



An EDA-based method for solving electric vehicle charging scheduling problem under limited power and maximum imbalance constraints

Hadi Shahmoradi^a, Majid Esmaelian^{a,*}, Hossein Karshenas^b

^a Department of Management, University of Isfahan, Hezarjerib St., Azadi Square, Isfahan, Iran

^b Department of Artificial Intelligence, Faculty of Computer Engineering, University of Isfahan, Hezarjerib St., Azadi Square, Isfahan, Iran

ARTICLE INFO

Keywords:

Electric Vehicle Charging Scheduling Problem (EVCSP)
Balance constraint
Estimation Distribution Algorithm (EDA)
Markov Network-based EDA
Mallows Model-based EDA
Constraint Programming (CP)

ABSTRACT

Electronic vehicles (EVs) are receiving increasing attention to addressing global warming challenges since fossil fuel is replaced with fuel cell technology. Hence, new challenges arise as demands have increased for using EVs. One of these challenges is the long waiting time of charging EVs spent in queues, especially during peak hours. So, in this study, we aim to propose an efficient method for the electric vehicle charging scheduling problem (EVCSP), which an actual charging station inspires. The most important constraint in this problem is balancing power consumption between charging lines, leading to a limited number of devices that can be charged simultaneously. Also, in this problem, EVs may have interrelationships with each other during the scheduling procedure. So, the estimation of distribution algorithm (EDA) as a competent method in handling the possible relations among decision variables is applied in our proposed hybrid EDA-based solving method. Our proposed method comprises two EDAs, a Markov network-based EDA and a Mallows model-based EDA. It achieves an appropriate schedule and charging line assignment simultaneously while minimizing the total tardiness considering problem constraints. We compared our method with a constraint programming (CP) model and the state-of-art meta-heuristic methods in terms of the objective function value by simulation on a benchmark dataset. Results from the experimental study show significant improvement in solving the introduced EVCSPs.

1. Introduction

Releasing greenhouse gas (GHG) into the atmosphere by burning fossil fuels, especially in the transportation sector, is one of the main contributors to global warming and climate change (Acar & Dincer, 2020; Zhou et al., 2020). To address the global warming challenges, non-fossil fuels (such as hydrogen in fuel cell technology) are receiving increasing attention (Chen et al., 2019; Wilberforce et al., 2017). Fuel cell technology in the automotive industry as one of the successful efforts results in emerging electric vehicles (EVs) in the market (Chung et al., 2018). Although EVs bring several advantages to transportation, they require frequent and long-time charging. Hence, with increasing demand for EVs (Jian et al., 2017), they should spend long queuing hours, especially during peak hours (Zhu et al., 2014). Therefore, an efficient method is required for the electric vehicle charging scheduling problem (EVCSP) (Al-Ogaili et al., 2019).

The EVCSP addressed in this study is inspired by an actual charging station first raised in (Hernández Arauzo et al., 2013). This problem includes an EV station with multiple charging points in three different

lines of a three-phase feeder. The assignment of EVs to the charging points is pre-defined. The critical constraint for charging scheduling in this station is to impose an upper bound on the number of EVs connected to each line at any given time.

Various studies have been proposed so far to address the aforementioned problem while minimizing the charging delays of all EVs such as (Hernández-Arauzo et al., 2015), (García-Álvarez et al., 2015), (Mencía et al., 2017), (García-Álvarez et al., 2018), (Mavrovouniotis et al., 2019). In this study, we also aim to achieve an effective schedule for this EVCSP. Moreover, we consider a more generalized form of this problem in which EVs are not pre-assigned to charging lines. So, we also aim to address this generalized problem requiring the assignment of EVs to charging lines to be handled by the proposed algorithm.

We propose a hybrid estimation of distribution algorithm (EDA) as our solving method for the outlined problem. Since these problems are NP-hard, EDAs as metaheuristic algorithms are proper for obtaining good results in a reasonable computational time (García-Álvarez et al., 2018). In addition, there are possible interactions between EVs during the scheduling procedure while considering the problem constraints,

* Corresponding author.

E-mail addresses: ha.shahmoradi@ase.ui.ac.ir (H. Shahmoradi), m.esmaelian@ase.ui.ac.ir (M. Esmaelian), h.karshenas@eng.ui.ac.ir (H. Karshenas).

<https://doi.org/10.1016/j.cie.2022.108544>

Received 14 July 2021; Received in revised form 1 March 2022; Accepted 2 August 2022

Available online 6 August 2022

0360-8352/© 2022 Published by Elsevier Ltd.

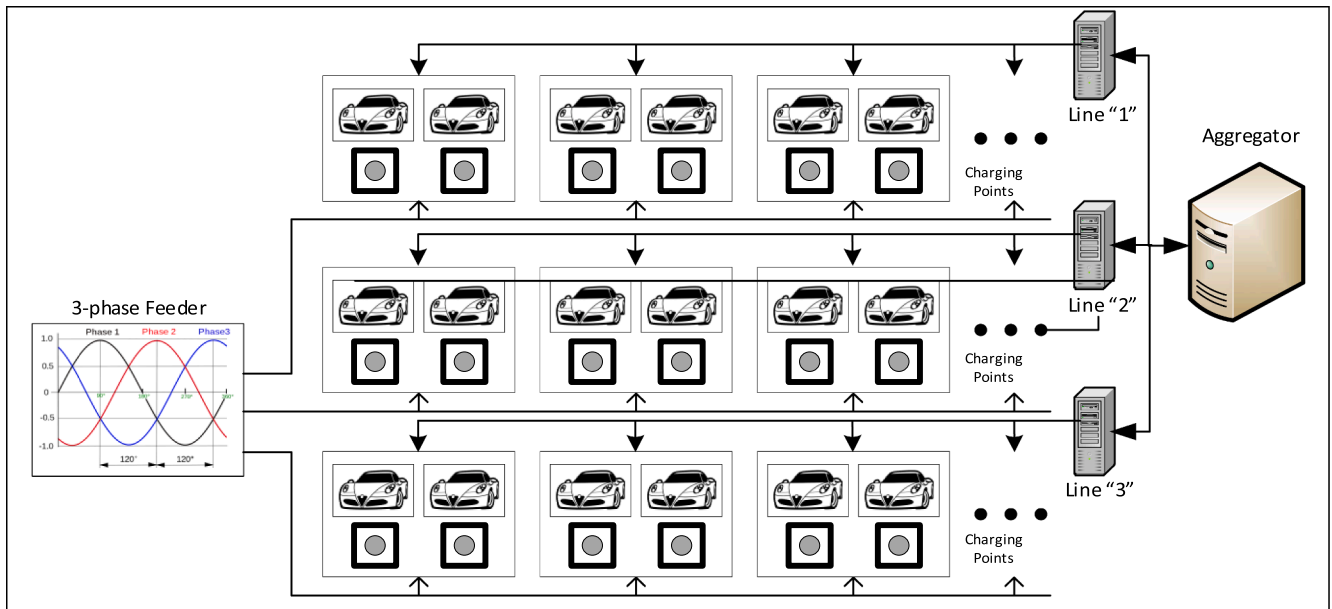


Fig. 1. A general structure of an EV charging station with multiple charging points in three charging lines (García-Álvarez et al., 2018).

and EDAs are competent in handling the possible relations among decision variables (Kacprzyk & Pedrycz, 2015).

The strength of EDAs is in their search mechanism that can learn and adapt itself according to the problem. They are rooted in evolutionary computing and machine learning. From the evolutionary perspective, they have adopted using a population of promising solutions that evolve iteratively by performing selection and variation steps. They also adopted the idea of data-driven modeling from machine learning in which a probabilistic model is learned based on a population of previously visited reasonable solutions, and the resulting model is used to infer where other reasonable solutions might be (Kacprzyk & Pedrycz, 2015).

Our EDA-based proposed approach comprises two EDAs: a Markov network-based EDA (MN-EDA) and a Mallows model-based EDA (MM-EDA) algorithms. We called our proposed method “HMM-EDA” after the Hybrid Markov network and Mallows model-based EDA. The reason for this hybridization is that we model the EVCSP with two types of decision variables; (1) assignment of each EV to a charging line and (2) sequence of all EVs charging. Markov network is a suitable model for dealing with the interaction among variables (Shakya et al., 2012). Due to the shared power supply in our considered EVCSP, EVs interact with each other once they seize a power line. So, MN-EDA is proposed to handle charging line assignment while considering these interactions. In addition, by considering EV sequence as a permutation problem, we offered MM-EDA since the Mallows model is a suitable approach for permutation modeling (Zangari et al., 2017). It is worth mentioning that permutation problems are defined as ordering specific symbols where two symbols cannot have the same position, and two positions cannot belong to the same symbol (Ceberio et al., 2012).

The proposed method is evaluated using the benchmark introduced in (García-Álvarez et al., 2018). This benchmark is obtained from a sample station with three lines where each line has 60 charging points (180 charging points in total). It includes arrival times and due times corresponding to the expected behavior of real users in different situations. Even though the charging line assignment is pre-determined in this benchmark, we have also considered this assignment as a decision variable due to specific features of our generalized problem.

To compare the performance of the HMM-EDA with other solving methods, we proposed a constraint programming (CP) model since it provides an optimal solution for our problem. Moreover, we solved our problem with state-of-art meta-heuristic methods, including simulated

annealing (SA), discrete particle swarm optimization (PSO), and artificial bee colony (ABC). Results are compared in terms of the gap from the best solution with respect to the objective, showing the superiority of the proposed HMM-EDA.

The contributions of this study are as follows:

1. According to the literature, it is assumed that the assignment of each EV to the charging lines is known. However, in this study, new decision variables are added to the problem formulation for handling EV assignment, which is very important for real-world applications. It is worth mentioning that solving the more generalized form of this EVCSP has not been addressed in the literature yet.
2. To the best of our knowledge, the EDA-based method has not been used to solve EVCSPs. In this study, we developed and proposed a hybrid Markov network and Mallows model EDA (HMM EDA) method to solve this scheduling problem, including problem constraints, to minimize the total tardiness. To enhance the performance of the proposed EDAs, we proposed an initial population generation algorithm, a diversity control mechanism, and a heuristic local search method.
3. In this study, a constraint programming (CP) model of the general proposed form of EVCSP is proposed.

The rest of this paper is organized in the following way. In the next section, we explained the problem formulation. Then in section 3, we briefly reviewed the related studies in EVCSP. We describe our proposed method in section 4, which is followed by the discussion and results section. In the end, the conclusion is presented in section 6.

2. Problem formulation

This study's EV charging scheduling problem (EVCSP) is motivated by (Hernández Arauzo et al., 2013). It includes an EV station with 60 charging points in each of the three lines (180 charging points total) of a three-phase feeder (Fig. 1). As it is shown in Fig. 1, there is an aggregator which is responsible for scheduling the charging of EVs while considering the following constraints:

1. Active charging points upper bound- restricts the maximum number of EVs that can be connected to each line at any given time.

Table 1

Problem parameters and variables in the static EVCSP variant.

Parameters	
$N_i(t)$	Number of active EVs in charging lines in time t
a_j	Arrival time corresponding to j^{th} EV
c_j	Charging time corresponding to j^{th} EV
d_j	The due time corresponding to j^{th} EV
N	Maximum number of active EVs in all charging lines at any given time
M	Total Number of EVs
Δ	$\Delta \in [0, 1]$ The maximum possible imbalance between lines
Variables	
s_{ij}	Start of charging time corresponding to j^{th} EV, which is assigned to i^{th} charging line
e_{ij}	End of charging time corresponding to j^{th} EV, which is assigned to i^{th} charging line

2. Power imbalance constraint- controls power imbalance among lines due to economic and electro-technical reasons.
3. Non-preemptive charging- refers to a no-interruption charging process once the charging of each EV starts.

In real-life EV stations, the arrival time, charging time, and due time (expected time of leaving the station) of each EV is unknown before an EV arrives in a station. This problem is dynamic. However, for solving and evaluation purposes, a static variant of the problem is mainly considered in studies where the behavior of each EV, including arrival, charging, and due times is known in advance (García-Álvarez et al., 2018). In this study, we considered both the static and the dynamic variant of the problem. The problem formulations of these variants are described in the following.

In the formulation of the static variant of EVCSP, it is assumed that arrival times, charging times, and due times of all EVs are known in advance. The problem parameters and variables are tabulated in Table 1.

