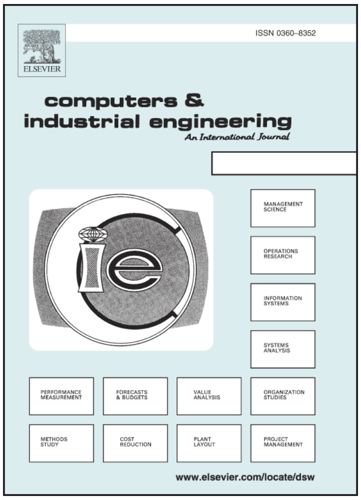


# Genetic Fuzzy Schedules for Charging Electric Vehicles

Jorge García-Álvarez, Inés González-Rodríguez, Camino R. Vela, Miguel A. González, Sezin Afsar

PII: S0360-8352(18)30220-1
DOI: https://doi.org/10.1016/j.cie.2018.05.019
Reference: CAIE 5225

To appear in: *Computers & Industrial Engineering*

Received Date: 6 February 2017
Revised Date: 27 April 2018
Accepted Date: 13 May 2018

# Genetic Fuzzy Schedules for Charging Electric Vehicles

Jorge García-Álvarez[a], Inés González-Rodríguez[b], Camino R. Vela[a], Miguel A. González[a,*], Sezin Afsar[c]

[a]*Dept. of Computing, University of Oviedo, (Spain)*
[b]*Dept. of Mathematics, Statistics and Computing, University of Cantabria, (Spain)*
[c]*Independent researcher*

## Abstract

This work tackles the problem of scheduling the charging of electric vehicles in a real-world charging station subject to a set of physical constraints, with the goal of minimising the total tardiness with respect to a desired departure date given for each vehicle. We model a variant of the problem that incorporates uncertainty in the charging times using fuzzy numbers. As solving method, we propose a genetic algorithm with tailor-made operators, in particular, a new chromosome evaluation method based on generating schedules from a priority vector. Finally, an experimental study avails the proposed genetic algorithm both in terms of algorithm convergence and quality of the obtained solutions.

*Keywords:* electric vehicle, charging station, scheduling, genetic algorithm, fuzzy number, heuristic

## Acknowledgements

This work was supported by the Spanish Government [grant numbers TIN2016-79190-R, MTM2014-55262-P].

[*]Corresponding author

*Email addresses:* jgarcia-alvarez@outlook.com (Jorge García-Álvarez), gonzalezri@unican.es (Inés González-Rodríguez), crvela@uniovi.es (Camino R. Vela), mig@uniovi.es (Miguel A. González), sezinafsar@gmail.com (Sezin Afsar)

# Genetic Fuzzy Schedules for Charging Electric Vehicles

**Abstract**

This work tackles the problem of scheduling the charging of electric vehicles in a real-world charging station subject to a set of physical constraints, with the goal of minimising the total tardiness with respect to a desired departure date given for each vehicle. We model a variant of the problem that incorporates uncertainty in the charging times using fuzzy numbers. As solving method, we propose a genetic algorithm with tailor-made operators, in particular, a new chromosome evaluation method based on generating schedules from a priority vector. Finally, an experimental study avails the proposed genetic algorithm both in terms of algorithm convergence and quality of the obtained solutions.

*Keywords:* electric vehicle, charging station, scheduling, genetic algorithm, fuzzy number, heuristic

## 1. Introduction

Perceived as futuristic prototypes not so long ago, electric vehicles (EVs) and all the technology surrounding them have experimented an extraordinary growth in the last years, to the point that now EVs are seen as a real alternative to fossil-fuelled vehicles, producing an increasing impact on the economy and the environment. Clearly, they reduce the dependency on petrol and promote alternative more environmentally-friendly sources of energy. Also, they can act as valuable distributed energy resources, smoothing intermittency due to renewable energy sources and supporting grid-wide frequency stability (Kang et al., 2013).

The use of EVs also presents several technological challenges, as deciding the optimal locations of new vehicle charging stations You and Hsieh (2014) or the development of smart systems that manage charging grids in order to optimally distribute and balance electricity consumption while satisfying EVs' electricity demands. A great variety of problems arise depending, for instance, on the charging infrastructure or the objectives to be satisfied (Rahman et al., 2016). Recent reviews of methods and strategies to manage electricity grids and schedule EVs charging can be found in Hernandez-Arauzo et al. (2015).

Here we consider a problem that consists in scheduling the charging of a set of EVs taking into account the characteristics of a real-life charging station described in Sedano et al. (2013). The station has been designed to be installed in private car parks under simplicity and economy criteria. Each parking place in the car park is owned by a particular user and has a charging point. Every charging point is connected to one of the lines of a three-phase electric feeder, with a centralised control that establishes the power available to the point at any time. There are power constraints limiting the number of EVs that can be simultaneously charging on one line. There is also a balance constraint that limits the difference in active charging points between different lines. It is this balance constraint that poses the most relevant difference with respect to other charging models in the literature.

We will tackle a variant of the static scheduling problem for the charging station presented by Hernandez-Arauzo et al. (2015). Their problem statement assumes that the set of EVs that need to be charged within a planning horizon is known in advance, together with the exact arrival time, desired departure time and charging time of each EV. The goal is to provide a charging schedule for all the EVs that is feasible in the sense that all technical constraints hold and such that the total tardiness with respect to the desired departure times is minimised. This static version of the problem is of great importance, since it constitutes the basis for solving a dynamic and hence more realistic version. However, it assumes that charging times are exactly known in advance, an assumption that might be deemed as unrealistic. We thus propose in this work to narrow the gap between the academic model and the real situation by incorporating uncertainty into charging times.

We propose to model uncertain charging times using fuzzy sets. The use of fuzzy sets to model uncertain durations in scheduling is well established. Among others, Chanas and Kasperski (2001) minimise maximum lateness in a single machine scheduling problem with fuzzy processing times using an extension of Lawler's algorithm, and the parallel machine scheduling problem with fuzzy processing times is tackled in Balin (2011). We also find numerous works focused on shop scheduling problems with fuzzy durations and, in particular, in the job shop in its multiple variants Liu et al. (2015) Abdullah and Abdolrazzagh-Nezhad (2014). Fuzzy random theory have been successfully used to model imprecise parameters in real-life problems, such as a location-allocation problem in a container freight station Zhong et al. (2015).

There are a few examples where fuzzy sets have also been used in problems related with EV charging. However, they are mostly used to model preferences or soft constraints and to perform multicriteria decision making, for instance, to decide on the optimal site selection of EV charging stations in Guo and Zhao (2015) or to coordinate EV charging in an electric grid in Hajforoosh et al. (2015). More related to our work, in Ansari et al. (2015) fuzzy sets model electricity market uncertainties with the aim of optimising the coordinated bidding of EVs used as ancillary services in the electricity grid. To our knowledge, this is the only proposal in the literature to use fuzzy sets to model uncertain parameters in problems related to EVs, making ours the first attempt to schedule the charging of EV considering fuzzy times.

The remaining of this paper is organised as follows. Section 2 will provide the necessary background on fuzzy numbers and a new result to handle fuzzy charging times. Then, in Section 3 we will show how the use of triangular fuzzy numbers to model charging times leads to a new formulation of the problem.

In Section 4 we will propose a first solving method for the resulting fuzzy problem. This is a genetic algorithm with indirect encoding, based on a competitive method for the deterministic problem given in García-Álvarez et al. (2015), but where some operators need to be adapted to the fuzzy setting. In particular, the decoding operator will consist in a new purpose-built algorithm to obtain feasible charging schedules from a priority vector. The resulting method will be evaluated with an experimental study in Section 5. Finally, some conclusions and perspectives for future work will be given in Section 6.

## 2. Uncertainty in charging times

In real-life problems it is frequent that different sources of uncertainty exist and some input data are not precisely known; in our case, the time it will take for an EV to charge is not exactly known in advance. Currently, crisp charging times are estimated from available historical data, despite the low quality of these data. The alternative of modelling these ill-known times instead with probability distributions is not an option, since it would require to have more and better data. Also, the resulting problem would be excessively complex from the computational point of view. In cases such as this, fuzzy sets constitute a very interesting alternative that may help building a tradeoff between the expressive power and the computational difficulties of stochastic scheduling techniques while tackling uncertainty (Dubois et al., 2003, 2008; Wong and Lai, 2011).

A fuzzy quantity $\widetilde{q}$ is a fuzzy set on the reals $\mathbb{R}$ with membership function $\mu_{\widetilde{q}} : \mathbb{R} \to [0, 1]$. The *$\alpha$-cuts* of a fuzzy quantity are given by $\widetilde{q}_\alpha = \{r \in \mathbb{R} : \mu_{\widetilde{q}}(r) \geq \alpha\}, \alpha \in (0, 1]$, and its *support* is defined as $\widetilde{q}_0 = \{r \in \mathbb{R} : \mu_{\widetilde{q}}(r) > 0\}$. A *fuzzy interval* is a fuzzy quantity whose $\alpha$-cuts are intervals (bounded or not) and a *fuzzy number* $\widetilde{m}$ is a fuzzy quantity with compact support and unique modal value whose $\alpha$-cuts are closed intervals, denoted $\widetilde{m}_\alpha = [\underline{m}_\alpha, \overline{m}_\alpha]$. In our context, a fuzzy interval is seen as a possibility distribution, representing more or less likely possible values for a charging time.

Here we propose that each uncertain charging time be modelled using a *triangular fuzzy number* or TFN, given by an interval $[a^1, a^3]$ of possible values (its support) and a unique modal value $a^2 \in [a^1, a^3]$ with $\mu_{\widehat{a}}(a^2) = 1$. Hence, a TFN $\widehat{a}$ can be denoted $\widehat{a} = (a^1, a^2, a^3)$ and its membership value is given by the following equation:

$$\mu_{\widehat{a}}(x) = \begin{cases} \frac{x-a^1}{a^2-a^1} & : a^1 \leq x \leq a^2 \\ \frac{x-a^3}{a^2-a^3} & : a^2 < x \leq a^3 \\ 0 & : x < a^1 \text{ or } a^3 < x \end{cases} \tag{1}$$

Notice that any real number $a \in \mathbb{R}$ can be seen as a particular case of TFN $\widehat{a} = (a, a, a)$ with all its defining points equal to $a$.

