA
Gabriella Cortellessa, ISTC-CNR, Italy
Minh Do, SGT Inc., NASA Ames, USA
Patrik Haslum, NICTA, Australia
Russell Knight, Artificial Intelligence Group, Jet Propulsion Laboratory, USA
Jana Koehler, Hochschule Luzern, Lucerne Univ. of Applied Sciences and Arts, Switzerland
Nicola Policella, ESA-ESOC, Germany
Riccardo Rasconi, ISTC-CNR, Italy
Bernd Schattenberg, University of Ulm, Germany
Tiago Vaquero, University of Toronto, Canada
Ramiro Varela, Universityof Oviedo, Spain
Gèrard Verfaillie, ONERA, France
Neil Yorke-Smith, American University of Beirut, Lebanon, and SRI International, USA
Terry Zimmerman, SIFT, USA

Foreword

Application domains that entail planning and scheduling (P&S) problems present a set of compelling challenges to the AI planning and scheduling community, from modeling to technological to institutional issues. New real-world domains and problems are becoming more and more frequently affordable challenges for AI. The international Scheduling and Planning Applications woRKshop (SPARK) was established to foster the practical application of advances made in the AI P&S community. Building on antecedent events, SPARK'13 is the seventh edition of a workshop series designed to provide a stable, long-term forum where researchers and practitioners can discuss the applications of planning and scheduling techniques to real-world problems. The series webpage is at http://decsai.ugr.es/~lcv/SPARK/

We are once more very pleased to continue the tradition of representing more applied aspects of the planning and scheduling community and to perhaps present a pipeline that will enable increased representation of applied papers in the main ICAPS conference.

We thank the Program Committee for their commitment in reviewing. We thank the ICAPS'13 workshop and publication chairs for their support.

Edited by
Luis Castillo Vidal, Steve Chien and Riccardo Rasconi

Table of Contents

Dynamic Scheduling of Electric Vehicle Charging under Limited Power and Phase Balance Constraints

Alejandro Hernández[1] and **Jorge Puente**[1] and **Miguel A. González** and **Ramiro Varela**[1] and **Javier Sedano**[2]
[1]Department of Computer Science,
University of Oviedo, 33271 Gijón (Spain)
e-mail: {alex, puente, ramiro}@uniovi.es
[2]Instituto Tecnológico de Castilla y León (ICTL)
e-mail: javier.sedano@itcl.es

Abstract

We confront the problem of scheduling the charge of electric vehicles, under limited electric power contract, with the objective of maximizing the users' satisfaction. The problem is motivated by a real life situation where a set of users demand electric charge while their vehicles are parked. Each space has a charging point which is connected to one of the lines of a three-phase electric feeder. We first define the problem as a Dynamic Constraint Satisfaction Problem (DCSP) with Optimization. Then, we propose a solution method which requires solving a number 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. We evaluated the proposed algorithm by means of simulation across some instances of the problem. The results of this study show that the proposed scheduling algorithm is effective and produces much better results than some classic dispatching rules.

Introduction

It is well known that the use of Electric Vehicles (EVs) may have a positive impact on the economies of the countries and on the environment, due to promoting the use of alternative sources of energy and relieving the dependency of foreign petrol. At the same time, the emerging fleet of EVs introduces some inconveniences such as the additional load on the power system. However, the charge of EVs is usually more flexible than the conventional load of oil vehicles as in many cases the owners require charging while their vehicles are parked during large time periods. This flexibility may be exploited to design appropriate algorithms for charging control (Wu, Aliprantis, and Ying 2012).

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. However, 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. 2012).

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 some dispatching rules such as First Come First Scheduled (FCFS) or Latest Starting Times (LST).

The rest of the paper is organized as follows. In the next section we summarize the aspects of the charging station that are relevant from the point of view of the scheduling algorithm. Then, we define the problem and describe the proposed algorithm to solve it. Finally, we report the results of the experimental study and give some conclusions and ideas for future research.

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. 2012). Figure 1 shows a schema of the distribution net of the charging station. The net is feeded by a three-phase source of electric power. In the model considered here, each line feeds a number of charging points. The station has about 180 spaces, each one having a charging point which may be in two states: active or inactive. When it is inactive, the charging point is not connected to the electric net, while in active state it is connected to the net and transfers energy at a constant rate (2.3Kw) in the so called mode 1 (Sedano et al. 2012).