The mathematical formulation of the static problem is aimed to minimize the total tardiness of all EVs according to Eq. (1) concerning the problem constraints, which are described in Eq. (2)–(5).

$$\sum_{i=1}^3 \sum_{j=1}^M \max(0, e_{ij} - d_j) \quad (1)$$

$$s_{ij} \geq a_j, \quad \forall 1 \leq i \leq 3, 1 \leq j \leq M \quad (2)$$

$$e_{ij} = s_{ij} + c_j, \quad \forall 1 \leq i \leq 3, 1 \leq j \leq M \quad (3)$$

$$N_i(t) \leq N, \quad \forall 1 \leq i \leq 3 \quad (4)$$

$$\frac{|N_i(t) - N_j(t)|}{N} \leq \Delta, \quad \forall 1 \leq i, j \leq 3 \quad (5)$$

The constraint defined in Eq. (2) restricts each EV from starting the charging process before its arrival time. According to the constraint in Eq. (3), EVs are charged in a non-preemptive process. So, they must not be disconnected from their charging point until their charging process is finished. Eq. (4) assures that the number of active EVs at any given time and any given line should not exceed N . At last, Eq. (5) imposes an upper bound on the imbalance between every-two lines.

The dynamic variant of the problem is modeled as a sequence of n static problem instances (P_1, P_2, \dots, P_n). Each instance includes EVs that are already arrived in the station in which some of the EVs are in the middle of their charging process, and others are pending to start their charging process. Consequently, EVs' charging time c_j and due time d_j of each instance is known. At each time interval, the aggregator checks for new EV arrivals. For the sake of server workload in the case of arriving many EVs simultaneously and efficient use of resources, the time

Table 2

Review of recent centralized EVCSP studies from an operations research perspective.

Related Studies	Objective function	Constraints	Optimization method
(Z. Wang et al., 2020)	minimize the overall peak-valley load difference	EV State-of-charge Capacity power limit	A scenario-based stochastic linear programming
(Zhou et al., 2020)	minimize the overall peak-valley load difference	Slow/fast charging EVs	Heuristic coordinated charging scheduling
(Savari et al., 2019)	minimize the total cost	EV State-of-charge Slow/medium/fast chargers	Particle swarm optimization
(Koufakis et al., 2019)	minimize the total cost	EV State-of-charge	Mixed Integer Programming
(Mavrouniotis et al., 2019)	minimize the total tardiness	Power and balance constraints	ant colony optimization
(García Álvarez et al., 2018)	minimize the total tardiness	Power and balance constraints	artificial bee colony
(García-Álvarez et al., 2018)	minimize the total tardiness	Power and balance constraints	Greedy Randomized Adaptive Search Procedure
(Jian et al., 2017)	minimize the overall peak-valley	EV State-of-charge Slow/medium/fast chargers	valley-filling strategy
(Mencia et al., 2017)	minimize the total tardiness	EV State-of-charge	Genetic algorithm
(Hernández-Arauzo et al., 2015)	minimize the total tardiness	Power and balance constraints	Dynamic Constraint Satisfaction Problems
(García-Álvarez et al., 2015)	minimize the total tardiness	Power and balance constraints	Genetic algorithm
(Mirzaei et al., 2014)	maximize the profit of the parking operator and the costs of EV owners	Upper bound for active EVs in charging lines	Particle swarm optimization

interval is usually set at two minutes (García Álvarez et al., 2018). Accordingly, the scheduling problem is defined as determining the starting time s_{ij} of each pending EV in each time interval with respect to all constraints described in the static variant of the problem while minimizing the total tardiness. Notice that the s_{ij} may be modified in further P_n instances as long as the corresponding EV does not start charging.

3. Literature review

A comprehensive review of scheduling, clustering and forecasting strategies for controlling EV charging can be found in (Al-Ogaili et al., 2019). From Al-Ogaili's perspective, there are two main EV charging scheduling strategies: centralized and decentralized. In the centralized approach, EVs are connected to an EV aggregator. The EV aggregator collects charging data from EVs and electrical grids to solve the optimization problem for the sake of its stakeholders, operators of charging stations, and EV owners (Jin et al., 2013). In the decentralized strategy, each EV owner has the authority to operate strategically to minimize their charging costs. Although this strategy offers more flexibility to EV owners, it does not guarantee optimality (Ma et al., 2011). Our study is focused on the centralized strategy.

As far as we know, EVCSPs are not reviewed from the operation research (OR) perspective. So, we reviewed EVCSP from their problem

Table 3

Review of recent EDA-based method in solving the scheduling problem.

Related Studies	Problem	EDA Type
(Du et al., 2021)	distributed flexible job shop scheduling with crane transportations	Order statistics
(Amiri & Behnamian, 2020)	Multi-objective green flowshop scheduling problem	Order statistics
(Xue et al., 2019)	the unrelated parallel-machine green scheduling problem	Order statistics
(Shao et al., 2019)	Multiobjective Distributed, No-Wait Flow-Shop Scheduling Problem	Order statistics
(Pérez-Rodríguez & Hernández-Aguirre, 2019)	vehicle routing problem with time windows	Mallows Estimation of Distribution
(Dai et al., 2019)	Energy-Efficient Job-Shop Scheduling Problem	Order statistics
(C. Liu et al., 2018)	resource consumption of heterogeneous, batch processing machines	copula-based EDA
(Zangari et al., 2017)	Permutation Flowshop, Scheduling Problem	Mallows Estimation of Distribution
(Qian et al., 2017)	m-machine reentrant permutation flow-shop scheduling problem	copula-based EDA
(K. Wang et al., 2015)	Stochastic Permutation Flowshop	Order statistics
(Tiwari et al., 2015)	multi-objective permutation flow shop scheduling problem	Order statistics
(Hao et al., 2014)	robust Resource, Constrained Project Scheduling	Markov network-based EDA
(S. Wang et al., 2013)	distributed permutation flow-shop scheduling problem	Order statistics
(Tzeng et al., 2012)	Permutation flow shop scheduling	Order statistics
(H. Liu et al., 2011)	permutation flowshop scheduling problem	Order statistics

constraints, objective function, and solving method points of view. Table 2 summarizes our findings of the recent literature dealing with centralized EVCSPs from the OR perspective.

According to Table 2, from the objective function, our study lies in minimizing total tardiness. From a constraints point of view, we consider the power and balance constraints in our research. Moreover, from our optimization and solving method perspective, we proposed an EDA-based method that has not been addressed in the EVCSP literature to the best of the authors' knowledge. In the following, we reviewed, EDA-based approach in scheduling literature.

3.1. EDA-based method in the scheduling problem

Considering that the EVCSPs lie in scheduling problems, we reviewed the EDA-based method in solving scheduling problems (Table 3). According to the following Table, EDAs are applicable in scheduling problems; nevertheless, it does not use in the EVCSPs yet. It should be pointed out here that most of the EDA-based solving methods in the studied scheduling problem applied the order statistics approach as the EDA probabilistic model. In this approach, EDA probabilistic model is a matrix of probabilities of symbol i in position j .

4. Proposed method

This section comprises three sub-sections. This study's preliminary concepts, including introductory concepts of EDAs, Markov network-based EDA, Mallows Model-based EDA, and the CP paradigm, are briefly described in the first part. Next, our proposed HMM-EDA model is illustrated for solving EVCSP. In the final section, CP as an optimal solving approach for EVCSP is explained.

4.1. Preliminaries

EDAs are random optimization methods investigating the potential solutions space by modeling the possible promising solutions (Hauschild & Pelikan, 2011). The resulting model helps recognize relationships among variables that provide valuable and understandable insights into the real-world problem structure. In addition, determining a potential distribution could increase the probability of finding an estimated area, including the promising solutions in the entire problem-solving space. Also, this distribution could reduce the number of function evaluations (NFE) for approaching the optimum solution (Larrañaga & Lozano, 2001). The repetitive steps in an EDA are:

1. Selection- selecting a set of promising solutions from the population;
2. Model Learning- learning a probability model from the promising solutions; and
3. Sampling- producing a sample from the probability model (Larrañaga & Lozano, 2001).

We described two EDA versions based on the Mallows model and the Markov network in the first two parts. Besides, a CP model is briefly explained in part three, and the optimal solution obtained from the CP model is considered as a baseline for comparison purposes.

4.1.1. Mallows models based EDA (MM-EDA)

In MM-EDA, a Mallows model is learned based on the current population. Then, the learned Mallows model is used to find other promising solutions for the next population generation. This process is repeated until the termination condition for MMEDA is met. In the following, we first describe the Mallows model and its learning procedure, then explain the sampling method for generating the next population.

(a) Mallows Model

The Mallows Model (MM) is an exponential family of probability models for permutation problems based on the distance between two permutations. The probability value of every permutation ($p(\delta)$) is defined by two parameters: (1) spread parameter (θ); and (2) a central permutation (δ_0). So, the MM is expressed as follows (Arrieta, 2014):

$$p_{\theta, \delta_0}(\delta) = \frac{\exp(-\theta d(\delta, \delta_0))}{\psi(\theta)} \quad \text{where} \quad \psi(\theta) = \sum_{\delta} \exp(-\theta d(\delta, \delta_0)) \quad (6)$$

Where $\psi(\theta)$ denotes the normalization constant. Also, δ_0 refers to the central permutation. When $\theta = 0$, the probability distribution is uniform. When $\theta > 0$, δ_0 is the permutation with the highest probability value (distribution mode), otherwise in the case that $\theta < 0$, δ_0 is the antimode, i.e., the permutation with the lowest probability. Also, $d(\delta, \delta_0)$ denotes the distance metric, which can be defined in several ways such as Kendall's- τ , Spearman's- ρ , Spearman's footrule, Cayley, Hamming, and Ulam (Diaconis, 1988).

Due to the dynamic and NP-hard nature of EVCSP and the importance of required time by the problem-solving method, we choose Kendall's- τ distance metric (Arrieta, 2014). Given two permutations δ_1 and δ_2 , the Kendall- τ distance is defined by the total number of pairwise disagreements between two permutations. In other words, it counts the minimum number of adjacent swaps to convert δ_1 into δ_2 . Formally, it can be written as follows (Ceberio et al., 2011):

$$d(\delta_1, \delta_2) = |\{(x, y) : x < y, (\delta_1(x) < \delta_1(y) \wedge \delta_2(x) > \delta_2(y)) \vee (\delta_2(x) < \delta_2(y) \wedge \delta_1(x) > \delta_1(y))\}| \quad (7)$$

Which can also be written as:

$$d(\delta_1, \delta_2) = \sum_{j=1}^{M-1} V_j(\delta_1, \delta_2)$$

Where $V_j(\delta_1, \delta_2)$ is the minimum number of adjacent swaps to set in the j^{th} position of δ_1 , $\delta_1(j)$, to the value in $\delta_2(j)$. Also, M represents the permutation length.

The MM (Eq. (6)) under Kendall's- τ distance can be rewritten as follows (Ceberio et al., 2013):

$$p(\delta) = \frac{\exp\left(\sum_{j=1}^{M-1} -\theta V_j(\delta, \delta_0)\right)}{\psi(\theta)} \quad (9)$$

So, the probability distribution of the random variables $V_j(\delta, \delta_0)$ can be written as (Ceberio et al., 2013):

$$p(V_j(\delta, \delta_0) = r_j) = \frac{\exp(-\theta r_j)}{\psi_j(\theta)} r_j \in \{0, \dots, M-j\}. \quad (10)$$

(b) Learning

The exact estimation of the MM parameters can be computationally intractable. By experimental evaluation, Arrieta showed that the approximate learning process and the Kendall's- τ distance metric find accurate estimators in polynomial time for MM (Arrieta, 2014).

Like any other EDAs, in MM-EDA, a probability model is learned in each step from a set of promising solutions. For this purpose, δ_0 and θ parameters are estimated given a dataset of permutations $\{\delta_1, \dots, \delta_Q\}$ using the maximum likelihood estimation (MLE) method. The log-likelihood function can be written as follows (Ceberio et al., 2011):

$$\begin{aligned} \log \mathcal{L}(\{\delta_1, \dots, \delta_Q\} | \delta_0, \theta) &= \sum_{k=1}^Q \log p(\delta_k | \delta_0, \theta) = \sum_{k=1}^Q \log \frac{\exp(-\theta V_j(\delta_k, \delta_0))}{\psi_j(\theta)} \\ &= \sum_{k=1}^Q (-\theta V_j(\delta_k, \delta_0) + \log \psi_j(\theta)) \\ &= -K(\theta \bar{V}_j) - K(\log \psi_j(\theta)) \end{aligned} \quad (11)$$

Where $\bar{V}_j = \sum_{k=1}^Q V_j(\delta_k, \delta_0)/Q$ which means \bar{V}_j is the observed mean for V_j .