Different proposals on how the TFNs can be elicited from historical data or experts can be found in the literature. Rommelfanger (1990) suggests obtaining three confidence intervals from the decision maker: the core, containing the most typical values; an $\alpha$-cut, with $\alpha = \lambda$ (e.g., $\lambda = 0.5$), containing unsurprising values; and a last $\alpha$-cut, with $\alpha = \varepsilon$ (taking $\varepsilon = 0$, i.e. the support, or $\varepsilon \approx 0$), outside which values are physically unattainable or can be neglected. This is the approach followed by Fortemps (1997) (with $\lambda = 0.5$, $\epsilon = 0$) or Hapke and Slowinski (1996) (with $\lambda = 0.6$, $\epsilon = 0.1$) among others to obtain 6-point fuzzy numbers. As suggested in (Dubois et al., 2003), a simpler model is the triangular fuzzy number, obtained from an interval (containing the possible values) and a single typical value, which provides a good trade-off between expressiveness and simplicity. In our case, an expert could thus provide a TFN modelling the charging time for one kWh, and the charging time of each vehicle would then be obtained by multiplying this TFN by the capacity (in kWh) that needs to be charged.

The fact that membership functions of fuzzy numbers (seen as possibility distributions) encode confidence intervals is also the basis of alternative proposals to derive TFN membership functions from historical data or measurements following a non-parametric approach. From the historical data we can obtain the support $[a^1, a^3]$ and the mode $a^2$ (understood as the most frequent value) of the unknown probability distribution function $p$ underlying those data. Then, the triangular possibility distribution with support $[a^1, a^3]$ and core $a^2$ provides a possibility distribution encoding and dominating the family of all probability distributions with the same support and mode which, in the absence of further knowledge, is shown to provide a tight bound of such family of probability distributions (Baudrit and Dubois, 2006). Furthermore, if only the support is known, with the added assumption that the underlying probability distribution is symmetric, it can be shown that a symmetric triangular possibility distribution with the same support is the tightest possibility distribution dominating the probability distribution, regardless of its shape (Dubois et al., 2004). Notice that representing measurements by nested confidence intervals and obtaining an upper bound of the probability distributions consistent with these intervals is compatible with the ISO Guide for the expression of uncertainty in measurement (Dubois et al., 2004; Mauris et al., 2001).

### *2.1. Arithmetic of TFNs*

To handle charging times given as TFNs, we need to be able to compute the addition, substraction and maximum of two TFNs. In principle, the arithmetic operations on TFNs can be obtained by extending the corresponding operations on real numbers using the *Extension Principle* Dubois and Prade (1986). Both the addition and the substraction of two TFNs $\widehat{a}$ and $\widehat{b}$ are TFNs given by

$$\widehat{a} + \widehat{b} = (a^1 + b^1, a^2 + b^2, a^3 + b^3) \tag{2}$$

$$\widehat{a} - \widehat{b} = (a^1 - b^3, a^2 - b^2, a^3 - b^1) \tag{3}$$

Regarding the maximum, computing the resulting expression is cumbersome, if not intractable; also, the set of TFNs is not closed under this operation. For the sake of simplicity and tractability of numerical calculations, it is fairly common in the literature to approximate the maximum, either using a ranking method, as it is done in Lei (2010), or by interpolation, following Fortemps (1997), evaluating only the operation on the three defining points of each TFN, that is:

$$\max(\widehat{a}, \widehat{b}) \approx \max_I(\widehat{a}, \widehat{b}) = (\max(a^1, b^1), \max(a^2, b^2), \max(a^3, b^3)). \tag{4}$$

The approximation $max_I$ has been widely used in the scheduling literature, from earlier works such as (Fortemps, 1997; Kuroda and Wang, 1996) to more recent ones (Palacios et al., 2015a; Wang et al., 2010), to mention but a few. Additional arguments to support this approximation can be found in (Palacios et al., 2017).

Unless otherwise stated and for the sake of a simpler notation, we shall simply write max when referring to the interpolated maximum $\max_I$.

Notice that, as a particular case, for a real number $b \in \mathbb{R}$, we have $\widehat{a} - b = (a^1 - b, a^2 - b, a^3 - b)$ and $\max(\widehat{a}, b) = (\max(a^1, b), \max(a^2, b), \max(a^3, b))$.

### *2.2. Expected value*

When working with fuzzy numbers, it is often useful in practice to obtain their expected value, similarly to what is done with probability distributions in stochastic settings. In particular, for a TFN $\widehat{a}$, its expected value is given by:

$$E[\widehat{a}] = \frac{1}{4}(a^1 + 2a^2 + a^3). \tag{5}$$

This expression is obtained following different approaches, among others as the expected value of a fuzzy number based on random sets (Heilpern, 1992), as the generative expected value induced by the evidence $\widehat{a}$ (Chanas and Nowakowski, 1988), as the centre of the mean value of $\widehat{a}$ (Dubois and Prade, 1987) and as the expected value of the so-called pignistic probability distribution that is the centroid of the set of probabilities $\mathcal{P}(\Pi_{\widehat{a}})$ dominated by the possibility measure associated with $\widehat{a}$ (Dubois, 2006).

Notice that, for a TFN $\widehat{a}$ it is always the case that the expected value lies in its support, that is, $a^1 \leq E[\widehat{a}] \leq a^3$. It is also easy to provide an alternative expression for the expected value as follows:

$$E[\widehat{a}] = a^2 + \frac{(a^3 - a^2) - (a^2 - a^1)}{4}. \tag{6}$$

An added value of expected values is that they allow to compare fuzzy numbers. Indeed, no natural order exists in the set of fuzzy numbers, so numerous ranking methods have been and keep being proposed in the literature (Brunelli and Mezei, 2013). Several of these methods have been used in the field of fuzzy scheduling (cf. (Abdullah and Abdolrazzagh-Nezhad, 2014; Dubois et al., 2003)). In particular, the expected value defines an index-based ranking method, inducing a total ordering $\leq_E$ in the set of TFNs (Fortemps, 1997), where for any two TFNs $\widehat{a}, \widehat{b}$,

$$\widehat{a} \leq_E \widehat{b} \text{ if and only if } E[\widehat{a}] \leq E[\widehat{b}]. \tag{7}$$

Notice that $\leq_E$ coincides with several other ranking methods from the literature which are not based on expected values, as highlighted in (Palacios et al., 2015b). Additionally, Brunelli and Mezei (2013) present a numerical study that suggests that, for TFNs, the ranking based on the expected value is very similar to seven more ranking methods, in the sense that the ordering they induce in a sample of TFNs is strongly correlated.

Interestingly, it is also possible to establish a relationship between the ranking method based on expected values and classical interval comparison in the light of imprecise probabilities (Destercke and Couso, 2015; Dubois, 2011). In particular, it comes down to using Hurwicz criterion for classical interval comparison on upper and lower expectations derived from $A$. This provides us with an interpretation for comparisons based on $\leq_E$ as those corresponding to a decision maker who keeps an equilibrium between pessimism and optimism.

### *2.3. Defining a TFN subject to a set of constraints*

When scheduling the charge of EVs, we will need to determine fuzzy starting times for the charging of each EV subject to a set of constraints. The following result gives us a means to do so.

**Theorem 1.** *Let $e \in \mathbb{R}$ be a number and $\widehat{c} = (c^1, c^2, c^3)$ a TFN such that $E[\widehat{c}] \leq e$. A TFN $\widehat{s} = (s^1, s^2, s^3)$ satisfying:*

$$E[\widehat{s}] = e \qquad (R_0)$$

$$c^i \leq s^i, i = 1, 2, 3 \qquad (R_{1,2,3})$$

*is given by the following expression:*

$$\begin{array}{lll}
\text{if } E[\widehat{c}] = e, & \widehat{s} = \widehat{c}; & \\
\text{if } E[\widehat{c}] < e, c^3 - e \leq e - c^1, & \widehat{s} = (s^1, s^2, s^3) \text{ where} & \begin{cases} s^3 & = \max(e, c^3), \\ s^2 & = \max(e, c^2), \\ s^1 & = 4e - 2s^2 - s^3; \end{cases} \\
\text{if } E[\widehat{c}] < e, c^3 - e > e - c^1, & \widehat{s} = (s^1, s^2, s^3) \text{ where} & \begin{cases} s^1 & = c^1, \\ s^3 & = c^3, \\ s^2 & = \frac{1}{2}(4e - s^1 - s^3). \end{cases}
\end{array} \qquad (8)$$

Proof. The definition of $\widehat{s}$ in (8) implies that $R_0$ always holds. Let us now prove that $s^i \geq c^i$, $i = 1, 2, 3$. We distinguish three cases depending on how $\widehat{s}$ is defined in (8):

1. If $E[\widehat{c}] = e$, it follows immediately since $\widehat{s} = \widehat{c}$.
2. If $E[\widehat{c}] < e$ and $c^3 - e \leq e - c^1$, then, $s^3 = \max(e, c^3) \geq c^3$ and $s^2 = \max(e, c^2) \geq c^2$. If $e > c^3$, it turns out that $s^3 = s^2 = s^1 = e > c^1$. If $e \leq c^3$ and therefore $0 \leq c^3 - e \leq e - c^1$, we prove that $s^1 \geq c^1$ by distinguishing 2 subcases:

(a) If $c^2 < e$, $s^2 = e$ and $s^3 = c^3$, so (8) together with the fact that $c^3 - e \leq e - c^1$ mean that:

$$s^1 = 2e - c^3 = e - (c^3 - e) \geq e - (e - c^1) = c^1.$$

(b) If $e \leq c^2$, $s^3 = c^3$ and $s^2 = c^2$. But since $e = E[\widehat{s}]$,

$$E[\widehat{s}] = c^2 + \frac{(c^3 - c^2) - (c^2 - s^1)}{4} = e \leq c^2,$$

if and only if

$$(c^3 - c^2) - (c^2 - s^1) \leq 0,$$

or equivalently,

$$s^1 \geq c^2 + c^2 - c^3.$$

But also $E[\widehat{c}] < e$, that is:

$$E[\widehat{c}] = c^2 + \frac{(c^3 - c^2) - (c^2 - c^1)}{4} < e \leq c^2,$$

and therefore, $c^3 - c^2 < c^2 - c^1$, that is, $c^2 - c^3 > c^1 - c^2$. In consequence,

$$s^1 \geq c^2 + c^2 - c^3 > c^2 + c^1 - c^2 = c^1.$$

3. If $E[\widehat{c}] < e$ and $c^3 - e > e - c^1$, then $s^1 = c^1$ y $s^3 = c^3$. Let us now show that $s^2 \geq c^2$. Indeed, $E[\widehat{c}] < e$ if and only if:

$$c^2 < \frac{4e - c^1 - c^3}{2}.$$

but since $c^1 = s^1$, $c^3 = s^3$, by definition of $\widehat{s}$:

$$s^2 = \frac{4e - c^1 - c^3}{2},$$

hence $c^2 < s^2$.

