

Planificación Genética de la Carga de Vehículos Eléctricos bajo Incertidumbre

Jorge García-Alvarez¹, Inés González-Rodríguez², Miguel A. González¹, and Camino R. Vela¹

¹ Departamento de Informática, Universidad de Oviedo
jgarcia-alvarez@outlook.com, {mig, crvela}@uniovi.es,

² Departamento de Matemáticas, Estadística y Computación,
Universidad de Cantabria gonzalezri@unican.es

Resumen Este trabajo aborda un problema motivado por una estación real de carga de vehículos eléctricos. El objetivo es planificar la carga de un conjunto de vehículos atendiendo a las restricciones técnicas de la estación a la vez que se minimiza el retraso total de la carga con respecto a los instantes de retirada marcados por los propietarios de los vehículos. Modificamos la formalización existente del problema para incorporar incertidumbre en los tiempos de carga en forma de conjuntos difusos y proponemos resolverlo mediante un algoritmo genético con operadores específicos, destacando el algoritmo de evaluación de cromosomas. Realizamos un estudio experimental en el que mostramos el funcionamiento satisfactorio del algoritmo, tanto en convergencia como en calidad de los resultados.

Keywords: Vehículos eléctricos, fuzzy scheduling, algoritmos genéticos, heurísticos, optimización combinatoria

1. Introducción

La presencia de vehículos eléctricos (VEs) se ha incrementado notablemente en los últimos años, produciendo un impacto positivo en la economía y en el entorno. Los VEs no sólo reducen la dependencia del petróleo, sino que también actúan como almacenes auxiliares de energía que permiten un uso más eficiente de la misma, puesto que pueden ayudar a regular el voltaje de los perfiles y potencia de la red y compensar las fluctuaciones en la generación de energía con fuentes renovables [9]. Sin embargo, el uso de grandes flotas de VEs puede sobrecargar la red debido al elevado tiempo de carga de las baterías. Así, uno de los grandes retos en la tecnología de VEs es el desarrollo de sistemas inteligentes de control de carga para evitar los picos de demanda [3]. Es necesario planificar adecuadamente la carga para distribuir adecuadamente el consumo y equilibrar la demanda de electricidad. De hecho, ya existen diversas propuestas para obtener sistemas inteligentes de control de carga [11], incluyendo técnicas metaheurísticas, como son los enjambres de partículas [14] o los algoritmos genéticos [7].

Aquí consideramos una variante del problema de carga de vehículos eléctricos definido en [8], inspirado a su vez en un proyecto real que consiste en planificar la carga de un elevado número de VEs estacionados en plazas de aparcamiento privadas [12]. El objetivo es satisfacer al máximo la demanda de electricidad de los usuarios durante su estacionamiento teniendo en cuenta las restricciones físicas de la estación de carga y la existencia de contratos con compañías eléctricas que limitan la cantidad de energía disponible. En este trabajo proponemos incorporar la incertidumbre que en el sistema real tienen los tiempos de carga al problema definido en [8]. Para ello, en la Sección 2 introducimos importantes modificaciones en el modelo original, sobre todo en lo que se refiere a las restricciones físicas de la estación de carga. En la Sección 3 proponemos un algoritmo genético (AG) para resolver el problema resultante y en la Sección 4 presentamos un estudio experimental para evaluar su rendimiento. Finalmente, en la Sección 5 presentamos algunas conclusiones y propuestas de trabajo futuro.

2. Descripción del problema

La red de la estación de carga para la que hay que planificar la carga de los VEs cuenta con una fuente de alimentación trifásica. En total, la estación tiene 180 plazas de estacionamiento, cada una de ellas con un punto de carga conectado a una de las tres líneas eléctricas. En todo momento es necesario mantener un equilibrio de energía entre las líneas. Una descripción más detallada puede encontrarse en [8].