Due to the independence calculation of δ_0 and maximization of Eq. (11) with respect to θ , MLE can be divided into two steps: first, δ_0 is obtained, and then θ is computed for the given $\hat{\delta}_0$. The problem of calculating δ_0 is called rank aggregation. The permutation δ_0 that maximize the log-likelihood can be expressed as follows (Arrieta, 2014):

$$\hat{\delta}_0 = \arg \max_{\delta_0} \sum_{k=1}^Q -V_j(\delta_k, \delta_0) = \arg \min_{\delta_0} \sum_{k=1}^Q V_j(\delta_k, \delta_0) \quad (12)$$

In other words, Eq.(12) denotes that the MLE for δ_0 is given by the permutation that minimizes the sum of the Kendall's- τ distances to the samples in the dataset. This problem is known to be NP-hard (Bartholdi et al., 1989). Hence, we use the heuristic method introduced by Ceberio et al. as our solving method (Ceberio et al., 2011).

Once δ_0 is known, the MLE for estimation of θ is achieved by differentiating Eq. (12) with respect to θ and equaling to zero. Hence, θ is estimated to satisfy the following expression (Arrieta, 2014):

$$\frac{M-1}{e^\theta - 1} - \sum_{j=2}^M \frac{M e^{-\theta j}}{1 - e^{-\theta j}} = \bar{d} \quad (13)$$

Eq. (13) has no closed-form solution. So, we solved it with the Netwon-Rapshon algorithm (Arrieta, 2014).

(c) Sampling

To sample a new solution from the learned model (Eq. (9)), a method described by Ceberio et al. is applied (Ceberio et al., 2013). In this sampling method, vectors $V_j(\delta_k, \delta_0)$, $1 \leq k \leq K-1$ with the distribution according to Eq. (10) is generated(9). Then, the new permutations are calculated using an algorithm proposed by Meila et al., as illustrated in Algorithm 1 (Meila et al., 2012). Finally, each permutation is obtained by inverting and composing with the central permutation.

Algorithm 1. Conversion from $V(\pi)$ to π^{-1}

```

1 Input: A vector  $V(\pi)$ 
2 Create an empty list  $\pi^{-1}$ 
3 Insert  $n$  in position 0
4 for  $j = M-1 : 1$  in decreasing order
5   Insert  $j$  in position  $V_j$  of  $\pi^{-1}$ 
6 output:  $\pi^{-1}$ 

```

4.1.2. Markov network-based EDA (MN-EDA)

In Markov network-based EDA (MN-DEA), a Markov network is learned based on the current population. The other procedure in this variant of EDA is the same as other EDAs. To elucidate MN-EDA, first, we describe Markov networks and their learning procedure, then explain the sampling method for generating the next population.

(a) Markov Network Model

A Markov network (MN) model is represented by an undirected graph, G , and a set of parameters φ . In this graph, each node denotes a decision/random variable, X_i , and the edges, E_{ij} , represent the relationships between the variables X_i and X_j through their affinity potentials. Consequently, the relationship between two nodes is seen as a neighborhood relationship in which the neighborhood H_i of node X_i includes all nodes connecting to the node X_i by an edge. The neighborhood relationship can be written in terms of probability as follows (Shakya et al., 2012):

$$p(x_i | x_{-i}) = p(x_i | H_i) \quad (14)$$

In addition, an MN is described in terms of cliques, ζ_i , a fully connected subset of nodes. Thus, the joint probability distribution can be written as follows (Shakya et al., 2012):

$$p(x) = \frac{1}{Z} e^{-\sum_{i=1}^Q u_i(\zeta_i)/T} \quad (15)$$

Where, Q is the number of cliques in the undirected graph structure G and Z is the normalization term called partition function. Also, $u_i(\zeta_i)$ is a potential function on clique ζ_i which can have many different functional forms. The parameters of this function that completely defines the joint probability are estimated from the data. T is a normalization constant.

(b) Learning

Several different approaches can be applied for estimating an undirected structure. We use the Mutual Information (MI) index to determine the graph structure to reduce the complexity. So, the matrix of mutual information is created in which the cross-entropy of each pair of variables is estimated. Using Eq. (16), the MI value between two random variables X_i and X_j is calculated as (Shakya et al., 2012):

$$MI(X_i, X_j) = \sum_{x_i \in X_i} \sum_{x_j \in X_j} p(x_i, x_j | D) \times \log \left(\frac{p(x_i, x_j | D)}{p(x_i | D) p(x_j | D)} \right) \quad (16)$$

Where $p(x_i | D)$ and $p(x_j | D)$ are the marginal probabilities of the decision variables X_i and X_j , respectively, on a set of promising solutions, D . Also, $p(x_i, x_j | D)$ is the joint probability of X_i and X_j and the summation represents all possible combinations of values for random variables X_i and X_j . If the MI value of two variables is higher than a threshold value, we consider them as neighbors and draw an edge between them. This

threshold could be given as a fixed value or can be updated based on MI values as follows:

$$\text{threshold} = \alpha \times \text{avg}(\text{MI}) \quad (17)$$

The value of α controls the density of the edges in the MN network. A larger value α implies fewer edges and, therefore, a sparser structure requiring less computational time. In addition, if the number of neighbors to a decision variable is higher than the maximum number, MN only keeps the neighbors with the highest MI.

The structure learning results in a graph with neighborhood relationships that estimate variables' conditional probabilities.

(c) Sampling

Unlike Bayesian networks, the MN model does not assume any ancestral ordering between variables to allow forward sampling techniques (Shakya et al., 2012). Thus, a Gibbs sampler is used as one of the Markov chain Monte Carlo (MCMC) methods for sampling. It is a randomized algorithm that allows obtaining samples from an approximation of the distribution of interest. Several different versions of the Gibbs sampler can be implemented for this purpose (Shakya et al., 2012). Algorithm 2 describes a version that is used for this work.

Algorithm 2. Gibbs sampler Algorithm in MN-based EDA.

```

1 Input: Neighborhood relationships
2 Generate a random solution  $x = (x_1, \dots, x_K)$ 
3 for each iteration
4   Randomly choose a variable  $x_i$  from  $x$ 
5   Compute conditional probabilities  $p(x_i|H_i)$  according to Eq. (14), and the
   neighborhood relationship
6   Sample  $p(x_i|H_i)$  to get new  $x_i$ .
7 output: new sampled  $x_i$ 

```

According to Algorithm 2, a single solution is created after terminating the Gibbs sampler algorithm. Hence, multiple executions should be done to generate the population.

4.1.3. Constraint programming (CP)

Constraint programming is a problem-solving approach that distinguishes between problem constraints and search algorithms. This approach is used to solve combinatorial problems requiring searching in a potentially exponential space to find an optimal solution, and it is often made possible by constraint propagation algorithms (Dincbas, 1988). Compared to other optimization models, CP models take advantage of the following key characteristics that make it a good method for optimization problems: (1) Rich modeling language suitable for representing complex real-world problems with discrete variables (Koltai et al., 2021), (2) Supporting a different variety of constraints including equality and inequality constraints, logic constraints, non-linear constraints, and almost any possible expression; (3) Prune the search space which contains infeasible solutions using constraint propagation mechanism which results in tackling several large problems. So unlike MIP problems, increasing the number of constraints may indeed be beneficial as it helps propagation (4) Ability to directly model with global constraints (handling relationships among large sets of decision variables), that implement efficient propagators (Oliveira & Carravilla, 2017). Although there are specific well-performing global constraints that have been designed and tested for scheduling problems (Bukchin & Raviv, 2018). Hence, CP models have attracted the attention of many researchers, particularly for scheduling problems (Esmaelian et al., 2021). It should be mentioned that our problem which is rooted in a complex real-world situation, is categorized as a scheduling problem. So it can benefit from some global constraints and non-linear constraints that CP offers.

In general, CP models can be divided into two groups: the constraint satisfaction problems; and the constraint optimization problems. In constraint satisfaction problems, the search procedure for finding a

R	r_1	\dots	r_j	\dots	r_M
L	l_1	\dots	l_j	\dots	l_M

Fig. 2. The encoding of decision variables for proposed modeling of the EVCSP.

solution includes assigning values to variables from each variable's domain, satisfying all the problem constraints. While in constraint optimization problems, the search procedures satisfy the problem constraints and find the best solution.

4.2. Proposed model

A competent EDA is designed using an efficient initial population, a suitable probabilistic model, a proper diversity control mechanism, and a hybrid local search method. An efficient initial population includes either diverse or high-quality solutions rather than random solutions. A suitable probabilistic model results in sampling more promising solutions. A diversity control mechanism provides an effective search that helps the algorithm prevent early trapping in local minima. Furthermore, a hybrid local search method enhances the algorithm's ability to explore the search space locally and provides an appropriate balance between the search method's exploitation and exploration abilities. Accordingly, in this study, we developed our EDA-based method while considering all these remarks. In the following, each remark is explained in more detail.

4.2.1. Efficient initial population

To achieve a suitable balance between the quality and diversity of solutions in the initial population, we use two stochastic dispatching rules, including the due time and the latest start time rules based on previous studies (Garcia Alvarez et al., 2018). The latest start time is defined as $LST_j = d_j - c_j$ where d_j is due time and c_j is the charging time of j^{th} EV. Hence, in this study, the initial population is generated according to Algorithm 3.

Algorithm 3. Initial Population Generation Algorithm.

```

1 Input: EVs due times and charging times
2 for each population member
3   Sort ascendingly the due time vector
4   Sort ascendingly the latest start time ( $LST_j = d_j - c_j$ ) vector
5   Generate an empty vector  $R$  with length  $M$ 
6   for  $i = 1 : n$ 
7     Select a rule randomly from the rule list {due time , latest start time ,
       random }
8     if random rule
9       Append an EV to position  $i$  that is randomly selected from the list of non-
       selected EVs
10    if due time rule
11      Append an EV to position  $i$  that is after previously appended EV in due
       time vector and not
12      yet selected in  $R$ 
13      If no EV is found, then select EV using random rule
14    if latest start time rule
15      Append an EV to position  $i$  that is after previously appended EV in latest
       start time
16      vector and not yet selected in  $R$ 
17      If no EV is found, then select EV using random rule
18 output: initial population

```

4.2.2. Probabilistic model

Considering the objective function in our EVCSP, minimizing the total tardiness, we modeled our problem using two decision variables, encoding as Fig. 2.

1. R - a vector representing a sequence of all EVs with length M ;

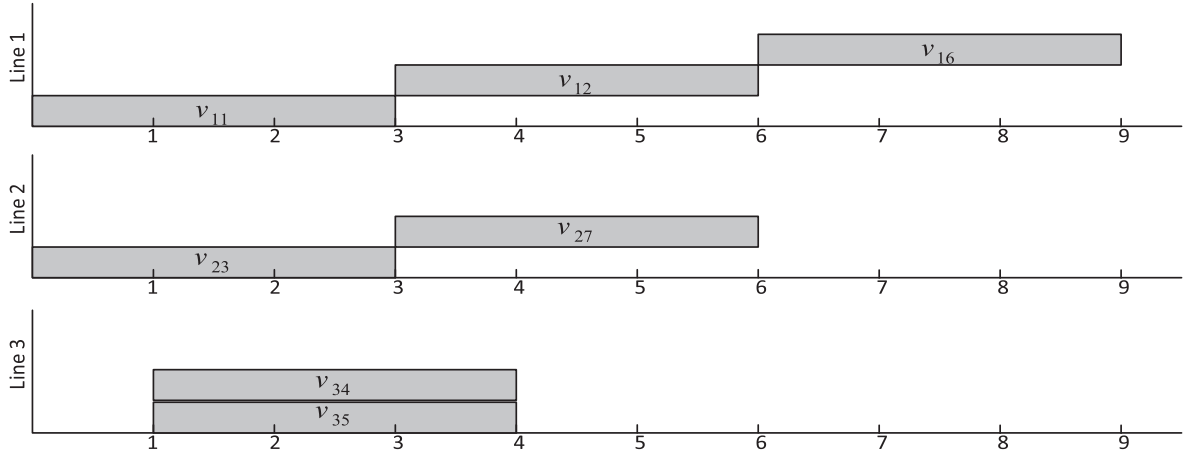


Fig. 3. An example of a feasible solution according to the decision variables.