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 has access to 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 charg-

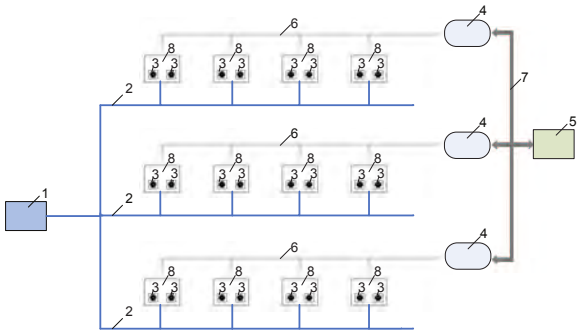

Figure 1: General structure of the distribution net of charging stations. It is formed by different parts such as: (1) power source, (2) three-phase electric power, (3) charging points, (4) masters, (5) server with database, (6) communication RS 485, (7) communication TCP/IP, (8) slaves.

ing points as well as registering asynchronous events such as a new vehicle arriving to the system.

The operating mode of the station is as follows. Each user has one vehicle and one space assigned. These are concrete spaces as each user has to be the owner or the renter of its stall and he cannot use the space of another user. This restriction makes the scheduling problem harder to solve as each stall is connected to one of the three lines of the three-phase feeder and so keeping the balance constraints may not be easy. So, for a vehicle and user can use the station, they have to be registered in the system. When entering in the station, the user has to check in and identify himself and the vehicle. At this time, the user has to connect the vehicle to the charging point and provide the charging time, as well as an expected time, or due date, for taking the vehicle away. These values are then used by the control system to schedule the vehicle, i.e., to establish a starting time. In principle, the vehicle will start to charge at this starting time unless the schedule is modified before it.

There are some constraints that must be satisfied for the station to work properly. For example, although there are 180 spaces available, not all the charging points in these spaces can be activated at the same time. In practice there is a maximum number of vehicles N that can be in charge (actives) at the same time in a line which depends on the contracted power. Also, due to electro-technical and economical reasons, the current consumption in the three lines must be balanced. This condition is considered as a maximum imbalance between any two lines.

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 deal with these situations with nothing more than 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 so that a new schedule cannot be obtained from one event to the next, new schedules are computed at most at time intervals of length ΔT. 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 could make the current schedule unfeasible, then the scheduler is launched to obtain a new feasible schedule which is applied from this time on.

Modeling frameworks

Given the characteristics of the charge scheduling problem, maybe the most appropriate framework for modeling is the dynamic constraint satisfaction problem framework (DCSP) introduced in (Dechter and Dechter 1988). 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.

A particular case of proactive method is the online stochastic optimization framework (Chang, Givan, and Chong 2000), (Bent and Van Hentenryck 2004), which has been applied to a variety of problems where a set of requests are given over time. Online stochastic optimization algorithms rely on two main components: an offline solver for a set of requests, and a sampler that generates fictitious requests with a given distribution along a time horizon. The online stochastic algorithms solve instances which are composed by subsets of requests including real and fictitious ones, then they take decisions from the solutions of these fictitious instances. So, the application of online stochastic optimization to the charge scheduling problem may make it difficult to maintain the balance constraints if the processing times of the fictitious operations differ much from the charging periods required for the incoming vehicles.

From all the above, we have opted to model the electric vehicle charging scheduling problem, termed PI in the sequel, in the framework of DCSP with optimization and to use a reactive method to solve it. So, in the next sections we give the formal definition of the problem 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 the vehicle arrivals. This helps to understand the subsequent dynamic definition.

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 purpose is twofold, firstly to clarify the overall problem and then to define the simulation framework which will be used in our experimental study. In the next subsections we give the problem data, the goal, the problem constraints and the evaluation function to be optimized.

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.

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} satisfying the constraints and optimizing the evaluation function.

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

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

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.

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.

To solve this problem, we adopted here a similar strategy to that used in (Rangsaritratsameea, Ferrell, and Kurzb 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.