Cuando un usuario (propietario de una determinada plaza de estacionamiento) llega al aparcamiento, conecta su vehículo al punto de carga y proporciona una estimación del tiempo de carga y el instante en el que pretende abandonar el parking, o tiempo de recogida. El sistema de control utiliza estos valores para planificar la carga de todos los VEs, estableciendo el instante de inicio de la carga para cada uno de ellos. Tal y como se propone en [8], en este trabajo consideramos una versión simplificada del problema en la que se supone que un usuario nunca retira su vehículo antes de que se haya cargado completamente, de modo que el objetivo sea minimizar el retraso del tiempo de fin de carga con respecto al tiempo de recogida marcado por el usuario.

Hay ciertas restricciones que se deben cumplir para que la estación de carga funcione correctamente. Aunque hay 180 puntos de carga, no todos pueden estar activos al mismo tiempo ya que, debido a la cantidad de energía contratada, hay un número máximo de vehículos N que pueden estar cargándose al mismo tiempo en una línea dada. Además, por razones técnicas y económicas, el consumo de energía en las tres líneas debe estar equilibrado.

2.1. Incertidumbre en los tiempos de carga

En problemas reales es frecuente que existan diversas fuentes de incertidumbre y que algunos datos no se conozcan con exactitud. En nuestro caso, el tiempo

exacto de carga que necesitará un vehículo no se conoce con antelación. Los valores utilizados hasta ahora son estimados a partir de datos históricos, sin tener en cuenta la dispersión ni el sesgo que puede haber en estos datos. La opción de modelar estos tiempos utilizando distribuciones de probabilidad exige disponer de datos históricos de gran calidad y, además, resulta en un problema de excesiva complejidad computacional. Los intervalos difusos o *fuzzy* constituyen una alternativa interesante [1], donde cada intervalo es una distribución de posibilidad, representando valores más o menos posibles para un tiempo de carga.

En concreto, modelamos cada tiempo de carga usando un *número triangular fuzzy* o TFN, dado por el intervalo $[a^1, a^3]$ de valores posibles y un valor modal a^2 en él. Para un TFN \hat{a} , denotado $\hat{a} = (a^1, a^2, a^3)$, la función de pertenencia tiene la siguiente forma:

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

El modelado del problema con TFNs desde los datos históricos puede realizarse, por ejemplo, considerando como valor modal el estimado actual y como extremos del soporte, los de un intervalo de confianza obtenido desde los datos históricos.

Para manejar tiempos de carga como TFNs, necesitamos realizar sumas y restas, así como calcular el máximo entre dos números. Estas operaciones se obtienen extendiendo las operaciones correspondientes con números reales usando el *principio de extensión* [2]. Tanto la suma como la resta de TFNs es un TFN que se calcula utilizando los 3 vértices: $\hat{a} + \hat{b} = (a^1 + b^1, a^2 + b^2, a^3 + b^3)$ y $\hat{a} - \hat{b} = (a^1 - b^3, a^2 - b^2, a^3 - b^1)$. Como caso particular, para $b \in \mathbb{R}$, $\hat{a} - b = (a^1 - b, a^2 - b, a^3 - b)$. En cuanto al máximo, el conjunto de TFNs no es cerrado bajo esta operación y además el cálculo de la expresión resultante puede ser complejo. Por este motivo, siguiendo a [4], aproximamos el resultado de esta operación por un TFN: $\max(\hat{a}, \hat{b}) \approx (\max(a^1, b^1), \max(a^2, b^2), \max(a^3, b^3))$. Ésta es una aproximación ampliamente utilizada en la literatura y además, coincide en soporte y valor modal con el máximo calculado según el *principio de extensión* [10].

Dada la interpretación de la función de pertenencia de TFN como una distribución de posibilidad sobre los reales, es posible calcular el valor esperado de un TFN \hat{a} como $E[\hat{a}] = \frac{1}{4}(a^1 + 2a^2 + a^3)$. Coincide con varios índices de *defuzzificación* utilizados para establecer un orden total en el conjunto de TFNs [1]. En particular, el valor esperado induce un orden total \leq_E tal que, para dos TFNs cualesquiera \hat{a}, \hat{b} $\hat{a} \leq_E \hat{b}$ si y sólo si $E[\hat{a}] \leq E[\hat{b}]$.