2. L - a vector representing an assignment of each EV to charging line with length M ($1 \leq l_j \leq 3$).

According to these decision variables, the start time of each EV (s_{ij}) is not considered as a decision variable. Indeed, s_{ij} is the earliest time that satisfies all problem constraints considering EV position in R sequence and its value in L vector (l_j). To shed light on this method, consider a hypothetical example including 7 EVs with the same charging time (3 minutes) and arrival times $\{1, 2, 1, 2, 2, 1, 2\}$. One feasible solution includes $R = \{3, 7, 1, 2, 6, 5, 4\}$ and $L = \{2, 2, 1, 1, 1, 3, 3\}$. Also, in this example, $N = 2$, $\Delta = 0.2$. Fig. 3 shows the corresponding feasible solution. Compared to models in which start time is considered as a decision variable, our model complexity is reduced due to domain reduction.

To find a solution for these decision variables, we use the Mallows model and Markov network for encoding the distributions of R and L vectors, respectively.

The EV sequence (R) is defined in a permutation space which makes the Mallows model a suitable choice for variable modeling. So, the probability distribution of permutation R is estimated according to Eq. (10). Then, promising solutions (EV sequences) are generated through learning and sampling steps described in section 4.1.1.

Due to the shared power supply, EVs interact with each other once they seize a line. So, assigning a charging line to each EV ($1 \leq l_j \leq 3$) depends on other EV's assignments. We use the Markov network to deal

with this behavior, a proper candidate for dealing with variable interactions. Here, each node of the Markov network represents l_j and edges indicate the interaction among EVs. Once the Markov network is constructed, each EV is assigned to a charging line based on the status of its neighboring nodes.

4.2.3. Diversity control mechanism

Generally, elitism selection is used as the selection strategy in EDAs, giving preference to higher-quality solutions (Pelikan et al., 2015). By sampling from these elite populations over generations, population diversity diminishes, and the population becomes very similar (Dehnad, 2012). To control the diversity and avoid being trapped in local optima, we generate a random population with the same length as the current population. When the value of the objective function does not improve in 20 consecutive generations, the random population is added to the current population. Then the new promising solutions are selected based on the elitist strategy.

4.2.4. Heuristic local search method

Here, we present a local search method to improve each sampled solution considering their corresponding R and L . The method moves backward from the current start time to the latest time for each randomly selected EV where the problem constraints are still satisfied. This procedure is repeated until all EVs are chosen.

To shed light on this method, consider the aforementioned

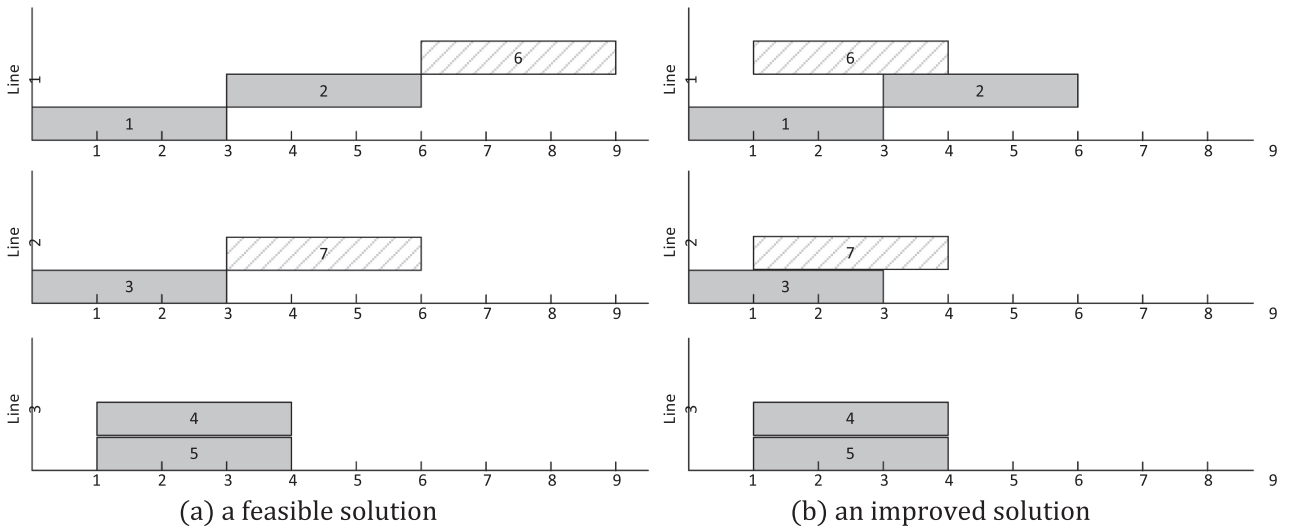


Fig. 4. An example of improving a feasible solution through the heuristic local search method.

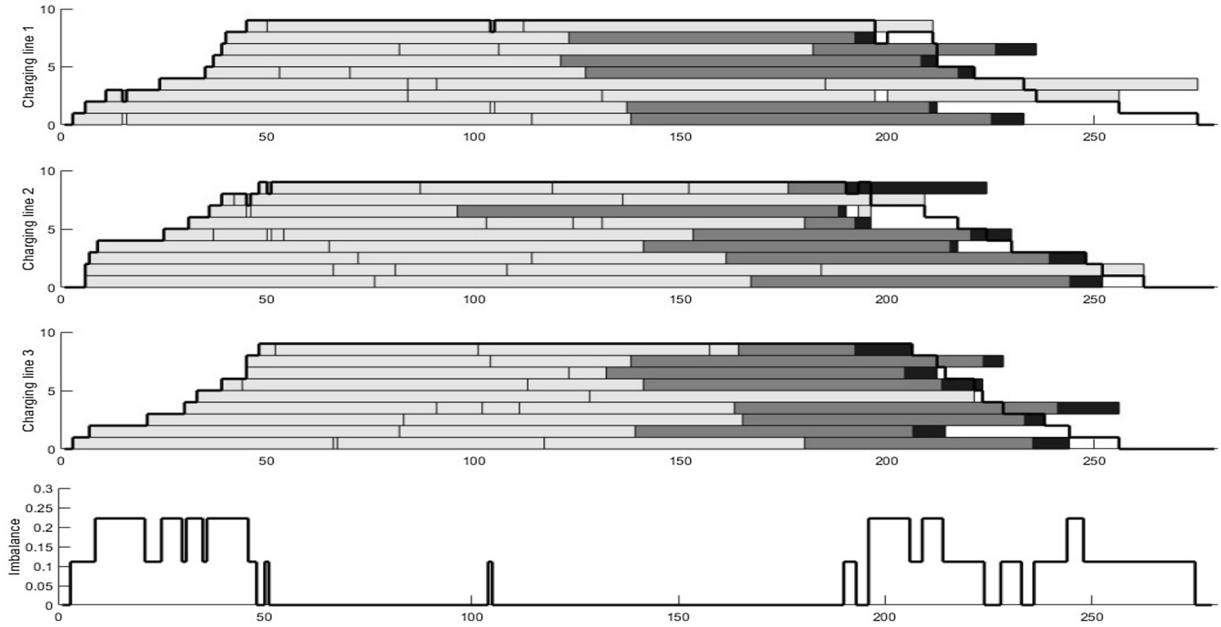


Fig. 5. A feasible scheduling of EVs on three charging lines with minimum tardiness obtained from the proposed HMM-EDA Algorithm.

hypothetical example in 4.2.2. Fig. 4(a) and (b) show one of the corresponding feasible solutions and an improved solution obtained by the local search method, respectively.

4.2.5. The proposed HMM-EDA

Algorithm 4 presents our proposed HMM algorithm to solve the generalized EVCSP while minimizing the total tardiness according to previously introduced concepts and algorithms.

Algorithm 4. HMM-EDA Algorithm for EVCSP.

- 1 **Inputs:**
 - 2 Parameters of Mallows-based EDA method: Parameters in learning, sampling, and selection methods
 - 3 Parameters of Markov Network-based EDA Method: Parameters in learning, sampling, and selection methods
 - 4 Parameters of Algorithm: Population parameters, iteration, and termination conditions
- 5 **Individual encoding:**
 - 6 R : a sequence of numbers ($1 \leq r_i \leq M$) with length M representing EV sequence
 - 7 L : a vector ($1 \leq l_i \leq 3$) with length M representing each EV's line assignment
- 8 **Algorithm:**
 - 9 Initialize variable R of the population using Algorithm 3 and randomly initialize variable L .
 - 10 For each population member, schedule each EV such that the earliest possible starting time is selected
 - 11 while all constraints are satisfied.
 - 12 Calculate the total tardiness for each population member and select promising solutions using elitism.
 - 13 Initialize Mallows probability model for variable R
 - 14 Initialize Markov network probability model for variable L
 - 15 **While** termination criteria not met:
 - 16 Sample solution using Algorithm 1 for variable R
 - 17 Sample solution using Algorithm 2 for variable L
 - 18 Schedule EVs through finding the possible earliest starting time for each EV in sampled R and L .
 - 19 Improve each schedule using the heuristic local search method described in section 4.2.4.
 - 20 Select promising solutions using the diversity control mechanism introduced in section 4.2.3.
 - 21 Update probability models of Mallows and Markov modeling using selected promising solutions
 - 22 **Output:** A feasible sequence of EVs on charging lines with minimum tardiness

In the static variant, the algorithm is applied once to solve the problem for all EVs. However, in the dynamic variant, the algorithm is used for each problem instance (P_n) in which the scheduling is applied for the pending EVs.

Fig. 5 shows feasible scheduling with minimum tardiness of an example obtained by Algorithm 4. This example includes a station with three charging lines with 100 EVs and $N = 9$, $\Delta = 0.25$. The X-axis and the Y-axis corresponds to the problem time horizon and the number of active EVs in each time point, respectively. Light gray rectangles represent EVs that are charged before their due times. While the rest two-colored rectangles are EVs facing delays, the black part represents a portion of the charging time exceeding its due time. The step chart at the bottom of Fig. 5 shows the maximum imbalance between the lines in each time point, which meets the maximum imbalance constraint.

4.3. Proposed constraint programming (CP) model

We compare the performance of our proposed HMM-EDA with state-of-art meta-heuristic methods. We proposed a CP model to provide a baseline solving approach since it provides an optimal solution for our problem. It is worth mentioning that the mathematical programming method is too time-consuming when we have a large-scale problem size (Peng et al., 2014). So MP is impractical for large-scale EVCSP.

The CP model is written in OPL (Optimization Programming Language) and solves problems using "IBM ILOG CP optimizer". The usual way for solving a CP model is to run an automatic search. In this study, we use "Restart" as our search type method in which a depth-first search is applied, which is restarted from time to time, guiding toward an optimal solution (Laborie et al., 2018). The rest of this section discusses the variable definition and problem formulation in the CP model.

4.3.1. Variable definition

In our EVCSP, each EV has a start time and a charging time according to its scheduling behavior. In the CP optimizer, such time intervals are presented by "interval variables" characterized by a starting and ending. Integer expressions "startOf" and "endOf" provide access to the start time and end time for the corresponding interval variable, the scheduled EV (Laborie et al., 2018).

Optionality in interval variables is essential in scheduling problems presented by a Boolean presence status (presenceOf) (Laborie et al.,

2018). In our EVCSP, optionality means that each EV can be left unperformed in charging lines. We define x_{ij} as an optional interval variable representing the j^{th} EV in the i^{th} charging line. Also, the charging time of an EV (c_j) is considered as the size of x_{ij} .