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

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. $t_{ij} \leq T_k$ and that have not started charging yet;
    Calculate a solution S for the instance $\mathbf{P_k}$; {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 $\mathbf{S}$, so that v_{ij} starts charging at st^*_{ij};
end for
return the schedule $\mathbf{S}$ for $\mathbf{P}$ and its total tardiness;

time T_k, a new instance $\mathbf{P_k}$ of PII is created if some vehicle arrived from the previous instant T_{k-1}. This instance is solved and the new solution replaces the current one from T_k on. If no vehicle has arrived from T_{k-1} to T_k then the current solution can remain active until the next iteration. The current solution is applied during the time it is active. This means that the vehicles start charging at the times st^*_{ij} given in the current solution, so st_{ij} is fixed to st^*_{ij} in the solution to the $\mathbf{P}$ instance, and disconnected at their expected times $C_{ij} = st_{ij} + ps_{ij}$.

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 $M^k_i(t)$ is given by

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

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, which are compatible with the starting times of the vehicles already scheduled by 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 the instance $\mathbf{P_k}$ is expected to be useful along a short time period (until the arrival of a new vehicle), 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.

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 two simple dispatching rules and a more sophisticated algorithm based on problem decomposition.

Solving PII with a dispatching rule. We propose to use a simple dispatching rule termed LTS (Latest Starting Time) to solve each $\mathbf{P_k}$ instance. This rule works as follows: the unscheduled vehicles at time T_k in the system are sorted in accordance with their latest starting times given by $d_{ij} - p_{ij}$, 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. This rule can be easily implemented, but the restriction that the balance constraint must be satisfied after scheduling each vehicle is very hard and may prevent the algorithm from reaching near optimal solutions.

Solving PII by problem decomposition. We also propose a method that uses problem decomposition in the following way. First of all, we establish a profile of maximum charge, $N^{max}_i(t), i = 1, 2, 3$, for each one of the three lines; so that these profiles satisfy the constraint

$$\max_{(t \geq T_k; 1 \leq i,j \leq 3)} (|N^{max}_i(t) - N^{max}_j(t)|/N) \leq \Delta \tag{8}$$

Then, we try to obtain a schedule for each one of the lines, so that $N_i(t)$ is as close as possible to $N^{max}_i(t)$ for $t \geq T_k$ while it satisfies the constraint

$$N_i(t) \leq N^{max}_i(t), \quad t \geq T_k \tag{9}$$

Hence, combining the solutions to the three lines may give rise to a feasible solution to the PII instance. If not, the profiles $N^{max}_i(t)$ are adjusted and new schedules are 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 $N^{max}_i(t)$ 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, $1 \leq i \leq 3$, derived from a $\mathbf{P_k}$ instance, 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 $N^{max}_i(t)$ and then iterates until a solution is reached. In each iteration, it solves the three $\mathbf{P_{ki}}$ instances subject to the profiles $N^{max}_i(t)$. 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. The adjustment of the profiles is the most controversial operation in this algorithm. We will reconsider this issue later.

Algorithm 2 Solving the PII problem.

Require: The data of an instance $\mathbf{P_k}$: 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 $\mathbf{S}$ 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 to $\mathbf{P_k}$;
  end if
end while
return the schedule $\mathbf{S}$ for $\mathbf{P}$ and its total tardiness;

Definition of the PIII problem

In an instance $\mathbf{P_{ki}}$ of the PIII problem we are given the set of vehicles $\{v_{i1}, \ldots, v_{il_i}, \ldots, v_{im_i}\}$ 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 $v_{il_i+1}, \ldots, v_{im_i}$, 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

$$\sum_{j=l_i+1,\ldots,m_i} \max(0, C_{ij} - d_{ij}) \tag{10}$$

which must be minimized.

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 has 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

$$Cap_i^k(t) = \min(M_i^k(t), N_i^{max}(t)), \quad t \geq T_k \tag{11}$$

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

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 considered, with different profiles of machine availability. This kind of problems are surveyed in (Ma, Chu, and Zuo 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 $\Gamma(\alpha)$ 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

$$\Pi_j = \frac{1}{p_j} \exp\left[\frac{-max(0, d_j - \Gamma(\alpha) - p_j)}{g\overline{p}}\right] \tag{12}$$

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.