2.2. Formulación del problema

Tal y como hemos descrito, para una instancia del problema de carga de VE's hay tres líneas de carga L_i , $1 \leq i \leq 3$, cada una con n_i puntos de carga. En cada una de las tres líneas se tiene un número máximo $N > 0$ de puntos de carga

que pueden estar activos simultáneamente. La línea L_i recibe un número M_i de vehículos v_{i1}, \dots, v_{iM_i} desde un instante de tiempo 0 hasta el horizonte de planificación. Cada vehículo v_{ij} viene caracterizado por un instante de llegada $t_{ij} \geq 0$, un tiempo de carga $\widehat{p}_{ij} > 0$ (un TFN) y un tiempo de recogida del vehículo d_{ij} , en el que la batería del vehículo debería estar completamente cargada. En esta formulación suponemos que tanto el instante de llegada y como el de recogida proporcionados por el usuario son nítidos y, además, son consistentes en el sentido de que el tiempo de recogida ha de permitir completar la carga del vehículo cualquiera que sea la duración que tenga finalmente, es decir $d_{ij} \geq t_{ij} + \widehat{p}_{ij}^3$.

Una planificación factible consiste en asignar instantes de inicio de carga \widehat{st}_{ij} a todos los VEs v_{ij} , $1 \leq i \leq 3$, $1 \leq j \leq M_i$ de modo que se cumplan las siguientes restricciones:

1. Los vehículos deben empezar a cargarse después de su instante de llegada:

$$\forall v_{ij}, \quad \widehat{st}_{ij}^1 \geq t_{ij} \quad (2)$$
2. Una vez que un vehículo empieza a cargarse, no puede desconectarse antes de que termine la carga, de modo que, si \widehat{c}_{ij} denota el instante de fin de carga para v_{ij} , ha de ocurrir:

$$\forall v_{ij}, \quad \widehat{c}_{ij} = \widehat{st}_{ij} + \widehat{p}_{ij} \quad (3)$$
3. El número de puntos de carga activos en una línea en un instante de tiempo t no puede exceder N . En un contexto determinista, un coche está cargándose en el instante t si t se encuentra entre \widehat{st}_{ij} y \widehat{c}_{ij} . Para extender este concepto al entorno fuzzy, consideramos el instante t como un TFN degenerado, $\widehat{t} = (t, t, t)$ y definimos el número de puntos de carga activos en un instante de tiempo $t \geq 0$ en la línea L_i , $1 \leq i \leq 3$, como:

$$N_i(t) = \sum_{j=1}^{M_i} \delta_{ij}(t) \quad (4)$$

siendo $\delta_{ij}(t) = 1$ si $\widehat{st}_{ij} \leq_E \widehat{t} \leq_E \widehat{c}_{ij}$ y 0 en caso contrario. Así, la restricción de máximo número de puntos de carga activos será:

$$\forall t \geq 0, \quad \max_{1 \leq i \leq 3} N_i(t) \leq N. \quad (5)$$

4. El desequilibrio máximo entre dos líneas cualesquiera L_i y L_j viene dado por un parámetro Δ , de modo que:

$$\forall t \geq 0, \quad \max_{1 \leq i, j \leq 3} \left(\frac{|N_i(t) - N_j(t)|}{N} \right) \leq \Delta. \quad (6)$$

El objetivo es encontrar una solución factible que minimice el retraso total respecto a los instantes de recogida, definido como:

$$\widehat{T} = \sum_{i=1}^3 \sum_{j=1}^{M_i} \max(0, \widehat{c}_{ij} - d_{ij}) \quad (7)$$

Puesto que el valor de la función objetivo es un TFN, distintas soluciones se comparan utilizando \leq_E .