## 3. Problem Description

As mentioned earlier, our problem is motivated by a real charging station, described in detail by Sedano et al. (2013). The station was designed to be installed in a community car park consisting of privately-owned parking spaces such as, for instance, the car park in a block of flats, where each parking place corresponds to a specific flat. Its general scheme is given in Figure 1. The charging grid is fed by a three-phase electric power source, resulting in three so-called charging lines. Each parking space has a charging point connected to one of the lines but the contracted power limits the number of charging points that can be simultaneously fed on each line. Additionally, due to legal and

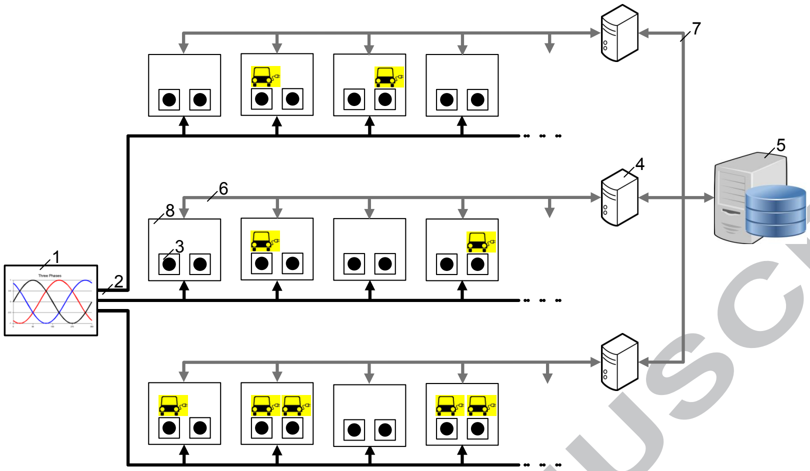


Figure 1: General structure of the charging station network and its components: (1) power source, (2) three-phase electric power, (3) charging points, (4) masters, (5) server with database, (6) communication RS 485, (7) communication TCP/IP, and (8) slaves.

technical reasons, there must be a load balance between different lines, so the power consumption (i.e. number of active charging points) does not differ much from one line to another. The station is controlled by a distributed system with a master agent per line controlling a number of slaves. Each slave in turn controls two consecutive charging points on the same line and is responsible for activating and deactivating them, as well as recording asynchronous events such as the arrival of a new vehicle. The master gathers information from the slaves and sends connection and disconnection orders to them, after accessing the database where the vehicles' data and the charging schedule are stored.

When users park their vehicles in their own parking spaces they need to provide a desired departure time. An estimate of the time needed by the vehicle to charge (charging time hereafter) is also obtained based on the battery capacity that needs to be charged. These values are used by the intelligent system to schedule the charging of all vehicles within a time frame. As proposed in Hernandez-Arauzo et al. (2015), we assume a simplified model of the problem whereby a user never departs before the vehicle has finished charging, so the objective is to minimise the tardiness of the charge completion time with respect to the desired departure date given by the user.

### *3.1. Problem formulation*

Following the general problem description, the Electric Vehicles Charging Scheduling Problem with Fuzzy charging times or FuzEVCSP in short is formulated as follows.

In an instance of the FuzEVCSP problem there are three charging lines $L_i, 1 \leq i \leq 3$. Each line $L_i$ has $n_i$ charging points and there is a maximum number $N > 0$ of charging points that can be simultaneously active. Line $L_i$ receives $M_i$ electric vehicles $v_{i1}, \ldots, v_{iM_i}$ between time 0 and a planning horizon $H$. Each vehicle $v_{ij}$ has an arrival time $t_{ij} \geq 0$, an uncertain charging time $\widehat{p_{ij}} > 0$ (represented as a TFN) and a departure time $d_{ij}$ by which the vehicle should ideally be fully charged, representing the time when the driver wishes to remove the vehicle. Here we assume that the arrival and departure times are crisp times obtained from the user. Additionally, we assume that they are consistent with charging times in the sense that $d_{ij} \geq t_{ij} + p^3_{ij}$. The rationale here is that the departure time should allow for the vehicle to be fully charged provided that it starts charging as soon as it arrives, even if it takes the longest possible charging time.

A *feasible schedule* is an assignment of charging starting times $\widehat{st_{ij}}$ to all EVs $v_{ij}, 1 \leq i \leq 3, 1 \leq j \leq M_i$ such that the following constraints hold.

*3.1.1. Constraints*

*Arrival time.* All vehicles must start charging after their arrival time:

$$\forall v_{ij}, \quad st^1_{ij} \geq t_{ij} \tag{9}$$

*No preemption.* Once a vehicle starts charging, it cannot be disconnected before it finishes charging. That is, if $\widehat{c_{ij}}$ denotes the time when $v_{ij}$ finishes charging, it must be the case that:

$$\forall v_{ij}, \quad \widehat{c_{ij}} = \widehat{st_{ij}} + \widehat{p_{ij}} \tag{10}$$

*Maximum active charging points per line.* The number of charging points that are active on a line at any instant cannot exceed a given threshold $N$. For any vehicle $v_{ij}$, its charging point is active in the time span between $\widehat{st_{ij}}$ and $\widehat{c_{ik}}$. We thus define the set of EVs that may be charging on the same line when $v_{ij}$ starts charging as:

$$A_{ij} = \{v_{ik} : \exists h \in \{1, 2, 3\}\ st^h_{ik} \leq st^h_{ij}, \max(\widehat{c_{ik}}, \widehat{st_{ij}}) \neq \widehat{st_{ij}}\}, \tag{11}$$

and $|A_{ij}|$ denotes its cardinal, then the constraint can be expressed as follows:

$$\forall v_{ij}, \quad |A_{ij}| \leq N. \tag{12}$$

*Balance between lines.* The load of active charging points must be evenly distributed among lines. Originally, for the deterministic problem, if $N_i(t)$ denotes the number of active charging points on line $L_i$, Hernandez-Arauzo et al. (2015) modelled this constraint using a threshold $\Delta \in [0, 1]$ representing the maximum imbalance between any two different lines $L_i$ and $L_j$, so that:

$$\forall t \geq 0, \max_{1 \leq i,j \leq 3} \left( \frac{|N_i(t) - N_j(t)|}{N} \right) \leq \Delta. \tag{13}$$

In a deterministic setting, we consider that a vehicle $v_{ij}$ is charging at instant $t$ if $st_{ij} \leq t \leq c_{ij}$. We extend this concept to the fuzzy setting by taking $t$ as a

TFN $\widehat{t} = (t, t, t)$, so the number of (expected) active charging points at instant $t \geq 0$ on line $L_i$, $1 \leq i \leq 3$, is defined as:

$$N_i(t) = \sum_{j=1}^{M_i} \delta_{ij}(t) \tag{14}$$

where

$$\delta_{ij}(t) = \begin{cases} 1 & : \text{ if } \widehat{st_{ij}} \leq_E \widehat{t} \leq_E \widehat{c_{ij}} \\ 0 & : \text{ otherwise.} \end{cases} \tag{15}$$

Thus, this constraint refers now to the expected load balance (since $N_i(t)$ is defined in terms of expected values).

### *3.1.2. Goal*

The *goal* is to find a feasible solution minimising the total tardiness with respect to departure dates, defined as follows:

$$\widehat{T} = \sum_{i=1}^{3} \sum_{j=1}^{M_i} \max(0, \widehat{c_{ij}} - d_{ij}) \tag{16}$$

Since the objective value $\widehat{T}$ is a TFN, solutions will be compared using $\leq_E$.

## 4. Genetic Algorithm

Genetic algorithms or GAs in short are metaheuristic methods capable of efficiently exploring large search spaces. They also allow to incorporate heuristic knowledge on the problem domain. Thus, they constitute solving methods for scheduling problems that yield good quality solutions with reasonable computational effort (Talbi, 2009).

Here, we propose a GA based on that proposed by García-Álvarez et al. (2015) for the deterministic version of the EV charging problem. This GA starts from an initial population composed by a mixture of random and heuristic individuals. It then evolves the initial population applying genetic operators until a stopping criterion is met, more precisely, until a solution with null total tardiness $\widehat{T} = 0$ is found or after a given number $max_{iter}$ of consecutive iterations have passed without improving the fitness of the best individual in the population. The pseudocode for this GA can be seen in Algorithm 1.

The GA uses an indirect encoding, so in order to build a solution it makes use of a purpose-built schedule generation scheme that schedules the EV charging according to the information coded in the chromosome. This decoding operator has been newly designed for the problem at hand, while the remaining operators have been adapted from the existing GA, taking into account the differences between the deterministic and the fuzzy setting, as explained in the following subsections.

```
Input  A problem instance $P$
Output  A schedule $S$ for $P$
  Generate the initial population;
  while Termination criterion is not satisfied do
    Group all chromosomes randomly in pairs;
    Apply crossover operator to each pair with probability $p_X$;
    Apply mutation operator to every offspring with probability $p_M$;
    Evaluate new chromosomes;
    Apply replacement strategy to pass chromosomes onto the next generation;
  end while
  return  The schedule $S$ from the best chromosome;
```

Algorithm 1: The genetic algorithm

### *4.1. Chromosome representation and evaluation*

Chromosomes code solutions as a permutation of the set of EVs. To build a charging schedule $S$ from a permutation $V$, we need a schedule-generation algorithm that assigns charging starting times $\widehat{st_{ij}}$ to all EVs based on the priority vector represented by $V$. To this end, we propose here a schedule generation scheme that takes into account the constraints and characteristics of the fuzzy problem. The rationale is to sequentially assign to each EV the earliest possible starting time such that the problem constraints as described in Section 3.1 hold with respect to the already existing partial schedule.