Allocation of scarce resources to time intervals is also another critical characteristic of scheduling. The evolution of a resource (power supply) over time in our EVCSP is modeled by a cumulative resource in which the CP optimizer provides “cumul function expression” for this aim. This expression is used to constrain the temporal evolution of resource usage (Laborie et al., 2018). In our model, the charging task related to each EV increases the usage level of charging lines by one unit. Similarly, once the charging task of the corresponding EV is ended, the usage level of charging lines is decreased by one unit. So, we consider U_i as a cumulative function which represents the usage of i^{th} charging line at any given time:

$$U_i = \sum_{j=1}^N \text{pulse}(x_{ij}, 1) \quad (18)$$

Where $\text{pulse}(x_{ij}, 1)$ is the usage of i^{th} charging line such that x_{ij} increases the power supply usage by 1 unit at its start time of charging process and decreases usage by 1 unit when it releases the resource at its end charging time.

Also, dependent variables are implemented as OPL variable expressions (dexpr) defined through equality constraints. To count the number of active EVs at any given time, we define count_{it} as a dependent variable which is described as follows:

$$\text{count}_{it} = \sum_{j=1}^m (\text{startOf}(x_{ij}) \geq t) \times (\text{endOf}(x_{ij}) < t + 1), \quad (19)$$

$\forall 1 \leq i \leq 3, 1 \leq j \leq M, t \in \text{time horizon}$

4.3.2. CP problem formulation

The proposed CP formulation for our EVCSP is described as follows:

$$\min \sum_{i=1}^3 \sum_{j=1}^M \text{presenceOf}(x_{ij}) \times (\text{endOf}(x_{ij}) - d_j) \quad (20)$$

$$\sum_{i=1}^3 \text{presenceOf}(x_{ij}) = 1 \quad (21)$$

$$\text{presenceOf}(x_{ij}) \times \text{startOf}(x_{ij}) \geq a_j, \quad (22)$$

$\forall 1 \leq i \leq 3, 1 \leq j \leq M$

$$U_i \leq N, \quad \forall 1 \leq i \leq 3 \quad (23)$$

$$\frac{\text{count}_{it}}{N} \leq \Delta, \quad \forall 1 \leq i, j \leq 3, t \in \text{time horizon} \quad (24)$$

The objective function (Eq. (20)) aims to minimize the total tardiness of all interval variables considering their end charging time and their due times with respect to problem constraints which are described in Eq. (21)-(24). The constraint defined in Eq. (21) ensures that an EV appears only in one charging line. Eq. (22) prohibits each EV from starting the charging process before its arrival time. According to constraint in Eq. (23), the number of active EVs at any given time in any given line should not exceed N . At last, Eq. (24) imposes an upper bound on the imbalance between two lines.

5. Experimental study

This section presents our experimental results regarding solving the EV charging scheduling problem using our proposed HMM-EDA method. To do that, we first described the benchmark dataset. Then, the parameters setting of the proposed method is presented. Then, in the last section, we evaluated our proposed HMM-EDA method and compared it

Table 4

Parameter setting of meta-heuristics based on the previous studies parameter settings.

Algorithms	Parameter	Description	Value	Reference
PSO	w	Inertia weight which is used to control the velocity	0.7298	(Clerc & Kennedy, 2002)
	c_1	Personal acceleration coefficient	1.49618	(Clerc & Kennedy, 2002)
	c_2	Social acceleration coefficient	1.49618	(Clerc & Kennedy, 2002)
SA	T_0	Initial temperature	$-\overline{\Delta E}/\ln(\chi_0)$	(Johnson et al., 1991)
ABC	L	Abandonment limit parameter	$0.1 (n_{ABC}^* \times D^{**})$	(Akay & Karaboga, 2009)

* Number of onlooker bees, ** Dimensions of the problem

with the proposed CP model and other state-of-art meta-heuristic methods, including simulated annealing (SA), discrete particle swarm optimization (PSO), and ant artificial bee colony (ABC).

5.1. Benchmark description

The introduced benchmark in this study (García-Álvarez et al., 2018) is used for evolution purposes. This benchmark data is aggregated from a sample station with three charging lines where each line has 60 charging points (180 charging points in total). This benchmark is classified into 72 sets. In each set, EVs' arrival and due times corresponding to the expected behavior of real EV owners in 30 different situations are given. So, there are 2160 samples (problems) in this benchmark. The benchmark is openly available on the web.¹

Each set is represented by a tuple $\{\text{scenario}, \text{type}, N, \Delta\}$. There are three different scenarios; In *scenario 1*, the EV that arrived in the station are distributed such that there are two peak demands in the time horizon. In *scenario 2* and 3, most EVs came to the station in a short time interval. However, in *scenario 3* a more intensive due time is considered compared to *scenario 2*. *Type* value determines the distribution of EVs in charging lines wherein *type 1*, all EVs are distributed uniformly in all charging lines (e.g., one-third of EVs enter each line). In *type 2*, 60 % of EVs enter line 1, 30 % enter line 2, and 10 % enter line 3. N is the maximum number of active EVs that can be charged at any time in each line and it can be one of the values 20, 30 or 40. Δ is the maximum imbalance parameter with values of 0.2, 0.4, 0.6, or 0.8.

We should note that in the introduced benchmark, line allocation is pre-determined. However, in this study, line assignment is also considered as a decision variable. Hence, *type* value in the set tuple is no longer meaningful in our considered problems which result in reducing the number of sets to 36.

5.2. Parameters setting

Fine-tuning of parameters is required for most of the algorithms to obtain desired solutions. In this study, besides our proposed HMM-EDA method, we also apply the state-of-art meta-heuristic methods including SA, PSO, and ABC for solving the studied EVCSP. We set some of these meta-heuristic parameters according to these algorithms' settings in previous studies (Table 4).

To tune the rest of the parameters, both for meta-heuristics and our proposed HMM-EDA method, we use the Taguchi method, a powerful tool for parameter design (Dehnad, 2012). It identifies the best parameters and improves the performance characteristic with orthogonal

¹ Repository section in <https://www.di.uniovi.es/iscop>.

Table 5

The level of factors used for the L27 orthogonal designing method and the best level.

Algorithms	Parameter	Description	Level		
			L	M	H
HMM-EDA	n_{HMM}	HMM-EDA population size	50	100	200
	g_{MN}	MN part: maximum number of neighbors in each clique	5	10	15
	α	MN part: significance parameter to control the density of the edges in the MN	0.5	1.5	3
PSO	n_{PSO}	PSO population size (swarm size)	50	100	200
SA	n_{SA}	SA population size	50	100	200
	m_{SA}	SA number of neighbors per individual	5	10	20
	$SubIter_{SA}$	SA maximum number of sub-iterations	10	20	30
ABC	n_{ABC}	ABC number of onlooker bees	50	100	200

array technique and signal-to-noise (SN) ratios. In this study, the L27 orthogonal designing method is applied, and values for each parameter in each level of factors are tabulated in Table 5. The best level is also shown in bold-face and underlined for each parameter.

5.3. Performance comparison

Since our study is the pioneer in solving this generalized EVCSP problem, there aren't any other methods to compare. Hence, we solved the EVCSP with state-of-the-art metaheuristic methods for comparison purposes, including SA, PSO, and ABC. Also, for achieving an optimal solution, we proposed a CP model. Eventually, we compared our proposed HMM-EDA method with the performance of these methods in

terms of the gap from the best solution with respect to the objective. Notice that the CP model does not meet the optimal solution within each interval (two minutes). Hence, the CP model performance is not considered in the performance comparison of the dynamic variant.

All implementations are run on a laptop with an Intel® Core™ i7- 3 GHz CPU, 32 GB of DDR4-3200 RAM, NVMe PCIe® SSD 1 TB, and Windows 10 PRO. Also, the proposed model was coded using MATLAB R2018a and the CP model was modeled and solved using IBM-ILOG-CPLEX-Optimisation Studio version 12.6. Due to the stochastic nature of the meta-heuristic algorithms, ten independent runs were executed for each instance (problem). Also, considering the dynamic nature of the problem (e.g., in/out EV flows in the station), it is necessary to reschedule EVs every couple of minutes (Garcia Alvarez et al., 2018). Hence, this study considers two-minute (120 s) execution time as the termination condition in our proposed HMM-EDA and meta-heuristic methods. Nevertheless, in the CP model, no termination condition is set to achieve an optimal solution. It should be mentioned that the definition of decision variables and initial population in HMM-EDA and meta-heuristic methods are all the same.

As mentioned earlier, in the introduced benchmark, there are 36 sets comprising 30 instances in which each set is distinguished with its scenarios, and N and Δ values. We solved all 30 EVCSPs in each set in which the total value of the object function (total tardiness) is calculated over all instances. Table 6, Table 7, and Table 8 reported the average value of the computed objective function in each set over ten independent runs. Also, the total objective function value of the CP model is reported in each set as the Obj* in the static problems. The best value for the objective function among all meta-heuristic and HMM-EDA methods is bold-faced and underlined in each table.

According to Table 6-8, our proposed HMM-EDA outperforms all other meta-heuristics, including ABC, PSO, and SA, in every single set in

Table 6

The total objective function values in scenario 1 with corresponding N and Δ values.

N	Δ	Static Problem					Dynamic Problem			
		Obj*	ABC	SA	PSO	HMM-EDA	ABC	SA	PSO	HMM-EDA
20	0.2	934	1777.2	1602.3	1752.1	1495.1	2246.9	2037.5	2236.5	1794.5
20	0.4	645	1251.3	1224.9	1270.9	1112.2	1700.6	1651.8	1755.8	1536.3
20	0.6	513	944.3	934.6	929	921.2	1530.6	1623.8	1640.2	1444.6
20	0.8	498	731.4	741.2	729.4	725.4	1272.2	1369.5	1322.7	1266.5
30	0.2	124	342.2	359.5	344.8	338.6	632.0	671.7	624.7	547.0
30	0.4	11	25.6	26.1	34.2	18.1	159.1	142.7	216.9	84.6
30	0.6	0	10.2	9.8	9.7	5.2	51.6	61.6	66.1	28.4
30	0.8	0	5.1	4.2	4.9	3.9	37.6	32.8	32.3	24.7
40	0.2	0	0	0	0	0	6.2	6.2	6.2	6.2
40	0.4	0	0	0	0	0	4.9	4.9	4.9	4.9
40	0.6	0	0	0	0	0	2.3	2.3	2.3	2.3
40	0.8	0	0	0	0	0	2.3	2.3	2.3	2.3

Obj* refers to the best solution with respect to the objective.

Table 7

The total objective function values in scenario 2 with corresponding N and Δ values.

N	Δ	Static Problem					Dynamic Problem			
		Obj*	ABC	SA	PSO	HMM-EDA	ABC	SA	PSO	HMM-EDA
20	0.2	3811	4609.3	4707.6	4740.4	4586.9	7419.5	5503.2	6061.3	5021.1
20	0.4	3123	3817.3	3791.8	3832.5	3770.9	5869.8	4553.2	4493.2	4423.7
20	0.6	3012	3591.6	3597.2	3585.6	3526.2	5606.0	4548.7	4286.1	4187.2
20	0.8	2988	3386.2	3272.1	3418.1	3183.5	4958.1	3978.3	4048.5	3766.0
30	0.2	433	1107.7	1077.7	1057.6	958	1577.9	1310.0	1315.2	1071.5
30	0.4	362	940.4	958.2	927.2	841.7	1263.2	1273.7	1155.3	988.8
30	0.6	320	807.8	801.4	753	716.4	997.7	1067.0	1012.4	882.9
30	0.8	319	626.9	652.2	684.7	605	834.5	864.8	958.4	796.1
40	0.2	0	121.3	119.3	131	110.1	330.4	425.3	374.6	232.7
40	0.4	0	110.2	102.1	107	85.5	309.8	396.2	332.8	187.9
40	0.6	0	92.2	89.2	95.1	70.2	292.5	345.4	324.3	181.2
40	0.8	0	80.4	84.2	82.6	55.3	249.8	310.7	309.1	139.1

Table 8The total objective function values in scenario 3 with corresponding N and Δ values.