Algoritmo 1 Algoritmo de asignación de tiempos de inicio

entrada: Una permutación de vehículos V
salida: Una planificación S
 Dividir cada línea L_i en N sublíneas $l_i^k : k = 1 \dots N$;
para todo vehículo $v_{ij} \in V$ **hacer**
 $k \leftarrow 0$;
 repetir
 $k \leftarrow k + 1$;
 hasta $t_{ij} \geq E[c_i^k]$ **o** $k = N$
 $st_{ij}^k \leftarrow \max(t_{ij}, E[c_i^k])$;
 $st_{ij}^k \leftarrow \min\{t \geq st_{ij}^k : \text{se cumple la restricción (6) en el intervalo } [t, t + \widehat{p}_{ij}]\}$;
 mientras $st_{ij}^k > E[c_i^{k-1}]$ **y** $k > 0$ **hacer**
 $k \leftarrow k - 1$;
 fin mientras
 Fuzzificar st_{ij}^k en \widehat{st}_{ij} ;
 $\widehat{c}_i^k \leftarrow \widehat{st}_{ij} + \widehat{p}_{ij}$;
 Reordenar las sublíneas l_i^k de mayor a menor según los valores de $E[c_i^k]$;
fin para
devolver La planificación S ;

3. Algoritmo genético

Los Algoritmos Genéticos (AGs) son métodos capaces de explorar de forma eficiente espacios de búsqueda de gran tamaño a la vez que permiten explotar conocimiento heurístico del dominio del problema. Así, constituyen un método de resolución de problemas de scheduling que permite obtener soluciones de buena calidad en un tiempo razonable [13].

En este trabajo proponemos adaptar el AG de [5] que, partiendo de una población inicial formada a partes iguales por individuos aleatorios y generados con una versión estocástica de las reglas de prioridad DDR y EST, la evolución aplicando los operadores genéticos hasta que se satisface un determinado criterio de parada; concretamente, el algoritmo termina si se alcanza una solución con retraso 0 o si se han realizado un número max_{iter} de iteraciones consecutivas sin mejorar el fitness de la mejor solución.

3.1. Evaluación de los cromosomas

Para construir una planificación S a partir de una permutación V , se asigna secuencialmente a cada vehículo el tiempo de inicio más temprano posible de forma que se cumplan las restricciones del problema descritas en la Sección 2.2 respecto a los vehículos ya planificados. El algoritmo 1 detalla este procedimiento. El proceso de decodificación divide cada línea L_i en tantas sublíneas l_i^k como el número máximo de puntos de carga activos por línea, $1 \leq k \leq N$. Para asignar el tiempo de inicio de un vehículo v_{ij} , estas sublíneas se ordenan de mayor a menor según el tiempo de fin de carga esperado del último vehículo planificado en

cada sublínea $E[\widehat{c}_i^k]$. Después, se busca la primera sublínea l_i^k tal que el tiempo de llegada del vehículo t_{ij} sea mayor o igual que $E[\widehat{c}_i^k]$, si existe, considerando la última sublínea, l_i^N , en caso contrario. Para fijar el tiempo de inicio de carga \widehat{st}_{ij} , buscamos el primer instante t mayor o igual a $\max(t_{ij}, E[\widehat{c}_i^k])$ para el que se cumpla la restricción de equilibrio entre todas las líneas L_i en el intervalo $[t, t + \widehat{p}_{ij}]$, de modo que el valor esperado de \widehat{st}_{ij} , st_{ij}^E se hace coincidir con t . A continuación, se comprueba si el vehículo v_{ij} se puede planificar en alguna sublínea k anterior a la inicialmente seleccionada. Una vez fijada la sublínea k en la que se va a planificar el vehículo, se obtiene su tiempo de inicio fuzzy \widehat{st}_{ij} a partir de st_{ij}^E aplicando la Proposición 1. Finalmente se reordenan las sublíneas l_i^k de mayor a menor en función de los valores actualizados $E[\widehat{c}_i^k]$.