The detailed procedure can be seen in Algorithm 2, where fuzz$(\widehat{c}, e)$ denotes the TFN built from $\widehat{c}$ and $e$ according to Theorem 1. This scheme divides every line $L_i$ into $N$ virtual sublines $l_i^k$, $1 \leq k \leq N$, where $N$ is the maximum number of active charging points allowed per line. By doing so, the constraint in (12) regarding the maximum number of active charging points holds immediately. Then, charging starting times for all vehicles are assigned in the order in which they appear in the input vector $V$. For a particular EV $v_{ij}$, its earliest possible starting time on each subline $l_i^k$, denoted $\widehat{est}^k$, is computed as the maximum between its arrival time $t_{ij}$ and the time $\widehat{c_i}^k$ when the last EV already scheduled on that subline finishes charging. This guarantees that the first constraint in (9) holds. The line with the smallest earliest possible starting time (according to $\leq_E$) is selected as a candidate to schedule $v_{ij}$ on that subline. However, before a starting time can be assigned to $v_{ij}$, we need to ensure that the balance between lines, as expressed in (13), holds with respect to the already scheduled EVs for all the charging time $\widehat{p_{ij}}$. This might imply delaying the start of charging of $v_{ij}$ so its expected starting time coincides with an instant $st_{ij}^E$ posterior to the expected value of its earliest possible starting time, with a big gap in between. The schedule building scheme thus tries to find another subline where keeping the balance generates the least idleness. It is only after this step that the subline $k^*$ where $v_{ij}$ should be charged is selected. The starting time $\widehat{st_{ij}}$ is then established as the fuzzy time that can be obtained from the earliest possible starting time

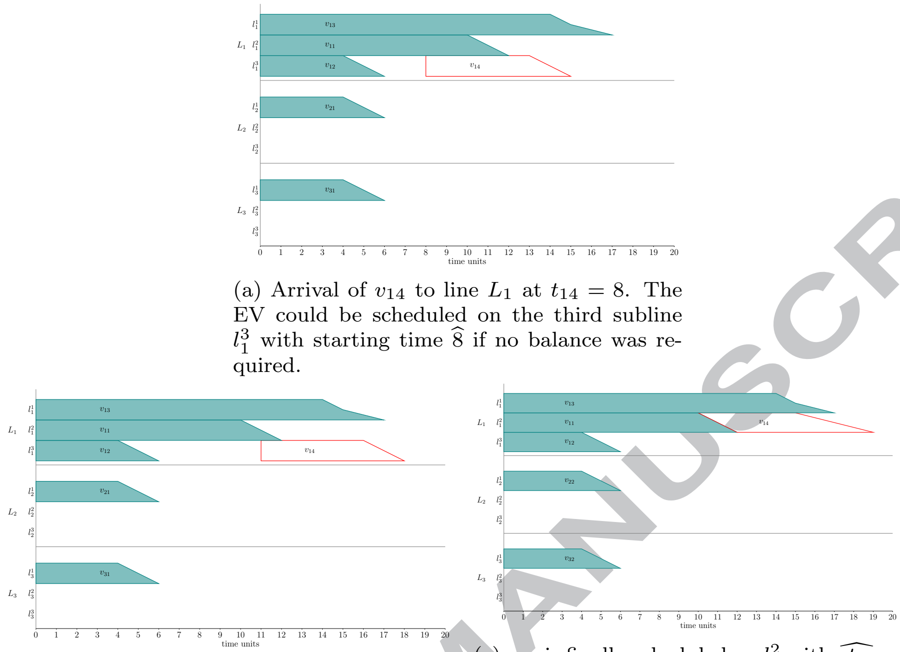
```
Input  A permutation $V$ of the EVs
Output  A schedule $S$ of charging starting times for all EVs
  Consider every line $L_i$ divided into $N$ sublines $l_i^k : k = 1 \dots N$;
  Initialise completion charging time on each subline $\widehat{c_i}^k \leftarrow 0$
  for all vehicle $v_{ij} \in V$ do
    //Select subline where $v_{ij}$ can start charging the earliest
    $\widehat{est}^k \leftarrow \max(\widehat{c_i}^k, t_{ij})$, $k = 1, \dots, N$
    $k^* \leftarrow \arg\min\{E[\widehat{est}^k]\}$
    //Find interval from $st_{ij}^E$ with balance between lines
    $st_{ij}^E \leftarrow \min\{t \geq E[\widehat{est}^{k^*}] : (13)$ holds in the interval $[t, t + \widehat{p_{ij}})\}$
    //Move to the subline where keeping balance generates least idleness
    $k^* \leftarrow \arg\max\{E[\widehat{est}^k] : E[\widehat{est}^k] \leq st_{ij}^E\}$
    //Assign charging starting time and update subline completion times
    $\widehat{st_{ij}} \leftarrow \mathrm{fuzz}(\widehat{est}^{k^*}, st_{ij}^E)$
    $\widehat{c_i}^{k^*} \leftarrow \widehat{st_{ij}} + \widehat{p_{ij}}$
  end for
  return  The schedule $S$
```

Algorithm 2: Schedule generation scheme

$\widehat{est}^{k^*}$ on that subline and the earliest instant where balance holds $st_{ij}^E$, according to Theorem 1. Notice that, according to that result, $E[\widehat{st_{ij}}] = st_{ij}^E$ (so balance holds) and $\widehat{st_{ij}} = \max(\widehat{st_{ij}}, \widehat{est}^{k^*})$ (so the charging time is posterior to the EV's arrival and the constraint regarding maximum number of active charging points holds as well).

Let us illustrate this schedule-generation procedure with a toy problem where a total of 6 EVs must be scheduled: four EVs $\{v_{11}, v_{12}, v_{13}, v_{14}\}$ on line $L_1$, one EV $\{v_{21}\}$ on $L_2$ and one EV $\{v_{31}\}$ on $L_3$. The arrival time is 0 for every EV except for $v_{14}$, with $t_{14} = 8$. The charging times are given by $\widehat{p_{12}} = \widehat{p_{21}} = \widehat{p_{31}} = (4, 5, 6)$, $\widehat{p_{11}} = (10, 11, 12)$, $\widehat{p_{13}} = (14, 15, 17)$, and $\widehat{p_{14}} = (5, 6, 7)$, and departure times are $d_{12} = d_{21} = d_{31} = 9$, $d_{11} = 11$, and $d_{13} = d_{14} = 20$. The maximum number of active charging points on every line is $N = 3$ and the maximum imbalance threshold is $\Delta = \frac{2}{3}$, so at every instant there should be at most an expected load difference of 2 EVs between any two lines.

Figure 2 illustrates the scheduling procedure for the situation where the algorithm's input is the permutation $(v_{11}, v_{21}, v_{31}, v_{12}, v_{13}, v_{14})$ and all EVs have been scheduled except for the last one $v_{14}$. In order to schedule it, we search for the subline $k$ in $L_1$ where $v_{14}$ could start to charge the earliest, which turns out to be $l_1^3$, with $\widehat{est}^3 = (8, 8, 8)$ (Figure 2(a)). Then, we compute the first instant $st_{ij}^E \geq 8$ after which balance between lines holds for the whole charging time of $v_{14}$, yielding $st_{ij}^E = 11$ (Figure 2(b)). We then notice that $v_{14}$ can be scheduled from that instant on a different subline $l_1^2$ (Figure 2(c)). Hence, the value for



(a) Arrival of $v_{14}$ to line $L_1$ at $t_{14} = 8$. The EV could be scheduled on the third subline $l_1^3$ with starting time $\widehat{8}$ if no balance was required.

(b) Balance holds only after $st_{14}^E = 11$

(c) $v_{14}$ is finally scheduled on $l_1^2$ with $\widehat{st_{14}} = (10, 11, 12)$

Figure 2: Example of schedule generation

$\widehat{st_{14}}$ is obtained from $st_{14}^E = 11$ in such a way that the EV can start charging after the earliest starting time on that subline $\widehat{est}^2$, that is, $\widehat{st_{14}} = (10, 11, 12)$.

### 4.2. Genetic operators

Except for the above decoding algorithm, the rest of the GA follows closely the one proposed by García-Álvarez et al. (2015) for the deterministic version of the problem. Let us briefly describe its main characteristics.

To generate an *initial population* both diverse and containing good individuals, one third of the individuals are generated as random permutations and each one of the other two thirds is generated by applying a stochastic version of a different dispatching rule. The first rule, DDR (Due-Date Rule) orders EVs in ascending order according to their departure time $d_{ij}$. The second rule, EST (Earliest Starting Time) orders EVs in ascending order according to their arrival time $t_{ij}$. The stochastic versions of both rules select the next EV to be added to the chromosome by applying tournament to *tourn* randomly selected individuals, where *tourn* is a parameter of the algorithm.

In the *selection phase*, all chromosomes are randomly paired. Then, crossover is applied (with probability $p_X$) to each pair in order to obtain two offspring, which may be subject to mutation with probability $p_M$. The *replacement strategy* selects from each set of parents and offspring the best two individuals with different fitness (i.e. expected tardiness) values. The goal is to maintain diversity in the population. If both parents and both offspring all have the same tardiness, then two of them are selected at random. Notice that this replacement strategy has an elitism effect, since the best individual in a generation is always equal to or better than the best individual in the previous generation.

Both crossover and mutation operators are those from (García-Álvarez et al., 2015), adapted to handling fuzzy times. The *crossover operator* SBX (Starting-time Based Crossover) was designed so offspring are likely to improve the expected tardiness value of their parents. The operator randomly selects an instant $t_0$, so the first offspring is generated by selecting from the first parent all EVs that are expected to start charging before $t_0$, while the remaining ones are rearranged according to the second parent. The second offspring is created similarly, with the first and second parent exchanging roles, that is, those EVs expected to start earlier than $t_0$ in the second parent pass onto the offspring in the same relative order and the remaining EVs are re-ordered following the first parent.