N	Δ	Static Problem					Dynamic Problem			
		Obj*	ABC	SA	PSO	HMM-EDA	ABC	SA	PSO	HMM-EDA
20	0.2	4791	5988.3	5836.7	6064.6	5645	6897.3	7110.0	7231.4	6825.9
20	0.4	4055	5194	5164.8	5218.7	5061.8	6382.7	6398.7	6097.8	5766.6
20	0.6	3934	4789.1	4848.6	4891	4773.6	5772.3	5986.8	5747.0	5685.3
20	0.8	3876	4776.1	4730.2	4864.6	4330.7	5624.7	5665.7	5382.4	4776.3
30	0.2	1068	2272.7	1989.5	2012.5	1964.3	2728.3	2550.4	2595.3	2522.6
30	0.4	925	1752.7	1783.5	1846.9	1712.6	2355.7	2478.2	2468.0	2322.2
30	0.6	927	1724.7	1744.4	1772.5	1712.3	2180.4	2315.5	2304.7	2126.2
30	0.8	934	1701.7	1720.5	1736	1651.9	2140.2	2253.8	2119.2	2118.2
40	0.2	155	751.2	655.6	737.4	650.2	1017.1	893.8	983.0	874.8
40	0.4	155	623.3	557.8	588	522.1	814.7	752.2	750.3	619.2
40	0.6	155	620.2	521.8	574.5	492.2	788.0	719.7	675.5	543.6
40	0.8	155	417.9	471.9	402.3	389.6	483.1	600.9	478.5	445.6

Table 9

Total gap values from the best solution with respect to the objective (CP results) in the static variant.

Scenario No./Method	ABC	SA	PSO	HMM-EDA
scenario1	2362.3	2177.6	2350	1894.7
scenario2	4923.3	4885	5046.8	4141.7
scenario3	9481.9	8895.3	9579	7776.3

terms of total tardiness in both dynamic and static problems.

In addition, in [Table 9](#), total gap values from the best solution with respect to the objective (CP results) are reported in the static variant. Results show the superiority of our proposed HMM-EDA algorithm.

In [Fig. 6](#), the performance of the proposed HMM-EDA method is compared with the CP and the meta-heuristics in all different sets of static problems. Each set is distinguished with a tuple (Scenario, N, Δ).

According to [Fig. 6](#), the solution of scenario 3 is those with higher total tardiness compared to the rest. This results from existing peak demands in the time horizon and intensive due time for EVs. Moreover, with increasing the N and Δ values in each scenario, the total tardiness is decreased due to the constraint on the active charging points' upper bound and power imbalance constraint. From the performance comparison point of view, our proposed HMM-EDA outperforms the meta-heuristics, especially in more complex problems (e.g., scenario 3 with the lowest N and Δ value).

5.4. Performance comparison based on previous studies

To compare the performance of EDA-based methods in solving the EVCSP, which is introduced in ([Garcia Alvarez et al., 2018](#)), we proposed a simpler version of our method. According to the specific problem setting in which EV's line assignment is pre-defined, we used MM-EDA as our solving method since we have only one type of decision variable, i.e., the sequence of EVs.

We compared the MM-EDA with other methods proposed in previous studies ([Garcia Alvarez et al., 2018](#)), including a Memetic Algorithm (MA), Artificial Bee Colony (ABC), a Hybrid Artificial Bee Colony (hABC), and EVS ([Hernández-Arauzo et al., 2015](#)). So, in [Table 10-12](#), we reported the total value of the objective function in these methods compared to our (MM-EDA) corresponding to each scenario. Since line assignments are pre-defined in these samples, each set is characterized with a tuple (*scenario*, *N*, Δ , *type*) in which *type* determines the distribution of EVs in power lines. The best value for the objective function is shown in bold-face and underlined in each table. Also, the number in parenthesis for each value represents the rank in ascending order of all compared method performances.

Based on the results tabulated in [Table 10-12](#), the proposed EDA-based method (MM-EDA) ranked first in most samples.

In [Table 13](#), methods are compared in terms of ranking from their objective function values. Notice that EVS and ABC method's results are not reported for all instances. Also, Their performance in the reported instances is not among the best; hence, we discarded them in the rank

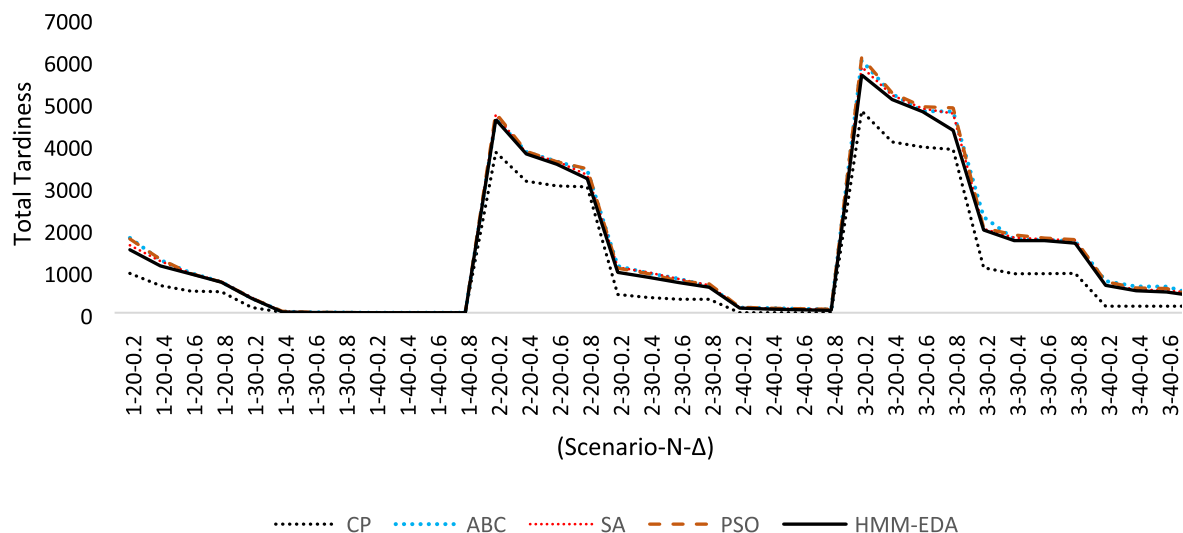
**Fig. 6.** Performance comparison of the proposed HMM-EDA method with CP and the meta-heuristics in all different sets in the static problems.

Table 10

The total objective function values in scenario 1 with the corresponding type, N, and Δ values (based on the EVCSF introduced in (Garcia Alvarez et al., 2018)). Value in parenthesis represents the rank of its corresponding method.

N	Δ	Static Problem				Dynamic Problem				
		MA	ABC	hABC	MM-EDA	EVS	MA	ABC	hABC	MM-EDA
Type 1 instances										
20	0.2	5210.8 (3)	5331.7 (4)	5089 (2)	<u>5028.7 (1)</u>	8386.3 (5)	6981.2 (3)	7027.9 (4)	6948.1 (2)	<u>6743.8 (1)</u>
20	0.4	2509.2 (2)	2586.1 (4)	2529.7 (3)	<u>2502.2 (1)</u>	4120.4 (5)	3884.2 (3)	3898.1 (4)	3877.4 (2)	<u>3576.5 (1)</u>
20	0.6	2252.7 (3)	2258.2 (4)	2250.6 (2)	<u>2242.6 (1)</u>	3670.6 (5)	3556.7 (3)	3558 (4)	3547.8 (2)	<u>3312.2 (1)</u>
20	0.8	2199 (3)	2200.4 (4)	2194.1 (2)	<u>2184.7 (1)</u>	3590.9 (5)	3503.3 (3)	3509.1 (4)	3494.5 (2)	<u>3481.4 (1)</u>
30	0.2	729.5 (3)	771.9 (4)	<u>712.6 (1)</u>	721.1 (2)	1959.3 (5)	1444.8 (2)	1445.7 (4)	1445 (3)	<u>1269.4 (1)</u>
30	0.4	<u>52.5 (1)</u>	76.8 (4)	75.4 (3)	65.2 (2)	421.2 (5)	367.6 (4)	366.6 (3)	364.8 (2)	<u>354.5 (1)</u>
30	0.6	<u>34.3 (1)</u>	34.9 (4)	34.4 (2)	34.4 (2)	347.9 (5)	316.2 (2)	317.7 (4)	316.8 (3)	<u>312.3 (1)</u>
30	0.8	33.8 (2)	34.1 (4)	<u>33.7 (1)</u>	33.9 (3)	347.6 (5)	<u>314.7 (1)</u>	316.4 (4)	315.9 (3)	315.6 (2)
40	0.2	<u>21(2),9 (1)</u>	238.6 (4)	221.4 (3)	219.3 (2)	735 (5)	541.6 (3)	545.2 (4)	540.3 (2)	<u>528 (1)</u>
40	0.4	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	14 (5)	7.7 (3)	7.8 (4)	<u>7.4 (1)</u>	<u>7.4 (1)</u>
40	0.6	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>
40	0.8	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	<u>0 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>	<u>3.4 (1)</u>
Type 2 instances										
20	0.2	122,615 (2)	123,409 (4)	122,690 (3)	<u>122521.2 (1)</u>	128,185 (5)	124,075 (3)	124,168 (4)	123,934 (2)	<u>123672.8 (1)</u>
20	0.4	43991.1 (2)	44152.7 (4)	44,030 (3)	<u>43781.5 (1)</u>	46319.3 (5)	45127.2 (3)	45183.9 (4)	45,104 (2)	<u>44811.4 (1)</u>
20	0.6	20526.9 (2)	20,597 (4)	20545.1 (3)	<u>20511.3 (1)</u>	22966.8 (5)	21822.6 (3)	21847.7 (4)	21788.5 (2)	<u>21566.7 (1)</u>
20	0.8	12762.8 (4)	12734.1 (3)	<u>12727.2 (1)</u>	12731.2 (2)	14573.1 (5)	14209.9 (3)	14212.1 (4)	14176.8 (2)	<u>13933.7 (1)</u>
30	0.2	<u>69610.6 (1)</u>	69954.3 (4)	69665.3 (2)	69851.4 (3)	72860.8 (5)	70894.5 (3)	70942.6 (4)	70853.6 (2)	<u>70657.7 (1)</u>
30	0.4	20031.7 (2)	20097.6 (4)	20043.6 (3)	<u>20009.6 (1)</u>	21479.9 (5)	21130.3 (3)	21150.5 (4)	21109.1 (2)	<u>21079.4 (1)</u>
30	0.6	7030.5 (3)	7033 (4)	<u>7019 (1)</u>	7025.3 (2)	8088.9 (5)	7921.5 (3)	7923 (4)	7905.2 (2)	<u>7773.6 (1)</u>
30	0.8	<u>3527.9 (1)</u>	3536.4 (4)	3530.8 (3)	3529.3 (2)	4486.3 (5)	4389.3 (3)	4391 (4)	4384.4 (2)	<u>4350 (1)</u>
40	0.2	43981.8 (2)	44146.7 (4)	44024.5 (3)	<u>43952.5 (1)</u>	46135.4 (5)	45139.2 (3)	45192.6 (4)	45113.7 (2)	<u>45047.6 (1)</u>
40	0.4	9760.7 (2)	9775.7 (4)	<u>9753.1 (1)</u>	9764.2 (3)	10869.3 (5)	10654.6 (3)	10669.2 (4)	<u>10635.7 (1)</u>	10654.3 (2)
40	0.6	2851.1 (2)	2855.7 (4)	<u>2850.4 (1)</u>	2852.2 (3)	3599.1 (5)	3515.6 (4)	3515.1 (3)	<u>3514.8 (1)</u>	3514.9 (2)
40	0.8	872.3 (3)	876.3 (4)	872 (2)	<u>869.1 (1)</u>	1635.5 (5)	1572.1 (3)	1574.9 (4)	1571.7 (2)	<u>1564.9 (1)</u>

Table 11

The total objective function values in scenario 2 with the corresponding type, N, and Δ values (based on the EVCSF introduced in (Garcia Alvarez et al., 2018)). Value in parenthesis represents the rank of its corresponding method.