Proposición 1 *Un tiempo de inicio fuzzy $\widehat{st}_{ij} = (s^1, s^2, s^3)$ tal que $E[\widehat{st}_{ij}] = st_{ij}^E$ y tal que $s^i \geq c^i$, $i = 1, 2, 3$ para un tiempo de fin $c_i^k = (c^1, c^2, c^3)$ viene dado por:*

$$\begin{aligned} \text{si } E[\widehat{c}_i^k] &= st_{ij}^E, & \widehat{st}_{ij} &= \widehat{c}_i^k; \\ \text{si } E[\widehat{c}_i^k] < st_{ij}^E, & c^3 - st_{ij}^E \leq st_{ij}^E - c^1, & \widehat{st}_{ij} &= \begin{cases} s^3 = \max(st_{ij}^E, c^3), \\ s^2 = \max(st_{ij}^E, c^2), \\ s^1 = 4st_{ij}^E - 2s^2 - s^3; \end{cases} \\ \text{si } E[\widehat{c}_i^k] < st_{ij}^E, & c^3 - st_{ij}^E > st_{ij}^E - c^1, & \widehat{st}_{ij} &= \begin{cases} s^1 = c^1, \\ s^3 = c^3, \\ s^2 = \frac{1}{2}(4st_{ij}^E - s^1 - s^3). \end{cases} \end{aligned} \quad (8)$$

La figura 1 ilustra el proceso de planificación para un problema con cuatro vehículos ($v_{11}, v_{12}, v_{13}, v_{14}$) en L_1 , un vehículo (v_{21}) en L_2 y un vehículo (v_{31}) en L_3 . El tiempo de llegada es 0 para todos los vehículos excepto para v_{14} que es 8, los tiempos de carga son $\widehat{p}_{12} = \widehat{p}_{21} = \widehat{p}_{31} = (4, 5, 6)$ $\widehat{p}_{11} = (10, 11, 12)$, $\widehat{p}_{13} = (14, 15, 17)$ y $\widehat{p}_{14} = (5, 6, 7)$ y los tiempos de recogida son $d_{12} = d_{21} = d_{31} = 9$, $d_{11} = 11$ y $d_{13} = d_{14} = 20$. El número máximo de puntos de carga activos en cualquier línea es $N = 3$ y el parámetro máximo de desequilibrio es $\Delta = \frac{2}{3}$, con lo que en cualquier instante de tiempo debería haber como máximo dos vehículos de diferencia entre dos líneas cualesquiera. Para la permutación ($v_{11}, v_{21}, v_{31}, v_{12}, v_{13}, v_{14}$), suponemos planificados todos los vehículos excepto el último, v_{14} . Para planificarlo, se busca la primera sublínea, k , de la L_1 para la que $E[\widehat{c}_1^k] \leq t_{14}$, que resulta ser l_1^2 (Figura 1(a)). El primer $st_{14}^E \geq 8$ para el que se cumple el equilibrio entre líneas en durante todo el intervalo de carga del vehículo es 11 (Figura 1(b)). Vemos que el vehículo v_{14} se puede planificar a partir de ese mismo instante en otra sublínea l_1^3 (Figura 1(c)): Por lo tanto, se fija el valor de \widehat{st}_{14} a partir de $st_{14}^E = 11$ de forma que sea mayor o igual, componente a componente, al tiempo de fin de la sublínea, \widehat{c}_1^3 , es decir, $\widehat{st}_{14} = (10, 11, 12)$. Finalmente, se reordenan las sublíneas de la línea L_1 (Figura 1(d)).