Operator SBX is illustrated in Figure 3. Provided that the randomly selected instant is $t_0 = 9$, to generate the first offspring $O_1$ we select from $P_1$ all EVs that are expected to start charging before $t_0$, that is, those $v_{ij}$ such that $E[\widehat{s}_{ij}] < t_0$: $3, 7, 1, 8, 2, 4, 10$ and $12$. These EVs are inserted in $O_1$ preserving their relative order. The chromosome is later completed with the remaining EVs (those expected to start after $t_0$ in $P_1$) in the same relative order as they appear in $P_2$, that is, $6, 5, 11, 9$. The second offspring is obtained similarly.

**Parents**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{P_1}$ | 3 | 7 | 1 | 8 | 2 | 4 | 6 | 5 | 10 | 9 | 12 | 11 |
| $\mathbf{P_2}$ | 8 | 6 | 5 | 2 | 11 | 9 | 7 | 3 | 10 | 12 | 1 | 4 |

*Expected charge starting times*

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{E[S]_1}$ | 0 | 0 | 0 | 0 | 0 | 7 | 10 | 10 | 0 | 10 | 7 | 11 |
| $\mathbf{E[S]_2}$ | 0 | 10 | 0 | 0 | 7 | 0 | 0 | 5 | 9 | 12 | 0 | 7 |

**Offspring**

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $\mathbf{O_1}$ | 3 | 7 | 1 | 8 | 2 | 4 | 10 | 12 | **6** | **5** | **11** | **9** |
| $\mathbf{O_2}$ | 8 | 5 | 2 | 11 | 9 | 7 | 3 | 1 | 4 | **6** | **10** | **12** |

Figure 3: Example of crossover SBX. We highlight in red those vehicles with expected charge starting time (also in red below) greater than or equal to $t_0 = 9$ .

The *mutation operator* mutates each line $L_i$ with probability $p_M$. If line $L_i$ is mutated, a random subset of consecutive EVs is selected on that line and this set is then shuffled, re-ordering these EVs. Figure 4 shows an example of mutation, supposing that only line $L_2$ is to be mutated. The random subset of consecutive EVs on that line that is selected is $2, 6, 12$. These are shuffled and reinserted in the original chromosome in the same positions occupied by this subset, but with the new relative order.

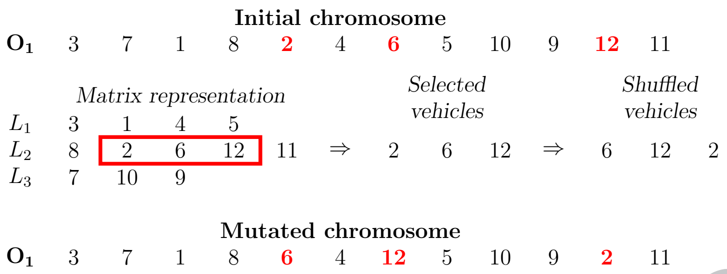


Figure 4: Example of mutation, where a random subset of consecutive EVs on line 2 (highlighted in red) is shuffled.

## 5. Experimental results

We now proceed to empirically evaluate the proposed method, which has been implemented in C++ and run on a PC with a Xeon E5520 at 2.2Ghz processor and 24Gb RAM running Linux (SL 6.0).

As benchmark instances, we take fuzzified versions of the problem instances used in Hernandez-Arauzo et al. (2015), which are based on real data. We simulate uncertainty in charging times using the fuzzification method from Ghrayeb (2003). In all cases, the charging station is supposed to be installed in a car park with 180 parking spaces. The planning horizon is 1 day and there are different profiles of arrival, charging and departure times based on real data. Instances are divided in two groups: instances of *Type 1*, the vehicles are uniformly distributed across the three lines, and instances of *Type 2*, where the proportion of EVs charging on lines $L_1$ to $L_3$ is 10% , 30% and 60% respectively. Clearly, situations of imbalance between lines are more likely to arise on instances of *Type 2*, something that the algorithm must take into account to build feasible schedules. Finally, we also consider different values for the imbalance parameter $\Delta \in \{0.2, 0.4, 0.6, 0.8\}$ and for the maximum charging points on a line $N \in \{20, 30, 40\}$. In total there are 24 instances, one per combination of *Type*, $\Delta$ and $N$ values.

After an initial parametric study, the algorithm parameters have been fixed as follows: population size is 200 individuals, the stopping parameter of maximum number of non-improving iterations is set to $max_{iter} = 25$, crossover and mutation probabilities are $p_X = 0.8$ and $p_M = 0.1$ respectively and tournament size for the heuristic initial population is set to $tourn = 8$. All experiments correspond to 30 runs of the GA on each instance, in order to obtain statistically significant results.

### *5.1. Algorithm's convergence*

First, we analyse the GA's evolution. To do so, we compare its results with a heuristic algorithm without evolution that generates as many groups of 200 individuals as generations takes for the GA to converge. Each group is formed by random individuals and individuals generated with the stochastic versions of rules DDR and EST in the same proportion as in the initial population of the GA. In order to make comparisons as fair as possible, this heuristic algorithm applies elitism by always keeping the best solution found so far.

Figure 5 illustrates the convergence of the GA for the instance of *Type 1* with $\Delta = 0.2$ and $N = 20$. This instance, the most constrained one of *Type 1*, has been selected as representative of the remaining ones, as the GA and the heuristic algorithm present a similar evolution pattern on all of them. The plot corresponds to the evolution of the average population fitness (i.e., the average of the expected total tardiness $E[\widehat{T}]$) across 30 runs both of the GA and the heuristic algorithm. It is clear that the improvement with respect to the best initial solution (the same for both methods) is greater with the GA (45.15% of improvement) than with the heuristic algorithm (12.86%).

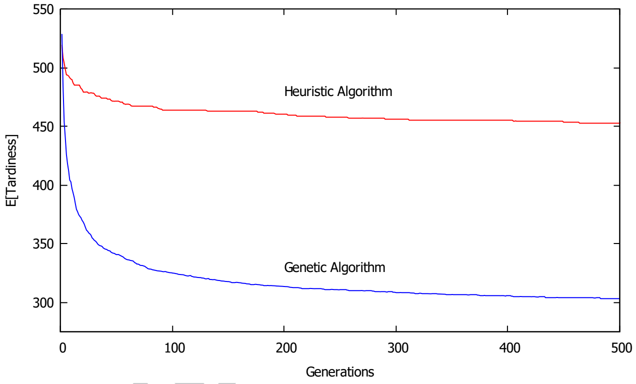


Figure 5: Average fitness evolution for the GA (in blue) and a heuristic algorithm (in red) on the instance of *Type 1* with $N = 20$ and $\Delta = 0.2$.

### 5.2. *Algorithm's performance*

To evaluate the GA's performance, since the problem formulation is new to this work and, hence, no other algorithms from the literature exist, we compare the results obtained by the GA with those obtained by the described heuristic algorithm, which essentially consists in applying random or heuristic scheduling rules. The results are shown in Tables 1 to 4, and they correspond to both algorithms running under equivalent conditions (equal number of evaluated individuals). Each row of each table corresponds to one of the benchmark instances, as indicated in the first two columns. For each instance, Tables 1 and 2 give the fuzzy tardiness $\widehat{T}$ together with its expected value $E[\widehat{T}]$ of the best solution found with each algorithm (columns 3,4 for the GA and 5,6 for the heuristic method) for instances of *Type 1* and *Type 2* respectively. All tardiness values are measured in hours. The last column corresponds to the improvement (in percentage) of the GA with respect to the Heuristic. We can see that, on *Type 1* instances, this improvement ranges between 6.2% and slightly more than 69%, with an average improvement near 33%. In fact, an improvement below

Table 1: Best total tardiness values for the GA and the heuristic algorithm on instances of *Type 1*.

| Instance | | GA | | Heuristic | | |
|---|---|---|---|---|---|---|
| $N$ | $\Delta$ | Best $\widehat{T}$ | $E[\widehat{T}]$ | Best $\widehat{T}$ | $E[\widehat{T}]$ | % Impr. |
| 20 | 0.2 | (197.7, 278.3, 430.9) | 296.3 | (335.9, 407.3, 545.7) | 424.1 | 30.12 |
| | 0.4 | (77.35, 126.3, 244.8) | 143.7 | (114.1, 165.2, 282.8) | 181.8 | 20.97 |
| | 0.6 | (62.65, 109.9, 219.8) | 125.5 | (72.13, 118.6, 227.1) | 134.1 | 6.38 |
| | 0.8 | (61.41, 109.6, 219.7) | 125.1 | (72.14, 118.6, 223.9) | 133.3 | 6.19 |
| 30 | 0.2 | (28.89, 49.37, 102.7) | 57.61 | (81.29, 110.3, 177.6) | 119.9 | 51.96 |
| | 0.4 | (1.300, 8.333, 43.84) | 15.45 | (8.342, 17.43, 60.75) | 25.99 | 40.54 |
| | 0.6 | (0.250, 6.616, 42.18) | 13.91 | (5.483, 12.96, 54.18) | 21.39 | 34.97 |
| | 0.8 | (0.716, 6.633, 41.64) | 13.90 | (5.483, 12.96, 54.18) | 21.39 | 35.01 |
| 40 | 0.2 | (3.013, 8.470, 27.19) | 11.78 | (22.83, 34.46, 60.54) | 38.07 | 69.05 |
| | 0.4 | (0, 0, 0) | 0.00 | (0, 0, 0) | 0.00 | - |
| | 0.6 | (0, 0, 0) | 0.00 | (0, 0, 0) | 0.00 | - |
| | 0.8 | (0, 0, 0) | 0.00 | (0, 0, 0) | 0.00 | - |

10% only takes place for two instances and the average improvement across the remaining instances is in fact greater than 40%. On instances of *Type 2* the improvement of the best solutions is more regular, with an average value of 21.42%. Tables 3 and 4 correspond to the average performance on instances of *Type 1* and *Type 2* respectively. They contain the average expected tardiness of the solution across 30 runs of the GA and the heuristic method together with the standard deviation and the average CPU time (in seconds). The last column reports the average improvement percentage values of the GA with respect to the heuristic method. We can see that, regarding the average expected tardiness, the GA is in average 40.52% better than heuristic algorithm on instances of *Type 1* and 21.38% better on instances of *Type 2* (where the balance between lines becomes more critical), presenting with average values a similar behaviour to the best solution.