Profiles of maximum load

As we have pointed out, the balance among the lines is the most critical issue of the whole charge scheduling problem. 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:

$$N_i^{max}(t) = \begin{cases} \delta_j & \tau_j \leq t < \tau_{j+1}, 1 \leq j < k \\ \delta_k & \tau_k \leq t \end{cases} \tag{13}$$

where $\delta_1 > \cdots > \delta_k$ 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 so as 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.

Experimental study

As it was pointed out, 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. In the next two subsections, we give the details of the benchmark defined and summarize the results of the experimental study.

Benchmark set

We consider that the charging station is installed in a car park with 180 spaces distributed uniformly in the three lines. We have generated some benchmarks[1] considering a time horizon of one day and a profile for arrival times which are based on the expected behavior of the users in some weekdays. Also, we have considered different demand and due date profiles.

Figure 2 shows the arrival profile of the vehicles along the day. As we can observe, there are peaks of arrivals at four different times of the day. The processing times p_{ij} and due dates d_{ij} follow the profiles represented in Table 1. We consider four different profiles depending on the state of the battery at the arrival time. The second column of the table represents the probabilities of each case and the two last columns represent the probability distributions for the values p_{ij} and d_{ij}, which are given by means of normal distributions.

We have defined two types of instances, in the first one (type 1), 60 vehicles arrive at each line L_i along the day and demand charging, while in the second (type 2) the vehicles are 108 in L_1, 54 in L_2 and 18 in L_3, i.e., 60%, 30% and 10% respectively. So, in the later case we may expect that the scheduling algorithm has to control many situations

[1]These instances are available at http://www.di.uniovi.es/tc (Repository).

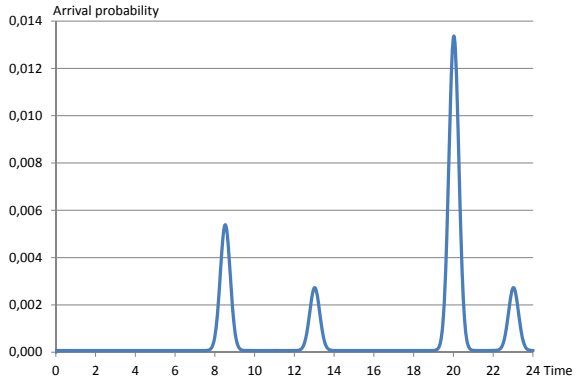

Figure 2: Arrival profile of vehicles along a day. x-axis represents the time of the day from 0 to 24 hours and the y-axis represents the arrival probability.

Table 1: Charging time and due date profiles used to generate PI instances. $N(x, y)$ denotes a normal distribution with mean x and standard deviation y. Time is given in hours.

Case	Prob.	p_{ij}	d_{ij}
1	0.1	$N(2, 1)$	$t_{ij} + \max(p_{ij}, N(4, 2))$
2	0.3	$N(5, 1.5)$	$t_{ij} + \max(p_{ij}, N(6, 2))$
3	0.3	$N(6.5, 0.75)$	$t_{ij} + \max(p_{ij}, N(8, 2))$
4	0.3	$N(8.8, 0.6)$	$t_{ij} + \max(p_{ij}, N(11, 2))$

of imbalance among the lines in order to build a feasible schedule. Also, we will consider different values for the imbalance parameter Δ (0.2, 0.4, 0.6 and 0.8) and for the maximum number of vehicles that can be charging at the same time in a line N (20, 30 and 40). 30 instances were generated for each combination of type, Δ and N, so we have 720 instances in all.

Evaluation of the proposed algorithm

Our main purpose is to evaluate the proposed algorithm to solve the problem PI, termed PD (Problem Decomposition) herein, under different demand conditions. We also compare it with two algorithms: a single dispatching rule FCFS (First Come First Scheduled) that could be implemented by a human operator, and the aforementioned scheduling algorithm that uses the LST rule. In FCFS, the vehicles are scheduled in the order they arrive to the car park and then each one is scheduled at the earliest time such that all the constraints of the problem PI are satisfied. After this, the starting time is never changed.