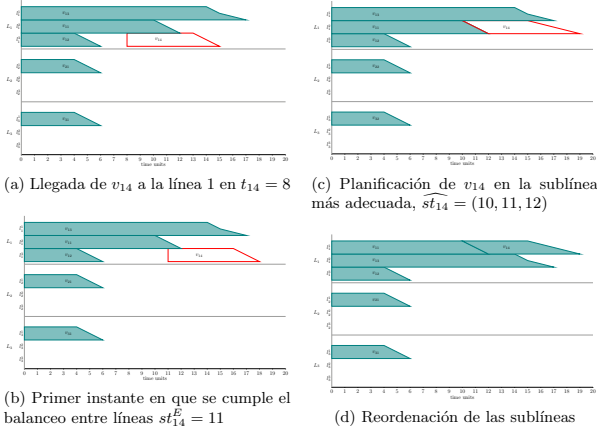


Figura 1: Pasos en la planificación de un vehículo

4. Resultados experimentales

Para evaluar el algoritmo propuesto hemos realizado un estudio experimental con el AG implementado en C++ rodando en un PC con un procesador Xeon E5520 con 24 GB de RAM y un sistema operativo Linux (SL 6.0).

Como banco de pruebas, consideramos una versión fuzzificada del benchmark usado en [8], con instancias basadas en datos reales, introduciendo incertidumbre en los tiempos de carga según el método de fuzzificación de [6]. En todos los casos se considera que la estación de carga está instalada en un parking con 180 plazas de estacionamiento, un horizonte temporal de un día y diferentes perfiles de tiempos de llegada, de carga y de recogida basados en datos reales. En un primer grupo de instancias (*tipo 1*) llegan 60 vehículos a cada línea L_i a lo largo del día, mientras que segundo grupo (*tipo 2*) llegan 108 vehículos a la línea L_1 , 54 a la L_2 y 18 a la L_3 . Es previsible que en este segundo tipo de instancias el algoritmo necesite controlar muchas situaciones de desequilibrio entre las líneas para construir una planificación factible. También consideramos diferentes valores para el parámetro de desequilibrio $\Delta \in \{0,2,0,4,0,6,0,8\}$ y para el número máximo de vehículos que pueden estar cargándose al mismo tiempo en una línea $N \in \{20,30,40\}$. Consideramos 24 instancias, una para cada combinación de *tipo*, Δ y N .

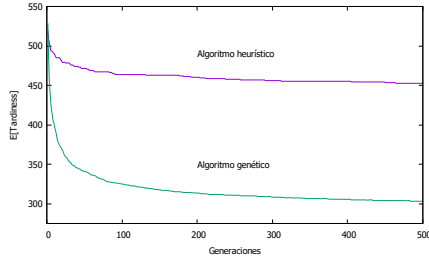


Figura 2: Evolución de un algoritmo heurístico y del AG sobre una instancia de Tipo 1, $N = 20$ y $\Delta = 0,2$.

Tras un estudio paramétrico inicial, se fijaron los siguientes parámetros: población de 200 cromosomas, número máximo de iteraciones sin mejora $max_{iter} = 25$, probabilidad de cruce $prob_{cruz} = 0,8$, probabilidad de mutación $prob_{mut} = 0,1$ y tamaño del torneo $tam_{tor} = 8$ para la creación de la población inicial. En todos los experimentos hemos realizado 30 ejecuciones sobre cada instancia para obtener resultados estadísticamente significativos.

Para analizar la componente evolutiva del AG, comparamos sus resultados con los de un algoritmo que genera tantos grupos de 200 soluciones como generaciones ha necesitado el AG para converger. Cada grupo está formado por algunos individuos aleatorios y otros generados con versiones estocásticas de los heurísticos DDR y EST, en la misma proporción que en la población inicial del AG.

Para ilustrar la convergencia del AG, la Figura 2 muestra, para una instancia de tipo 1, $\Delta = 0,2$ y $N = 20$, la evolución del retraso total para el AG y para el algoritmo mencionado anteriormente, manteniendo la mejor solución encontrada hasta el momento. Hemos realizado 30 repeticiones y calculado la media, generación a generación, para representar la evolución de los dos algoritmos. Obviamente el punto de partida es el mismo y se puede comprobar que el AG es capaz de mejorar en mayor medida la solución inicial (un 45,15 % de mejora) que el algoritmo heurístico (12,86 %).