Additionally, to assess the difficulty of finding a solution to this problem with exact methods, we have provided an alternative integer linear programming (ILP) formulation (see Appendix A) which allows to run IBM ILOG CPLEX Optimizer (IBM, 2014) on small instances of the problem. As expected, Table 1 shows that the easiest instances are those of *Type 1* with largest values for $N$ and $\Delta$; in fact, instances of *Type1* with $N = 40$ and $\Delta = 0.8$ are solved to optimality by both the GA and the heuristic algorithm. Therefore, we have generated new smaller instances of *Type 1* by "pruning" the existing ones, so only the first 20 EVs out of the 180 original ones are considered, the balance parameter $\Delta = 0.8$ remains unchanged and $N$ is rescaled to $N = 5$, which is proportionally less strict than $N = 40$ for 180 EVs. For these very simple instances, IBM ILOG CPLEX Optimizer takes 12 hours to provide solutions with an optimality gap of 50%. For slightly larger instances with 30 EVs, $N = 7$ and $\Delta = 0.8$ the same solver cannot provide a feasible solution after running for 12 hours.

To complement the data on the tables, we have run some statistical tests.

Table 2: Best total tardiness values for the GA and the heuristic algorithm on instances of *Type 2*.

| Instance | | GA | | Heuristic | | |
|---|---|---|---|---|---|---|
| $N$ | $\Delta$ | Best $\widehat{T}$ | $E[\widehat{T}]$ | Best $\widehat{T}$ | $E[\widehat{T}]$ | % Impr. |
| 20 | 0.2 | (4481, 4644, 4896) | 4666 | (5350, 5514, 5757) | 5534 | 15.67 |
| | 0.4 | (1560, 1707, 1949) | 1731 | (1961, 2117, 2350) | 2137 | 19.00 |
| | 0.6 | (701.9, 836.1, 1056) | 857.7 | (935.2, 1037, 1210) | 1054 | 18.70 |
| | 0.8 | (372.4, 489.9, 691.1) | 510.8 | (487.0, 592.4, 778.7) | 612.6 | 16.61 |
| 30 | 0.2 | (2473, 2667, 2967) | 2694 | (3019, 3180, 3418) | 3199 | 15.81 |
| | 0.4 | (681.2, 804.5, 1000) | 822.7 | (909.9, 1007, 1174) | 1024 | 19.71 |
| | 0.6 | (177.5, 252.2, 394.9) | 269.2 | (270.4, 341.1, 478.9) | 357.9 | 24.78 |
| | 0.8 | (43.95, 81.29, 182.4) | 97.24 | (68.12, 114.8, 216.8) | 128.6 | 24.43 |
| 40 | 0.2 | (1564, 1705, 1931) | 1726 | (1961, 2117, 2350) | 2137 | 19.20 |
| | 0.4 | (302.6, 385.0, 531.6) | 401.0 | (412.5, 501.3, 652.6) | 516.9 | 22.41 |
| | 0.6 | (28.84, 64.54, 153.3) | 77.81 | (58.15, 100.2, 187.7) | 111.6 | 30.29 |
| | 0.8 | (0.299, 6.307, 57.50) | 17.60 | (3.506, 15.04, 67.64) | 25.30 | 30.43 |

First, we have run a Wilcoxon signed-rank test for matched samples on the average expected tardiness values obtained by each method across 30 runs on each of the 24 instances. The test confirms that the GA outperforms the heuristic algorithm with a $p$-value $9.702 \times 10^{-6}$.

Secondly, we have performed a more detailed analysis on an instance of each type. We have chosen in each case that instance for which the average percentage improvement of the GA with respect to the heuristic algorithm is closest to the average value across all instances of that type, that is, instance of *Type 1* with $N = 30$ and $\Delta = 0.4$ and instance of *Type 2* with $N = 40$ and $\Delta = 0.4$. For each instance, we have taken the expected tardiness results obtained on the 30 runs of each algorithm and, after checking normality with a Kolmogorov-Smirnoff test, we have run a t-test to compare both algorithms leading to the same conclusion as above with $p$-values, $3.85 \times 10^{-65}$ for *Type 1* and $3.35 \times 10^{-69}$ for *Type 2*. The corresponding boxplots for both instances, in Figure 6, allow to visualise the superiority of the GA.

Finally, let us notice that the tardiness values on all tables reflect how the difficulty of an instance depends on the (un)even load of EVs on each line $L_i$ and on the values of the parameters $N$ and $\Delta$. As expected, it is easier to find good solutions (with low total tardiness) for instances of *Type 1*, where the load on all lines is even, than on problems of *Type 2*, with uneven loads. It is also clear that problems with a low number of active charging points per line $N$ and strong balance restriction, represented by low values of $\Delta$, are harder to solve. In fact, at the extreme case we find two pathological instances, both with very uneven loads, a small proportion of active charging points per line and under a very strict load balance constraint. These are two instances of *Type 2*, where the first line $L_1$ supports 60% of the demand, with 108 EVs, 30% (54 EVs) of the demand falls on the second line $L_2$ and only 10% (18 EVs) falls on the third line $L_3$. Additionally, these two instances have a small threshold for the

Table 3: Average expected tardiness of the GA and the heuristic algorithm on instances of *Type 1*.

| Instance | | GA | | Heuristic | | |
|---|---|---|---|---|---|---|
| $N$ | $\Delta$ | Avg.±SD | CPU(s) | Avg.±SD | CPU(s) | % Impr. |
| 20 | 0.2 | 336.0±12.17 | 40 | 458.5±7.17 | 194 | 29.91 |
| | 0.4 | 149.4±1.39 | 50 | 192.3±2.30 | 199 | 21.50 |
| | 0.6 | 129.3±0.91 | 51 | 137.6±0.66 | 201 | 7.00 |
| | 0.8 | 127.1±0.46 | 55 | 134.9±0.40 | 201 | 6.19 |
| 30 | 0.2 | 77.74±4.74 | 54 | 144.8±5.13 | 199 | 52.61 |
| | 0.4 | 17.33±0.46 | 35 | 27.29±0.36 | 206 | 38.51 |
| | 0.6 | 17.21±0.71 | 29 | 23.08±0.36 | 206 | 34.04 |
| | 0.8 | 16.66±0.60 | 30 | 23.08±0.36 | 206 | 34.73 |
| 40 | 0.2 | 19.41±2.09 | 44 | 45.26±2.18 | 202 | 64.00 |
| | 0.4 | 0.091±0.03 | 9 | 0.137±0.03 | 206 | 69.03 |
| | 0.6 | 0.091±0.04 | 8 | 0.133±0.03 | 206 | 64.33 |
| | 0.8 | 0.091±0.04 | 8 | 0.133±0.03 | 206 | 64.33 |

maximum number of active charging points $N = 20$ and $N = 30$ (respectively, 33% and 50% of the 60 points available in each line) and a very strict balance constraint with maximum expected imbalance threshold $\Delta = 0.2$. As a result, once the EVs (only 18) that go to the third line $L_3$ finish charging, the number of EVs that can be expected to be charging at any instant on any of the other two lines is limited to only 4 or 6 (with $N = 20$ and $N = 30$ respectively). This creates a bottleneck and the subsequent delays on the first and second lines, this being specially problematic for $L_1$, where a total of 90 EVs must be charged. As mentioned above, these instances are pathological and cannot be expected to be solved to satisfaction. They pose instead extreme cases that serve as a challenge for any solving methods that may be proposed for the problem at hand.

## 6. Conclusions

We have tackled the problem of scheduling the charging of EVs on a real-world charging station. The schedule must take into account some technical constraints of the system, while simultaneously minimise a total tardiness objective function. This work constitutes a first approach to incorporating uncertainty to the problem, to narrow the gap between the model and the real-world situation. In particular, we have considered uncertain charging times modelled as triangular fuzzy numbers. This has resulted in a new formulation of the problem, based on the deterministic one, but with considerable changes, for instance, in the translation of the constraints. Additionally, to solve the resulting problem we have proposed a GA based on a previous algorithm with good performance in the deterministic version. Although the general scheme remains the same, some operators, such as crossover, have had to be adapted to handle fuzzy numbers and a new decoding operator has been proposed, consisting on a purpose-built schedule builder. Finally, we have presented an experimental

Table 4: Average expected tardiness of the GA and the heuristic algorithm on instances of *Type 2*.

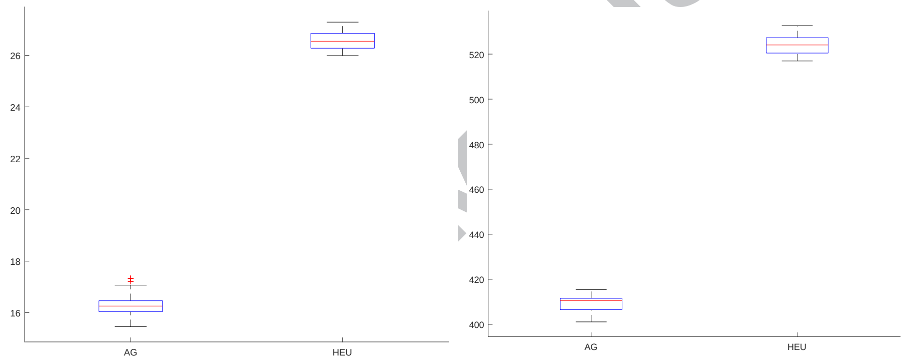
| Instance | | GA | | Heuristic | | |
|---|---|---|---|---|---|---|
| $N$ | $\Delta$ | Avg.±SD | CPU(s) | Avg.±SD | CPU(s) | % Impr. |
| 20 | 0.2 | 4777±24.00 | 186 | 5658±30.41 | 194 | 15.60 |
| | 0.4 | 1801±14.58 | 145 | 2203±15.55 | 198 | 19.10 |
| | 0.6 | 896.7±8.47 | 111 | 1085±6.93 | 200 | 18.66 |
| | 0.8 | 528.5±4.39 | 88 | 631.5±4.78 | 201 | 16.56 |
| 30 | 0.2 | 2772±20.60 | 173 | 3343±29.70 | 196 | 16.95 |
| | 0.4 | 859.9±9.24 | 122 | 1055±6.43 | 202 | 19.44 |
| | 0.6 | 279.6±2.53 | 54 | 370.8±3.30 | 207 | 24.85 |
| | 0.8 | 104.9±1.97 | 40 | 136.5±1.87 | 208 | 23.50 |
| 40 | 0.2 | 1794±15.12 | 148 | 2186±12.29 | 198 | 18.54 |
| | 0.4 | 415.4±3.69 | 92 | 532.6±4.17 | 205 | 21.87 |
| | 0.6 | 83.86±1.57 | 42 | 118.3±1.65 | 210 | 30.63 |
| | 0.8 | 20.74±0.69 | 40 | 28.19±0.62 | 210 | 30.80 |

study on new benchmark fuzzy instances inspired in real-world data that show the correct convergence of the GA as well as its good performance. Additionally, the benchmark instances will be made openly available to the research community on the web, to encourage future advancements on solving this problem.