Table 2 summarizes the results of these experiments. Each line of the table shows the average tardiness obtained for the 30 instances of the same type and the same values of Δ and N. The parameter ΔT was set to 120 s. Regarding the time taken by the algorithms, in the case of PD it depends on the number of adjustments required to reach a solution, and it is larger for this algorithm than it is for the dispatching rules FCFS and LST. However, in no case the time required by

PD was larger than 1 s., which is negligible w.r.t. to 120 s. between two consecutive executions of the scheduler. As we can observe the total tardiness is lower with PD than it is with FCFS and LST in almost all the cases. In average, the total tardiness obtained by FCFS and LTS in this benchmark is very similar and it is about $33,4\%$ larger than that obtained by PD.

Analyzing the schedules obtained for the algorithms, we have observed that PD is able to adjust the imbalance of the schedules up to the limit allowed by the parameter Δ, at difference of the dispatching rules, and this is the very reason for its superior performance. As we have conjectured, one of the reasons of the poor performance of FCFS and LST is that they have to keep the imbalance constraint after each operation is scheduled, what requires delaying the starting time of many operations.

Therefore, from these results, we can conclude that the proposed algorithm PD is effective to solve the problem PI.

Adjustments of the maximum charge profiles $N_i^{max}(t)$. 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 to reach a solution to the whole PI problem.

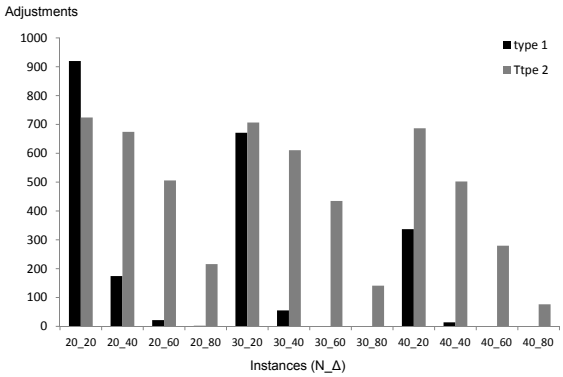

Figure 3: Number of adjustments of the maximum charge profiles depending on the type of the problems (type 1 or type 2), the maximum imbalance Δ (0.2, 0.4, 0.6, 0.8) and the maximum number of active vehicles in one line N (20, 30, 40). The x-axis represents the group of instances for each type of problem ($\Delta \times 100\_N$) and the y-axis the average number of profile adjustments made to solve the PI instances of each class.

Table 2: Summary of results from PD, LST and FCFS on two instances of types 1 and 2 with different values of the parameters Δ and N. The values of tardiness are given in minutes.

PI Instance			Total Tardiness		
Type	N	Δ	FCFS	LST	PD
1	20	0.2	2,03E+06	2,02E+06	9,26E+05
		0.4	7,97E+05	7,96E+05	5,09E+05
		0.6	5,73E+05	5,70E+05	4,62E+05
		0.8	5,45E+05	5,42E+05	4,57E+05
	30	0.2	6,80E+05	6,94E+05	2,14E+05
		0.4	1,08E+05	1,04E+05	5,87E+04
		0.6	6,73E+04	6,69E+04	5,50E+04
		0.8	6,61E+04	6,56E+04	5,50E+04
	40	0.2	2,21E+05	2,32E+05	7,76E+04
		0.4	9,62E+03	9,01E+03	3,35E+03
		0.6	1,50E+03	1,49E+03	1,04E+03
		0.8	1,44E+03	1,43E+03	1,04E+03
2	20	0.2	2,00E+07	2,02E+07	1,53E+07
		0.4	7,56E+06	7,53E+06	5,55E+06
		0.6	3,98E+06	3,97E+06	2,76E+06
		0.8	2,41E+06	2,42E+06	1,78E+06
	30	0.2	1,15E+07	1,15E+07	8,70E+06
		0.4	3,44E+06	3,43E+06	2,57E+06
		0.6	1,40E+06	1,40E+06	9,69E+05
		0.8	7,21E+05	7,20E+05	5,40E+05
	40	0.2	7,37E+06	7,37E+06	5,53E+06
		0.4	1,71E+06	1,72E+06	1,31E+06
		0.6	5,99E+05	6,00E+05	4,22E+05
		0.8	2,60E+05	2,59E+05	1,99E+05
	Average		9,94E+06	9,95E+06	7,46E+06

Figure 3 shows the average number of adjustments required to solve each one of the 24 groups of instances defined by the same type and values of N and Δ. For the instances of type 2, the number of adjustments depends on the allowed imbalance Δ and, as can be expected, this number is in inverse ratio with Δ. The adjustments also depend weakly on the maximum number of active vehicles in a line N, this dependency being also in direct ratio.