Por último, en la Tabla 1 comparamos los resultados del AG con los del algoritmo descrito, utilizando el mismo esfuerzo computacional que el AG. Se puede observar que el AG obtiene mejores resultados en todas las instancias, siendo la solución media del AG un 27,5 % mejor que la solución media del algoritmo heurístico para las instancias de tipo 1 y un 17,6 % para las instancias de tipo 2.

Tabla 1: Comparación de los resultados del AG con los obtenidos por el algoritmo heurístico. Se muestran los valores del tardiness para cada instancia en horas.

Instancia N	Δ	AG			Heurístico		
		Mejor M	E[M]	Media	Mejor M	E[M]	Media
Tipo 1							
20	0.2	(200.2, 274.6, 423.5)	293.2	313.0	(344.3, 414.8, 550.7)	431.1	447.0
	0.4	(73.04, 126.7, 244.5)	142.7	144.9	(100.6, 152.8, 271.9)	169.5	183.4
	0.6	(58.06, 109.1, 224.1)	125.1	126.0	(69.17, 117.6, 231.0)	133.8	135.6
30	0.8	(58.44, 109.3, 220.7)	124.4	125.2	(65.83, 116.9, 230.0)	132.4	133.5
	0.2	(19.15, 42.71, 120.7)	56.33	63.34	(82.19, 116.4, 181.6)	124.1	137.4
	0.4	(0.578, 8.474, 46.69)	16.05	16.93	(7.014, 16.50, 63.28)	25.82	26.54
40	0.6	(0.176, 6.549, 42.04)	13.83	14.66	(5.520, 12.96, 54.18)	21.40	22.20
	0.8	(0.092, 6.599, 42.19)	13.87	14.42	(5.520, 12.96, 54.18)	21.40	22.20
	0.2	(1.602, 8.806, 28.13)	11.83	14.54	(14.32, 28.35, 73.53)	36.14	41.08
0.4, 0.6, 0.8		(0, 0, 0)	0.00	0.00	(0, 0, 0)	0.00	0.00
<i>Media</i>			<i>66.46</i>	<i>69.43</i>		<i>91.30</i>	<i>95.74</i>
Tipo 2							
20	0.2	(4407, 4649, 5038)	4686	4742	(5351, 5514, 5756)	5533	5604
	0.4	(1554, 1700, 1954)	1727	1761	(1975, 2115, 2325)	2133	2176
	0.6	(685.6, 823.1, 1060)	847.9	870.8	(925.0, 1035, 1210)	1051	1068
30	0.8	(350.7, 484.9, 716.8)	509.4	519.5	(488.5, 594.2, 777.9)	613.7	623.4
	0.2	(2454, 2664, 2986)	2692	2730	(3001, 3177, 3441)	3199	3293
	0.4	(692.6, 800.3, 989.4)	820.7	839.0	(901.3, 1013, 1192)	1030	1042
40	0.6	(175.6, 254.5, 396.8)	270.3	273.7	(267.1, 342.9, 482.5)	358.9	365.2
	0.8	(38.99, 80.30, 186.2)	96.46	99.88	(64.81, 109.9, 209.3)	123.4	130.2
	0.2	(1567, 1707, 1945)	1731	1765	(1975, 2115, 2325)	2133	2160
40	0.4	(302.4, 386.0, 537.1)	402.9	409.2	(404.4, 493.5, 652.3)	510.9	523.7
	0.6	(27.72, 63.46, 149.5)	76.05	79.23	(56.19, 97.68, 191.4)	110.7	114.8
	0.8	(0.245, 6.807, 56.94)	17.70	18.63	(2.636, 15.41, 71.36)	26.20	27.54
<i>Media</i>			<i>1156</i>	<i>1175</i>		<i>1401</i>	<i>1427</i>