This constitutes the first approach to the charging problem with uncertainty and as such opens many lines for future research. First, we would like to establish links between the constraint formulations and possibility theory which, in turn, may suggest new and better schedule-building algorithms. It is also possible to give alternative formulations for some constraints in the uncertain setting, in particular, for the balance between lines. Regarding the solving method, it could be improved by combining the GA with local search, which would require defining good neighbourhood structures and efficient neighbour evaluation methods. Another interesting perspective, as considered in Burdett and Kozan (2015), is to consider the robustness of the obtained solutions, either as an objective function to optimise, either as a criterion to compare the fuzzy solutions and the deterministic ones. Finally, we aim at incorporating new features that make the problem closer to the real-life situation, such as having variable energy costs depending on the time of the day.

## Acknowledgements

This work was supported by the Spanish Government grant number TIN2016-79190-R.



*Type 1*, $N = 30$ , $\Delta = 0.4$ *Type 2*, $N = 40$ , $\Delta = 0.4$

Figure 6: Boxplot for the results of 30 runs of GA and the heuristic algorithm on two representative instances.

## Appendix A. ILP formulation

It is possible to model our problem using an alternative ILP formulation, using the notation summarised in Table A.5, with auxiliary decision variables $\alpha_{ijt}, \beta_{ijt}$ and $\delta_{ijt}$ to model the constraint of balance between lines represented in (15), and $\gamma_{ijk}$, $\rho_{ijk}$ and $\epsilon_{ijk}$ to model the set of overlapping vehicles defined in (11), used in the constraint of maximum active charging points per line. For the experimental results, the time horizon $|H|$ has been calculated as $|H| = \sum_{i,j} p^3_{ij} + \max_{i,j}\{t_{ij}\}$, which provides an upper bound on the total completion time.

Table A.5: Notation used in the ILP formulation for problem parameters

| **Parameters** | |
|---|---|
| $i \in \{1, 2, 3\}$ | charging lines |
| $j \in M_i$ | number of vehicles arriving at each line |
| $l \in \{1, 2, 3\}$ | components of a TFN |
| $N$ | maximum number of charging points that can be active simultaneously |
| $v_{ij}$ | vehicle that arrives at $j^{th}$ order to line $i$ |
| $t_{ij}$ | arrival time of vehicle $v_{ij}$ |
| $\widehat{p_{ij}}$ | fuzzy processing time of vehicle $v_{ij}$ |
| $d_{ij}$ | departure time of the vehicle given by driver |
| $\Delta$ | maximum load imbalance between any two lines |
| $M$ | a sufficiently large number for disjunctive constraints<br>taken as $2 \times \|H\|$ in experiments |
| $t \in H$ | discrete time periods within a horizon $H$ |
| **Decision variables** | |
| $\widehat{T_{ij}} = (T^1_{ij}, T^2_{ij}, T^3_{ij})$ | fuzzy tardiness of vehicle $v_{ij}$, $T^l_{ij} \in \mathbb{Z}$ |
| $\widehat{st_{ij}} = (st^1_{ij}, st^2_{ij}, st^3_{ij})$ | fuzzy charging starting time for vehicle $v_{ij}$, $st^l_{ij} \in \mathbb{Z}$ |
| $\alpha_{ijt}$ | binary variable, equal to 1 if $v_{ij}$ is expected to have started charging before time $t$ |
| $\beta_{ijt}$ | binary variable, equal to 1 if $v_{ij}$ is expected to continue charging after time $t$ |
| $\delta_{ijt}$ | binary variable, equal to 1 if $v_{ij}$ is expected to be charging at time $t$ |
| $\gamma_{ijk}$ | binary variable, equal to 1 if $v_{ik}$ might have started charging before $v_{ij}$ starts charging |
| $\rho_{ijk}$ | binary variable, equal to 1 if $v_{ik}$ might finish charging after $v_{ij}$ starts charging |
| $\epsilon_{ijk}$ | binary variable, equal to 1 if $v_{ik}$ might be charging when $v_{ij}$ starts charging |

$$\min_{\widehat{st},\widehat{T},\alpha,\beta,\delta,\gamma,\rho,\epsilon} \sum_{i,j} \frac{T^1_{ij}+2T^2_{ij}+T^3_{ij}}{4}$$

s.t.

$$T^l_{ij} \geq st^l_{ij} + p^l_{ij} - d_{i,j} \qquad \forall i,j,l \tag{A.1}$$

$$t - \frac{(st^1_{ij}+2st^2_{ij}+st^3_{ij})}{4} + 1 \leq M\alpha_{ijt} \qquad \forall i,j,t \tag{A.2}$$

$$\frac{st^1_{ij}+2st^2_{ij}+st^3_{ij}}{4} - t \leq M(1-\alpha_{ijt}) \qquad \forall i,j,t \tag{A.3}$$

$$t - \frac{(st^1_{ij}+2st^2_{ij}+st^3_{ij}+p^1_{ij}+2p^2_{ij}+p^3_{ij})}{4} \leq M(1-\beta_{ijt}) \qquad \forall i,j,t \tag{A.4}$$

$$\frac{st^1_{ij}+2st^2_{ij}+st^3_{ij}+p^1_{ij}+2p^2_{ij}+p^3_{ij}}{4} - t + 1 \leq M\beta_{ijt} \qquad \forall i,j,t \tag{A.5}$$

$$\delta_{ijt} = \alpha_{ijt} + \beta_{ijt} - 1 \qquad \forall i,j,t \tag{A.6}$$

$$\sum_{j \in M_i} \delta_{ijt} - \sum_{j \in M_{i'}} \delta_{i'jt} \leq \Delta N \qquad \forall i,i' \in \{1,2,3\}, \forall t \tag{A.7}$$

$$st^1_{ij} \geq t_{ij} \qquad \forall i,j \tag{A.8}$$

$$st^l_{ij} - st^l_{ik} + 1 \leq M\gamma_{ijk} \qquad \forall i, \forall j,k \in M_i, \forall l \tag{A.9}$$

$$st^l_{ik} + p^l_{ik} - st^l_{ij} \leq M\rho_{ijk} \qquad \forall i, \forall j,k \in M_i, \forall l \tag{A.10}$$

$$\epsilon_{ijk} = \gamma_{ijk} + \rho_{ijk} - 1 \qquad \forall i, \forall j,k \in M_i \tag{A.11}$$

$$\sum_{k \in M_i} \epsilon_{ijk} \leq N \qquad \forall i,j \tag{A.12}$$

$$0 \leq st^1_{ij} \leq st^2_{ij} \leq st^3_{ij} \qquad \forall i,j \tag{A.13}$$

$$0 \leq T^1_{ij} \leq T^2_{ij} \leq T^3_{ij} \qquad \forall i,j \tag{A.14}$$

$$\alpha_{ijt}, \beta_{ijt}, \delta_{ijt} \in \{0,1\} \qquad \forall i,j,t \tag{A.15}$$

$$\gamma_{ijk}, \rho_{ijk}, \epsilon_{ijk} \in \{0,1\} \qquad \forall i, \forall j,k \in M_i \tag{A.16}$$

$$st^l_{ij}, T^l_{ij} \in \mathbb{Z} \qquad \forall i,j,l \tag{A.17}$$

Regarding the constraints in the ILP model, (A.1) corresponds to minimising the tardiness of each vehicle, as implicitly defined in (16). (A.2) and (A.3) ensure that $\alpha_{ijt}$ takes value 1 if time $t \geq E[\widehat{st_{ij}}]$ and 0 otherwise; similarly, (A.4) and (A.5) ensure that $\beta_{ijt}$ takes value 1 if time $t \leq E[\widehat{st_{ij}} + \widehat{p_{ij}}]$ (i.e. $t \leq E[\widehat{c_{ij}}]$) and 0 otherwise and (A.6) causes $\delta_{ijt}$ to be 1 if $\alpha_{ijt}$ and $\beta_{ijt}$ are both 1, and 0 otherwise (notice that it cannot be -1 since $\alpha$ and $\beta$ cannot be both 0 at the same time). In consequence, (A.7) ensures that the load imbalance between any two lines does not exceed $\Delta$ (constraint (13) in the original model). Constraint (A.8) is the same as (9) for the EV's arrival time. (A.9) ensures that $\gamma_{ijk}$ takes value

1 if vehicle $v_{ik}$ might have started charging before vehicle $v_{ij}$ and 0 otherwise, while (A.10) ensures that $\rho_{ijk}$ takes value 1 if vehicle $v_{ik}$ might be still charging when vehicle $v_{ij}$ starts charging and 0 otherwise. Thus, (A.11) causes $\epsilon_{ijk}$ to take value 1 if vehicles $v_{ik}$ and $v_{ij}$ might be charging simultaneously, so (A.12) ensures that there cannot be more than $N$ vehicles charging at the same time on the same line when vehicle $v_{ij}$ might start charging. This corresponds to the constraint of maximum active charging points per line modelled by (11) and (12). Finally, (A.13) and (A.14) ensure the right order of the three TFN components and nonnegativity, (A.15) and (A.16) model integrality constraints of binary variables and (A.17) states that $st^l_{ij}$ and $T^l_{ij}$ are all integer variables.