Regarding the instances of type 1, where the three lines receive the same proportion of vehicles, these dependencies are much more stronger than they are in the type 2. For the largest values of Δ the number of adjustments is negligible. However, for $\Delta = 0.2$, i.e., when the allowed imbalance is very low, the number of adjustments is really large. The reason for this is that for the instances of type 1, when the allowed imbalance is very low, the scheduling algorithm has to do adjustments in the profiles of maximum load, $N_i^{max}(t)$, in more than one line, while for the problems of type 2 the adjustments are almost restricted to the line with the largest number of vehicles.

Conclusions and future work

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 is termed PI and it is motivated by a real environment in which a number of vehicles may 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 algo-

rithm 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 $(1, Cap(t)||\sum T_i)$. As far as we know, the $(1, Cap(t)||\sum T_i)$ problem was not yet considered in the literature.

The overall charge scheduling algorithm was evaluated by simulation over a benchmark set inspired in some real scenarios. The results of this study shown that the proposed algorithm is better than some dispatching rules such as First Come First Scheduled, which could be followed by a human operator, and a more sophisticated scheduling algorithm that uses the Least Staring Time rule. In our opinion, the performance of our algorithm 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 other two algorithms, 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 algorithm produces much better schedules than the other two algorithms.

This work leaves some issues open for future research. Firstly, we will make a more comprehensive experimental study considering instances derived from different expected scenarios to that considered here. For instance weekend scenarios or situations derived from some contingencies. In this study we will consider some variants of the algorithm that solves the $(1, Cap(t)||\sum T_i)$ problem. As we have mentioned, the apparently tardiness rule may be used probabilistically to obtain a variety of solutions.

Then, we will have to consider a number of characteristics of the real situations that have been skipped here. 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 and other situations, we will have to add new asynchronous events to the model.

Another important characteristic that must be considered is the fact that 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.

Another line for future research will be devoted to generalize the problem formulation and the solution methods to situations where, for example, the contracted power changes over the time or the vehicles can be charged at non constant rate. As it is pointed in (Sedano et al. 2012), the later is technically possible under certain restrictions and offers much more flexibility to organize the charging of vehicles over time. At the same time, it may make the scheduling problem harder to solve.

Acknowledgments

This research has been supported by the Spanish Government under research project TIN-20976-C02-02. We would like to thank the anonymous referees for their useful comments and suggestions.

References

Bent, R., and Van Hentenryck, P. 2004. Regrets only! online stochastic optimization under time constraints. In *Proceedings of the 19th national conference on Artifical intelligence*, AAAI'04, 501–506. AAAI Press.

Chang, H. S.; Givan, R.; and Chong, E. K. P. 2000. Online scheduling via sampling. In *In Artificial Intelligence Planning and Scheduling (AIPS)*, 62–71.

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

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

Kaplan, S., and 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(1):276–285.

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

Ma, Y.; Chu, C.; and Zuo, C. 2010. A survey of scheduling with deterministic machine availability constraints. *Computers & Industrial Engineering* 58(2):199–211.

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

Sang-Oh Shim, S.-O., and Kim, Y.-D. 2007. Scheduling on parallel identical machines to minimize total tardiness. *European Journal of Operational Research* 177(1):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.; and Varela, R. 2012. Sistema de control autónomo para distribución de energía en una estación de carga de vehículos eléctricos: diseño y operación. *Technical Report. Instituto Tecnológico de Castilla y León ITCL.*

Verfaillie, G., and Jussien, N. 2005. Constraint solving in uncertain and dynamic environments: A survey. *Constraints* 10(3):253–281.

Wu, D.; Aliprantis, D.; and Ying, L. 2012. Load scheduling and dispatch for aggregators of plug-in electric vehicles. *IEEE Transactions on Smart Grid* 3(1):368–376.