5. Conclusiones

En este trabajo hemos considerado el problema de planificar la carga de VEs en una estación real de carga de manera que se cumplan una serie de restricciones técnicas a la vez que se minimiza una función objetivo de retraso total. Hemos incorporado incertidumbre en los tiempos de carga de los vehículos, modelada mediante números fuzzy triangulares, modificando considerablemente la formulación existente para el problema determinista. Para resolver este complejo problema proponemos un algoritmo genético con operadores diseñados específicamente. De especial importancia es la evaluación de cromosomas, encargada de construir una planificación a partir de una permutación de vehículos. También hemos presentado un estudio experimental sobre instancias basadas en datos

reales que muestra cómo el algoritmo converge adecuadamente obteniendo buenos resultados.

Agradecimientos

Este trabajo ha sido financiado por los proyectos TIN2013-46511-C2-2-P y MTM2014-55262-P.

Referencias

1. Dubois, D., Fargier, H., Fortemps, P.: Fuzzy scheduling: Modelling flexible constraints vs. coping with incomplete knowledge. *European Journal of Operational Research* 147, 231–252 (2003)
2. Dubois, D., Prade, H.: *Possibility Theory: An Approach to Computerized Processing of Uncertainty*. Plenum Press, New York (USA) (1986)
3. EDSO: Position paper on electric vehicles charging infrastructure. Tech. rep., European Distribution system Operators for Smart Grids (EDSO) (10 April 2012), <http://www.edsoformartgrids.eu>
4. Fortemps, P.: Jobshop scheduling with imprecise durations: a fuzzy approach. *IEEE Transactions of Fuzzy Systems* 7, 557–569 (1997)
5. García-Álvarez, J., González, M., Vela, C.: A genetic algorithm for scheduling electric vehicle charging. In: *Proceedings of GECCO 2015*. pp. 393–400 (2015)
6. Ghrayeb, O.A.: A bi-criteria optimization: minimizing the integral value and spread of the fuzzy makespan of job shop scheduling problems. *Applied Soft Computing* 2(3), 197–210 (2003)
7. Hajforoosh, S., Masoum, M.A., Islam, S.M.: Real-time charging coordination of plug-in electric vehicles based on hybrid fuzzy discrete particle swarm optimization. *Electric Power Systems Research* 128, 19–29 (2015)
8. Hernandez-Arauzo, A., Puente, J., Varela, R., Sedano, J.: Electric vehicle charging under power and balance constraints as dynamic scheduling. *Computers & Industrial Engineering* 85, 306–315 (2015)
9. Kang, J., Duncan, S.J., Mavris, D.N.: Real-time scheduling techniques for electric vehicle charging in support of frequency regulation. *Procedia Computer Science* 16, 767–775 (2013)
10. Palacios, J.J., Puente, J., Vela, C.R., González-Rodríguez, I.: Benchmarks for fuzzy job shop problems. *Information Sciences* 329, 736–752 (2016)
11. Rahman, I., Vasant, P., Singh, B., Abdullah-Al-Wadud, M., Adnan, N.: Review of recent trends in optimization techniques for plug-in hybrid, and electric vehicle charging infrastructures. *Renewable and Sustainable Energy Reviews* 58, 1039–1047 (2016)
12. Sedano, J., Portal, M., Hernández Arauzo, A., Villar, J., Puente, J., Varela, R.: Sistema de control autónomo para distribución de energía en una estación de carga de vehículos eléctricos: diseño y operación. Tech. rep., Instituto Tecnológico de Castilla y León ITCL (2012)
13. Talbi, E.G.: *Metaheuristics. From Design to Implementation*. Wiley (2009)
14. Yang, J., He, L., Fu, S.: An improved pso-based charging strategy of electric vehicles in electrical distribution grid. *Applied Energy* 128, 82–92 (2014)