Abdullah, S., Abdolrazzagh-Nezhad, M., 2014. Fuzzy job-shop scheduling problems: A review. Information Sciences 278, 380–407. doi:10.1016/j.ins.2014.03.060.

Ansari, M., Al-Awami, A.T., Sortomme, E., Abido, M.A., 2015. Coordinated bidding of ancillary services for vehicle-to-grid using fuzzy optimization. IEEE Transactions on Smart Grid 6, 261–270. doi:10.1109/TSG.2014.2341625.

Balin, S., 2011. Parallel machine scheduling with fuzzy processing times using a robust genetic algorithm and simulation. Information Sciences 181, 3551–3569.

Baudrit, C., Dubois, D., 2006. Practical representations of incomplete probabilistic knowledge. Computational Statistics & Data Analysis 51, 86–108.

Brunelli, M., Mezei, J., 2013. How different are ranking methods for fuzzy numbers? A numerical study. International Journal of Approximate Reasoning 54, 627–639. doi:10.1016/j.ijar.2013.01.009.

Burdett, R., Kozan, E., 2015. Techniques to effectively buffer schedules in the face of uncertainties. Computers & Industrial Engineering 87, 16–29.

Chanas, S., Kasperski, A., 2001. Minimizing maximum lateness in a single machine scheduling problem with fuzzy processing times and fuzzy due dates. Engineering Applications of Artificial Intelligence 14, 377–386.

Chanas, S., Nowakowski, M., 1988. Single value simulation of fuzzy variable. Fuzzy Sets and Systems 25, 43–57.

Destercke, S., Couso, I., 2015. Ranking of fuzzy intervals seen through the imprecise probabilistic lens. Fuzzy Sets and Systems 278, 20–39. doi:10.1016/j.fss.2014.12.009.

Dubois, D., 2006. Possibility theory an statistical reasoning. Computational Statistics & Data Analysis 51, 47–69.

Dubois, D., 2011. The role of fuzzy sets in decision sciences: Old techniques and new directions. Fuzzy Sets and Systems 184, 3–28. doi:10.1016/j.fss.2011.06.003.

Dubois, D., Fargier, H., Fortemps, P., 2003. Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. European Journal of Operational Research 147, 231–252.

Dubois, D., Fargier, H., Fortemps, P., 2008. Scheduling under flexible constraints and uncertain data: the fuzzy approach, in: Production Scheduling. Wiley. chapter 11, pp. 301–332.

Dubois, D., Foulloy, L., Mauris, G., Prade, H., 2004. Probability-possibility transformations, triangular fuzzy sets and probabilistic inequalities. Reliable Computing 10, 273–297.

Dubois, D., Prade, H., 1986. Possibility Theory: An Approach to Computerized Processing of Uncertainty. Plenum Press, New York (USA).

Dubois, D., Prade, H., 1987. The mean value of a fuzzy number. Fuzzy Sets and Systems 24, 279–300.

Fortemps, P., 1997. Jobshop scheduling with imprecise durations: a fuzzy approach. IEEE Transactions of Fuzzy Systems 7, 557–569.

García-Álvarez, J., González, M., Vela, C., 2015. A genetic algorithm for scheduling electric vehicle charging, in: Proceedings of the Genetic and Evolutionay Computation Conference (GECCO 2015), pp. 393–400.

Ghrayeb, O.A., 2003. A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. Applied Soft Computing 2, 197–210.

Guo, S., Zhao, H., 2015. Optimal site selection of electric vehicle charging station by using fuzzy TOPSIS based on sustainability perspective. Applied Energy 158. doi:10.1016/j.apenergy.2015.08.082.

Hajforoosh, S., Masoum, M.A., Islam, S.M., 2015. Real-time charging coordination of plug-in electric vehicles based on hybrid fuzzy discrete particle swarm optimization. Electric Power Systems Research 128, 19–29. doi:10.1016/j.epsr.2015.06.019.

Hapke, M., Slowinski, R., 1996. Fuzzy priority heuristics for project scheduling. Fuzzy Sets and Systems 83, 291–299. doi:10.1016/0165-0114(95)00338-X.

Heilpern, S., 1992. The expected value of a fuzzy number. Fuzzy Sets and Systems 47, 81–86.

Hernandez-Arauzo, A., Puente, J., Varela, R., Sedano, J., 2015. Electric vehicle charging under power and balance constraints as dynamic scheduling. Computers & Industrial Engineering 85, 306–315. doi:10.1016/j.cie.2015.04.002.

IBM, 2014. IBM CPLEX Optimizer. URL: http://www-01.ibm.com/software/commerce/optimization/cplex-optimizer/index.html.

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. doi:10.1016/j.procs.2013.01.080.

Kuroda, M., Wang, Z., 1996. Fuzzy job shop scheduling. International Journal of Production Economics 44, 45–51.

Lei, D., 2010. Solving fuzzy job shop scheduling problems using random key genetic algorithm. International Journal of Advanced Manufacturing Technologies 49, 253–262.

Liu, B., Fan, Y., Liu, Y., 2015. A fast estimation of distribution algorithm for dynamic fuzzy flexible job-shop scheduling problem. Computers & Industrial Engineering 87, 193–201.

Mauris, G., Lasserre, V., Foulloy, L., 2001. A fuzzy approach for the expression of uncertainty in measurement. Measurement 29, 165–177. doi:10.1016/S0263-2241(00)00036-1.

Palacios, J.J., González, M.A., Vela, C.R., González-Rodríguez, I., Puente, J., 2015a. Genetic tabu search for the fuzzy flexible job shop problem. Computers & Operations Research 54, 74–89. doi:10.1016/j.cor.2014.08.023.

Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J., 2015b. Coevolutionary makespan optimisation through different ranking methods for the fuzzy flexible job shop. Fuzzy Sets and Systems 278, 81–97. doi:10.1016/j.fss.2014.12.003.

Palacios, J.J., González-Rodríguez, I., Vela, C.R., Puente, J., 2017. Robust multiobjective optimisation for fuzzy job shop problems. Applied Soft Computing 56, 604–616. doi:10.1016/j.asoc.2016.07.004.

Rahman, I., Vasant, P., Singh, B., Abdullah-Al-Wadud, M., Adnan, N., 2016. Review of recent trends in optimization techniques for plug-in hybrid, and electric vehicle charging infrastructures. Renewable and Sustainable Energy Reviews 58, 1039–1047. doi:10.1016/j.rser.2015.12.353.

Rommelfanger, H., 1990. FULPAL — an interactive method for solving (multiobjective) fuzzy linear programming problems, in: Slowinski, R., Teghem, J. (Eds.), Stochastic Versus Fuzzy Approaches to Multiobjective Mathematical Programming under Uncertainty. Springer, pp. 279–299. doi:10.1007/978-94-009-2111-5_14.

Sedano, J., Portal, M., Hernández-Arauzo, A., Villar, J.R., Puente, J., Varela, R., 2013. Intelligent system for electric vehicle charging: Design and operation. DYNA 88, 644–651. doi:10.6036/5788.

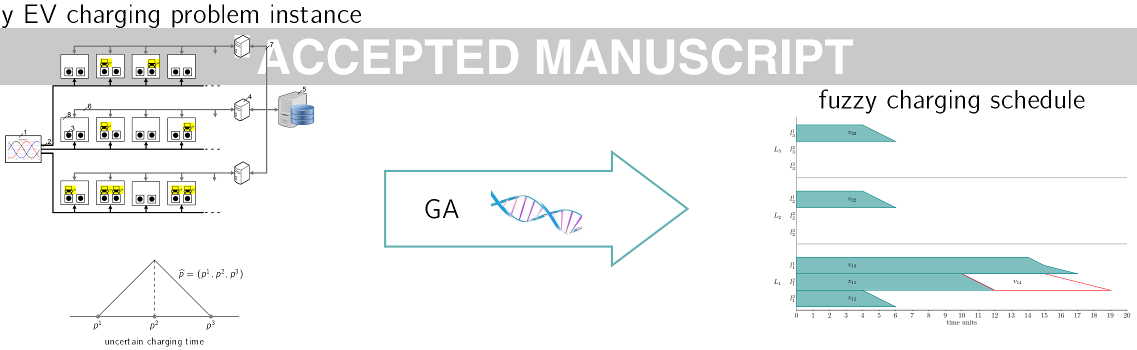
Talbi, E.G., 2009. Metaheuristics. From Design to Implementation. Wiley.

Wang, B., Li, Q., Yang, X., Wang, X., 2010. Robust and satisfactory job shop scheduling under fuzzy processing times and flexible due dates, in: Proc. of the 2010 IEEE International Conference on Automation and Logistics, pp. 575–580.

Wong, B.K., Lai, V.S., 2011. A survey of the application of fuzzy set theory in production and operations management: 1998–2009. International Journal of Production Economics 129, 157–168.

You, P.S., Hsieh, Y.C., 2014. A hybrid heuristic approach to the problem of the location of vehicle charging stations. Computers & Industrial Engineering 70, 195–204.

Zhong, S., Chen, Y., Zhou, J., 2015. Fuzzy random programming models for location-allocation problem with applications. Computers & Industrial Engineering 89, 194–202.


fuzzy EV charging problem instance
1
2
3
4
5
6
7
8
$\widehat{p} = (p^1, p^2, p^3)$
$p^1$
$p^2$
$p^3$
uncertain charging time
GA
fuzzy charging schedule
$L_3$
$l_3^1$
$l_3^2$
$l_3^3$
$v_{32}$
$L_2$
$l_2^1$
$l_2^2$
$l_2^3$
$v_{22}$
$L_1$
$l_1^1$
$l_1^2$
$l_1^3$
$v_{13}$
$v_{11}$
$v_{14}$
$v_{12}$
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20
time units

Highlights

- We tackle the problem of scheduling the charging of electric vehicles.
- We model a variant with uncertain charging times using fuzzy numbers.
- We propose a genetic algorithm with tailor-made operators.
- A new chromosome evaluation method generates schedules from a priority vector.
- We test and analyse the algorithm in terms of convergence and solution quality.