N	Δ	Static Problem			Dynamic Problem			
		MA	hABC	MM-EDA	EVS	MA	ABC	MM-EDA
Type 1 instances								
20	0.2	14411.5 (3)	14097.9 (1)	14382.4 (2)	16,886 (4)	16176.3 (3)	15933.8 (1)	16055.7 (2)
20	0.4	11703.6 (1)	11713.9 (3)	11705.2 (2)	14131.2 (4)	13915.9 (3)	13869.9 (1)	13888.7 (2)
20	0.6	11328.3 (2)	11335.7 (3)	11320.3 (1)	13648.3 (4)	13559.1 (3)	13534.8 (2)	13,533 (1)
20	0.8	11242.1 (2)	11242.9 (3)	11240.1 (1)	13547.9 (4)	13492.7 (3)	13464.7 (1)	13475.8 (2)
30	0.2	3920.6 (3)	3887.3 (2)	3880.6 (1)	6394.7 (4)	5821.5 (3)	5798.8 (2)	5775.5 (1)
30	0.4	2726.2 (3)	2717.1 (1)	2720.2 (2)	4824.3 (4)	4688.4 (3)	4687.2 (2)	4685.3 (1)
30	0.6	2592.8 (3)	2588.3 (2)	2580.3 (1)	4668.6 (4)	4560.3 (3)	4559.3 (2)	4541.8 (1)
30	0.8	2591.4 (3)	2587.2 (2)	2585.9 (1)	4672.6 (4)	4559 (3)	4556.7 (2)	4551.9 (1)
40	0.2	638 (1)	662.1 (3)	642 (2)	1951.4 (4)	1593.5 (3)	1590 (2)	1572.6 (1)
40	0.4	186.4 (1)	186.4 (1)	186.4 (1)	1212.7 (4)	1102 (3)	1101.5 (2)	1088.9 (1)
40	0.6	172.2 (3)	171.5 (2)	170.5 (1)	1210.7 (4)	1103.3 (3)	1102.8 (2)	1094.4 (1)
40	0.8	171.9 (2)	171.6 (1)	171.9 (2)	1210.6 (4)	1102.7 (2)	1102.5 (1)	1103.6 (3)
Type 2 instances								
20	0.2	137,204 (2)	137,271 (3)	137195.2 (1)	143,883 (4)	139,192 (3)	138,961 (1)	139033.7 (2)
20	0.4	57453.4 (3)	57295.2 (1)	57352.1 (2)	62246.2 (4)	59,194 (3)	59065.2 (1)	59106.7 (2)
20	0.6	34548.2 (3)	34499.5 (2)	34,452 (1)	39869.4 (4)	36298.4 (3)	36231.7 (2)	36189.6 (1)
20	0.8	27735.1 (3)	27,711 (1)	27725.2 (2)	30887.2 (4)	29,791 (3)	29747.5 (1)	29771.6 (2)
30	0.2	82,970 (2)	83224.2 (3)	82851.2 (1)	86,392 (4)	84,470 (3)	84297.4 (2)	84134.3 (1)
30	0.4	32714.9 (2)	32659.2 (1)	32725.2 (3)	34475.4 (4)	33936.8 (3)	33891.5 (2)	33626.5 (1)
30	0.6	17077.3 (2)	17069.9 (1)	17085.3 (3)	19355.7 (4)	18709.8 (3)	18682.5 (1)	18,709 (2)
30	0.8	12364.5 (3)	12359.5 (2)	12325.2 (1)	14375.1 (4)	14201.7 (3)	14182.4 (2)	14149.8 (1)
40	0.2	57,652 (3)	57,630 (1)	57642.2 (2)	59,775 (4)	58714.1 (3)	58591.2 (1)	58653.9 (2)
40	0.4	20786.9 (2)	20765.5 (1)	20800.3 (3)	22,254 (4)	21984.2 (3)	21947.1 (2)	21783.1 (1)
40	0.6	9582.4 (3)	9581.7 (2)	9581.5 (1)	10991.3 (4)	10815.9 (3)	10808.5 (2)	10715.8 (1)
40	0.8	5806 (1)	5814.1 (3)	5812.2 (2)	7260.5 (4)	7178.5 (3)	7170.9 (2)	7119.2 (1)

comparison reported in Table 13.

As shown in Table 13, MM-EDA outperforms the other techniques and shows the superiority of EDA-based methods in solving EVCSFs.

To show the superiority of the proposed MM-EDA over other methods, the objective function values obtained from each method are subtracted from the best answer in each set. The sum of the total gap from the best values on all data is reported in the third row of Table 13.

Best values are underlined and bold-faced in Table 10-12. Notice that the smaller this metric is, the better the corresponding method performance is. Hence, considering the problem's objective (minimizing the total tardiness) and real-world problem conditions, the proposed MM-EDA shows significant improvement in solving the introduced EVCSF.

Table 12

The total objective function values in scenario 3 with the corresponding type, N, and Δ values (based on the EVCSP introduced in (Garcia Alvarez et al., 2018)). Value in parenthesis represents the rank of its corresponding method.

N	Δ	Static Problem			Dynamic Problem			
		MA	hABC	MM-EDA	EVS	MA	ABC	MM-EDA
Type 1 instances								
20	0.2	19209.6 (3)	18,874 (2)	<u>18752.9 (1)</u>	20704.7 (4)	20242.4 (3)	20055.1 (2)	<u>19752.4 (1)</u>
20	0.4	16628.5 (2)	16632.1 (3)	<u>16625.2 (1)</u>	18001.7 (4)	17916.8 (3)	17875.2 (2)	<u>17869.7 (1)</u>
20	0.6	16218.8 (1)	16227.3 (2)	16229.2 (3)	<u>17528.2 (1)</u>	17573.4 (4)	17553.6 (2)	17560.3 (3)
20	0.8	16136.2 (3)	16134.3 (2)	<u>16130.3 (1)</u>	<u>17463.3 (1)</u>	17524.5 (4)	17505.9 (3)	17500.5 (2)
30	0.2	7791.9 (3)	<u>7733 (1)</u>	7765.6 (2)	9051.1 (4)	8569.1 (3)	8556.5 (2)	<u>8482.4 (1)</u>
30	0.4	6528 (3)	<u>6524.1 (1)</u>	6526.2 (2)	7347.3 (4)	7283.5 (3)	7276.2 (2)	<u>6950.1 (1)</u>
30	0.6	<u>6379.6 (1)</u>	6380.1 (2)	6382.1 (3)	7150.4 (4)	7115.3 (2)	<u>7113.5 (1)</u>	7116 (3)
30	0.8	6371.6 (3)	6371.2 (2)	<u>6370.9 (1)</u>	7144.5 (4)	7106.9 (3)	7103.8 (2)	<u>7103.2 (1)</u>
40	0.2	2422.5 (2)	2438.5 (3)	<u>2420.6 (1)</u>	3478.1 (4)	3106.7 (2)	3108.5 (3)	<u>3095.9 (1)</u>
40	0.4	<u>1853.4 (1)</u>	<u>1853.4 (1)</u>	<u>1853.4(1)</u>	2373.1 (4)	2328.6 (2)	2329.6 (3)	<u>2311.8 (1)</u>
40	0.6	1810.3 (3)	<u>1810 (1)</u>	<u>1810 (1)</u>	2276.7 (4)	2247.6 (3)	2247.2 (2)	<u>2228.9 (1)</u>
40	0.8	1810.2 (3)	<u>1810 (1)</u>	<u>1810 (1)</u>	2276.7 (4)	2246.7 (3)	2246.4 (2)	<u>2228.2 (1)</u>
Type 2 instances								
20	0.2	129,339 (3)	<u>129,171 (1)</u>	129175.2 (2)	138,064 (4)	131,037 (3)	130,903 (2)	<u>130822.2 (1)</u>
20	0.4	56,672 (3)	<u>56513.3 (1)</u>	56612.8 (2)	62988.8 (4)	58225.8 (3)	<u>58080.9 (1)</u>	58101.7 (2)
20	0.6	37089.8 (3)	37034.4 (2)	<u>37012.5 (1)</u>	42528.3 (4)	38574.6 (3)	38507.2 (2)	<u>38455.7 (1)</u>
20	0.8	31715.6 (3)	<u>31705.2 (1)</u>	31710.7 (2)	33998.2 (4)	33360.3 (3)	<u>33313.8 (1)</u>	33314.1 (2)
30	0.2	<u>79182.4 (1)</u>	79359.1 (3)	79192.8 (2)	83169.5 (4)	80526.7 (3)	80393.4 (2)	<u>80314.4 (1)</u>
30	0.4	33084.2 (3)	33021.5 (2)	<u>33011.6 (1)</u>	34799.7 (4)	34045.1 (3)	33990.5 (2)	<u>33948.1 (1)</u>
30	0.6	19592.6 (3)	<u>19559.8 (1)</u>	19585.5 (2)	21321.1 (4)	20645.1 (3)	<u>20622.7 (1)</u>	20632.5 (2)
30	0.8	15812.5 (3)	15805.1 (2)	<u>15795.6 (1)</u>	16,972 (3)	16979.7 (4)	16962.8 (2)	<u>16948.6 (1)</u>
40	0.2	55832.1 (3)	55747.9 (2)	<u>55735.9 (1)</u>	58306.3 (4)	56737.6 (3)	56635.1 (2)	<u>56580.2 (1)</u>
40	0.4	21,932 (3)	<u>21901.8 (1)</u>	21909.6 (2)	22988.7 (4)	22710.2 (3)	22680.2 (2)	<u>22453.6 (1)</u>
40	0.6	11462.9 (3)	11455.3 (2)	<u>11452.3 (1)</u>	12220.3 (4)	12168.1 (3)	12161.5 (2)	<u>12039.8 (1)</u>
40	0.8	8388.2 (3)	8387.9 (2)	<u>8387.3 (1)</u>	9127.3 (2)	9204.6 (4)	9198.8 (3)	<u>9114.1 (1)</u>

Table 13

Method comparison in terms of ranking from their objective function perspective in solving the introduced EVCSP in (Garcia Alvarez et al., 2018).

Methods	MA	hABC	MM-EDA
Frequency of Rank 1st	19	47	93
Average Rank	2.6	1.9	1.5
Sum of gaps from the best values	10515.1	6697.9	1491.9
Final Ranking	3	2	1

6. Conclusion

This study aims to solve an EVCSP inspired by an actual charging station while minimizing all EVs' charging delays (tardiness). The most important constraint in this problem is balancing power consumption between charging lines, limiting simultaneous charging EVs. Usually, line allocation is pre-determined in EVCSPs. However, in this study, line assignment is also considered as a decision variable. Therefore, considering the objective function in our EVCSP, minimizing the total tardiness, we modeled our problem using two decision variables: a vector representing a sequence of all EVs; and a vector representing an assignment of each EV to a charging line.

Several methods have been proposed so far for solving this problem. Deterministic optimization models such as CP or linear programming (LP) usually suffer from the curse of dimensionality (De Farias, 2002). Thus, these models require a long processing time to find the optimal solution in EVCSP due to the high dimensions of decision variables in real-world situations (the introduced benchmark, which includes 180 EVs). Accordingly, metaheuristic algorithms are proper choices to obtain good results in a reasonable computational time. We proposed a method based on EDAs. This is because of existing possible interrelationships among EVs during scheduling procedures and the competency of EDAs in handling these relations among decision variables. Our proposed method is a hybrid EDA in which a Markov network and a Mallows model are simultaneously used, called "HMM-EDA". Markov network is a suitable approach for dealing with interaction among variables. So, it is proposed to handle charging line assignment

while considering these interactions. In addition, we proposed a Mallows-based-EDA since the Mallows model is suitable for permutation problems by considering EV sequence as a permutation problem.

Our method is evaluated by simulation on a benchmark dataset comprising several examples with different characteristics. We compared our method with a constraint programming (CP) model and the state-of-art meta-heuristic methods in terms of the objective function value. Results show the superiority of our proposed method in terms of the gap from the best solution with respect to the objective compared to other meta-heuristics.

It remains for future work to investigate the performance of our proposed HMM-EDA method while considering other constraints such as EV state-of-charge and slow/fast charging EVs (introduced in Table 2) in the EVCSP formulation. In addition, other objective functions such as charging price (Savari et al., 2019) combined with total tardiness could also be considered in future studies to examine our proposed HMM-EDA method in multi-objective optimization.

CRedit authorship contribution statement

Hadi Shahmoradi: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization. **Majid Esmaelian:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Resources, Data curation, Writing – original draft, Writing – review & editing, Visualization, Supervision, Project administration. **Hossein Karshenas:** Conceptualization, Methodology, Validation, Formal analysis, Investigation, Writing – original draft, Writing – review & editing.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

- Acar, C., & Dincer, I. (2020). The potential role of hydrogen as a sustainable transportation fuel to combat global warming. *International Journal of Hydrogen Energy*, 45(5), 3396–3406.
- Akay, B., & Karaboga, D. (2009). Parameter tuning for the artificial bee colony algorithm. *International Conference on Computational Collective Intelligence*, 608–619.
- Al-Ogailli, A. S., Hashim, T. J. T., Rahmat, N. A., Ramasamy, A. K., Marsadek, M. B., Faisal, M., & Hannan, M. A. (2019). Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations. *IEEE Access*, 7, 128353–128371.
- Amiri, M. F., & Behnamian, J. (2020). Multi-objective green flowshop scheduling problem under uncertainty: Estimation of distribution algorithm. *Journal of Cleaner Production*, 251, Article 119734.
- Arrieta, E. I. (2014). *Sampling and learning distance-based probability models for permutation spaces*. Universidad del País Vasco-Euskal Herriko Unibertsitatea.
- Bartholdi, J., Tovey, C. A., & Trick, M. A. (1989). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, 6(2), 157–165.
- Bukchin, Y., & Raviv, T. (2018). Constraint programming for solving various assembly line balancing problems. *Omega*, 78, 57–68.
- Ceberio, J., Irurizki, E., Mendiburu, A., & Lozano, J. A. (2012). A review on estimation of distribution algorithms in permutation-based combinatorial optimization problems. *Progress in Artificial Intelligence*, 1(1), 103–117.
- Ceberio, J., Irurizki, E., Mendiburu, A., & Lozano, J. A. (2013). A distance-based ranking model estimation of distribution algorithm for the flowshop scheduling problem. *IEEE Transactions on Evolutionary Computation*, 18(2), 286–300.
- Ceberio, J., Mendiburu, A., & Lozano, J. A. (2011). Introducing the mallows model on estimation of distribution algorithms. *International Conference on Neural Information Processing*, 461–470.
- Chen, J., Wu, Y., Xu, C., Song, M., & Liu, X. (2019). Global non-fossil fuel consumption: Driving factors, disparities, and trends. *Management Decision*.
- Chung, H.-M., Li, W.-T., Yuen, C., Wen, C.-K., & Crespi, N. (2018). Electric vehicle charge scheduling mechanism to maximize cost efficiency and user convenience. *IEEE Transactions on Smart Grid*, 10(3), 3020–3030.
- Clerc, M., & Kennedy, J. (2002). The particle swarm-explosion, stability, and convergence in a multidimensional complex space. *IEEE Transactions on Evolutionary Computation*, 6(1), 58–73.
- Dai, M., Zhang, Z., Giret, A., & Salido, M. A. (2019). An enhanced estimation of distribution algorithm for energy-efficient job-shop scheduling problems with transportation constraints. *Sustainability*, 11(11), 3085.
- De Farias, D. P. (2002). *The linear programming approach to approximate dynamic programming: Theory and application*. Ph. D. Thesis, Stanford University.
- Dehnad, K. (2012). *Quality control, robust design, and the Taguchi method*. Springer Science & Business Media.
- Diaconis, P. (1988). Group representations in probability and statistics. *Lecture Notes-Monograph Series*, 11, i–192.
- Dincbas, M. (1988). Solving a cutting-stock problem in constraint logic programming. In *Proc. of the Fifth International Conference and Symposium on Logic Programming* (pp. 42–58).
- Du, Y., Li, J., Luo, C., & Meng, L. (2021). A hybrid estimation of distribution algorithm for distributed flexible job shop scheduling with crane transportations. *Swarm and Evolutionary Computation*, 62, Article 100861.
- Esmaelian, M., Sobhani, A., Shahmoradi, H., & Mohammadi, M. (2021). Scheduling the capacitated identical parallel machines problem: A new formulation with sequence-dependent setup costs and different due dates. *European Journal of Industrial Engineering*, 15(5), 643–674.
- García-Álvarez, J., González, M. A., & Vela, C. R. (2018). Metaheuristics for solving a real-world electric vehicle charging scheduling problem. *Applied Soft Computing*, 65, 292–306.
- García-Álvarez, J., González, M. A., & Vela, C. R. (2015). A genetic algorithm for scheduling electric vehicle charging. In *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (pp. 393–400).
- García Álvarez, J., González, M.A., Rodríguez Vela, C., & Varela, R. (2018). Electric vehicle charging scheduling by an enhanced artificial bee colony algorithm. *Energies*, 11(10), 2752.
- Hao, X., Lin, L., & Gen, M. (2014). An effective multi-objective EDA for robust resource constrained project scheduling with uncertain durations. *Procedia Computer Science*, 36, 571–578.
- Hauschild, M., & Pelikan, M. (2011). An introduction and survey of estimation of distribution algorithms. *Swarm and Evolutionary Computation*, 1(3), 111–128.
- Hernández-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.
- Hernández Arauzo, A., Puente Peinador, J., González, M. A., Varela Arias, J. R., & Sedano Franco, J. (2013). Dynamic scheduling of electric vehicle charging under limited power and phase balance constraints. *Proceedings of SPARK 2013-Scheduling and Planning Applications Workshop*.
- Jian, L., Zheng, Y., & Shao, Z. (2017). High efficient valley-filling strategy for centralized coordinated charging of large-scale electric vehicles. *Applied Energy*, 186, 46–55. <https://doi.org/10.1016/j.apenergy.2016.10.117>
- Jin, C., Tang, J., & Ghosh, P. (2013). Optimizing electric vehicle charging: A customer's perspective. *IEEE Transactions on Vehicular Technology*, 62(7), 2919–2927.
- Johnson, D. S., Aragon, C. R., McGeoch, L. A., & Schevon, C. (1991). Optimization by simulated annealing: An experimental evaluation; part II, graph coloring and number partitioning. *Operations Research*, 39(3), 378–406.
- Kacprzyk, J., & Pedrycz, W. (2015). *Springer handbook of computational intelligence*. Springer.
- Koltai, T., Dimény, I., Gallina, V., Gaal, A., & Sepe, C. (2021). An analysis of task assignment and cycle times when robots are added to human-operated assembly lines, using mathematical programming models. *International Journal of Production Economics*, 242, Article 108292.
- Koufakis, A.-M., Rigas, E. S., Bassiliades, N., & Ramchurn, S. D. (2019). Offline and online electric vehicle charging scheduling with V2V energy transfer. *IEEE Transactions on Intelligent Transportation Systems*, 21(5), 2128–2138.
- Laborie, P., Rogerie, J., Shaw, P., & Vilím, P. (2018). IBM ILOG CP optimizer for scheduling. *Constraints*, 23(2), 210–250.
- Larrañaga, P., & Lozano, J. A. (2001). *Estimation of distribution algorithms: A new tool for evolutionary computation* (Vol. 2). Springer Science & Business Media.
- Liu, C., Chen, H., Xu, R., & Wang, Y. (2018). Minimizing the resource consumption of heterogeneous batch-processing machines using a copula-based estimation of distribution algorithm. *Applied Soft Computing*, 73, 283–305.
- Liu, H., Gao, L., & Pan, Q. (2011). A hybrid particle swarm optimization with estimation of distribution algorithm for solving permutation flowshop scheduling problem. *Expert Systems with Applications*, 38(4), 4348–4360.
- Ma, Z., Callaway, D. S., & Hiskens, I. A. (2011). Decentralized charging control of large populations of plug-in electric vehicles. *IEEE Transactions on Control Systems Technology*, 21(1), 67–78.
- Mavrouniotis, M., Ellinas, G., & Polycarpou, M. (2019). Electric Vehicle Charging Scheduling Using Ant Colony System. *IEEE Congress on Evolutionary Computation (CEC)*, 2019, 2581–2588.
- Meila, M., Phadnis, K., Patterson, A., & Bilmes, J. A. (2012). Consensus ranking under the exponential model. *ArXiv Preprint*.
- Mencia, C., Sierra, M. R., Mencia, R., & Varela, R. (2017). Genetic algorithm for scheduling charging times of electric vehicles subject to time dependent power availability. *International Work-Conference on the Interplay Between Natural and Artificial Computation*, 160–169.
- Mirzaei, M. J., Kazemi, A., & Homaei, O. (2014). RETRACTED: Real-world based approach for optimal management of electric vehicles in an intelligent parking lot considering simultaneous satisfaction of vehicle owners and parking operator. *Energy*, 76, 345–356.
- Oliveira, B. B., & Carravilla, M. A. (2017). Understanding complexity in a practical combinatorial problem using mathematical programming and constraint programming. *Congress of APDIO, the Portuguese Operational Research Society*, 269–295.
- Pelikan, M., Hauschild, M. W., & Lobo, F. G. (2015). Estimation of distribution algorithms. In *Springer Handbook of Computational Intelligence* (pp. 899–928). Springer.
- Peng, Y., Lu, D., Chen, Y., et al. (2014). A constraint programming method for advanced planning and scheduling system with multilevel structured products. *Discrete Dynamics in Nature and Society*.
- Pérez-Rodríguez, R., & Hernández-Aguirre, A. (2019). A hybrid estimation of distribution algorithm for the vehicle routing problem with time windows. *Computers & Industrial Engineering*, 130, 75–96.
- Qian, B., Li, Z., & Hu, R. (2017). A copula-based hybrid estimation of distribution algorithm for m-machine reentrant permutation flow-shop scheduling problem. *Applied Soft Computing*, 61, 921–934.
- Savari, G. F., Krishnasamy, V., Sugavanam, V., & Vakesan, K. (2019). Optimal charging scheduling of electric vehicles in micro grids using priority algorithms and particle swarm optimization. *Mobile Networks and Applications*, 24(6), 1835–1847.
- Shakya, S., Santana, R., & Lozano, J. A. (2012). A markovianity based optimisation algorithm. *Genetic Programming and Evolvable Machines*, 13(2), 159–195.
- Shao, W., Pi, D., & Shao, Z. (2019). A Pareto-Based Estimation of Distribution Algorithm for Solving Multiobjective Distributed No-Wait Flow-Shop Scheduling Problem With Sequence-Dependent Setup Time. *IEEE Transactions on Automation Science and Engineering*.
- Tiwari, A., Chang, P.-C., Tiwari, M. K., & Kollanoor, N. J. (2015). A Pareto block-based estimation and distribution algorithm for multi-objective permutation flow shop scheduling problem. *International Journal of Production Research*, 53(3), 793–834.
- Tzeng, Y.-R., Chen, C.-L., & Chen, C.-L. (2012). A hybrid EDA with ACS for solving permutation flow shop scheduling. *The International Journal of Advanced Manufacturing Technology*, 60(9), 1139–1147.
- Wang, K., Choi, S. H., & Lu, H. (2015). A hybrid estimation of distribution algorithm for simulation-based scheduling in a stochastic permutation flowshop. *Computers & Industrial Engineering*, 90, 186–196.
- Wang, S., Wang, L., Liu, M., & Xu, Y. (2013). An effective estimation of distribution algorithm for solving the distributed permutation flow-shop scheduling problem. *International Journal of Production Economics*, 145(1), 387–396.
- Wang, Z., Jochem, P., & Fichtner, W. (2020). A scenario-based stochastic optimization model for charging scheduling of electric vehicles under uncertainties of vehicle availability and charging demand. *Journal of Cleaner Production*, 254, Article 119886.
- Wilberforce, T., El-Hassan, Z., Khatib, F. N., Al Makky, A., Baroutaji, A., Carton, J. G., & Olabi, A. G. (2017). Developments of electric cars and fuel cell hydrogen electric cars. *International Journal of Hydrogen Energy*, 42(40), 25695–25734.
- Xue, Y., Rui, Z., Yu, X., Sang, X., & Liu, W. (2019). Estimation of distribution evolution memetic algorithm for the unrelated parallel-machine green scheduling problem. *Memetic Computing*, 11(4), 423–437.
- Zangari, M., Mendiburu, A., Santana, R., & Pozo, A. (2017). Multiobjective decomposition-based Mallows Models estimation of distribution algorithm. A case of

- study for permutation flowshop scheduling problem. *Information Sciences*, 397, 137–154.
- Zhou, K., Cheng, L., Wen, L., Lu, X., & Ding, T. (2020). A coordinated charging scheduling method for electric vehicles considering different charging demands. *Energy*, 213, Article 118882.
- Zhu, M., Liu, X.-Y., Kong, L., Shen, R., Shu, W., & Wu, M.-Y. (2014). The charging-scheduling problem for electric vehicle networks. *IEEE Wireless Communications and Networking Conference (WCNC)*, 2014, 3178–3183.