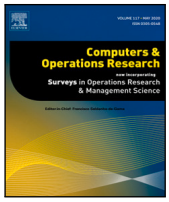
Computers & Operations Research 157 (2023) 106261

Contents lists available at ScienceDirect

Computers and Operations Research

journal homepage: www.elsevier.com/locate/cor



A matheuristic for the electric vehicle routing problem with time windows and a realistic energy consumption model

M. Bruglieri [a,*], M. Paolucci [b], O. Pisacane [c]

[a] Dipartimento di Design, Politecnico di Milano, Via Giovanni Durando 10, 20158, Milano, Italy
[b] Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi, Università degli Studi di Genova, Via Opera Pia 13, 16145, Genova, Italy
[c] Dipartimento di Ingegneria dell'Informazione, Università Politecnica delle Marche, Via Brecce Bianche 12, 60131, Ancona, Italy

ARTICLE INFO

Keywords:
Routing
Variable vehicle speed
Mixed integer linear program
Cloneless formulation
Matheuristic

ABSTRACT

We address an Electric Vehicle Routing Problem with Time Windows (E-VRPTW) considering several real-like factors in the energy consumption model, e.g., the payload and the vehicle speed. We model E-VRPTW as a Mixed Integer Linear Program (MILP) where the speeds of vehicles are continuous variables that can vary between a minimum and a maximum value. Moreover, the proposed MILP formulation is *cloneless* since it allows using more than once the same recharging station without introducing dummy copies of it. To efficiently solve large-sized instances of the problem, we design a Random Kernel Search (RKS) matheuristic approach, based on the cloneless MILP formulation, that in turn exploits another matheuristic, called Random k-Degree Search (RkDS), to generate an initial feasible solution. We compare the results produced by a MILP solver using the MILP formulation with the ones obtained by the RKS on instances up to 100 customers derived from the benchmark instances of E-VRPTW. We show that the proposed matheuristic outperforms the cloneless MILP formulation on the medium/large-sized instances and also that it is robust, being not significantly sensitive to the values of the parameters used by the RkDS to generate the initial solution and to the initial time limit for the restricted MILP models.

1. Introduction

In the first half of 2021, the sales volume of Electric Vehicles (EVs) has grown of about 168% with respect to 2020, although the low base of 2020 (14% less than in 2019) was especially due to the pandemic (www.ev-volumes.com/). Indeed, the recent policies of the European Commission (EC) are going to promote the use of Alternative Fuel Vehicles, like EVs, in place of the traditional Internal Combustion Engine Vehicles (ICEVs). The aim of the EC is to reduce the GreenHouse Gas (GHG) emissions, in particular the ones due to the road transport, that amount to about 70% of the total GHG emissions produced by the whole transport sector.

In some ways, the EVs may be the answer to the need to reduce such GHG emissions. From the *environmental sustainability* standpoint, in fact, an EV guarantees a significant reduction of the CO_2 emissions, assuring a total elimination of local emissions due to both NOx and Pmx. This positively impacts on the citizens quality of life too (*social sustainability*). From an *economic sustainability* perspective, the maintenance cost of EVs is by far less than that of ICEVs, resulting in a cost saving of about 50%. In addition, the cost for kilometer of an EV is usually by far less than the one of an ICEV; for example, since in Italy the energy cost is about 0.20€ for KWh against the 1.50€ for liter of gasoline, and that an electric midsize sedan car consumes about 14 kWh for 100 km, against 4.5 gasoline liters for an internal combustion one, the cost for 100 km is about 2.8€ for the electric against 6.75€ for the internal combustion. Finally, an EV can enter also the so called Limited Traffic Zones free of charge and this represents a very significant advantage especially for companies operating in the Logistics distribution field.

However, despite these advantages, the limited driving range of EVs still remains an issue to take into account, since they may require to be recharged also more than once at the Recharging Stations (RSs) along their trips. This, in fact, together with a poor distribution of the RSs on the territory, causes the so-called "range anxiety" that over the years has strongly limited the diffusion of such vehicles (Franke et al., 2016). Indeed, the possibility of EVs running out of energy to either reach the closest RS or the depot, requires properly planning in advance their routes, including possible stops at RSs. The problem of recharging EVs along their trips is particularly recurrent in the context of the mid-haul logistics, taken as a reference in this work, since the driving range of medium-duty EVs is usually not sufficient to cover distances longer

* Corresponding author.
E-mail addresses: maurizio.bruglieri@polimi.it (M. Bruglieri), massimo.paolucci@unige.it (M. Paolucci), o.pisacane@univpm.it (O. Pisacane).

https://doi.org/10.1016/j.cor.2023.106261
Received 17 April 2023; Accepted 21 April 2023
Available online 4 May 2023

than 80 kilometers (Schiffer et al., 2021). Such an issue is attracting the attention of many operations researchers, especially because more and more logistic companies are opting for electric mobility solutions, leading to the *Electric Vehicle Routing Problem with Time Windows* (E-VRPTW) (Schneider et al., 2014). This problem, in fact, aims at routing a given fleet of EVs, based at a common depot, in order to serve a set of geographically distributed customers within their time windows, allowing possible recharges en-route and minimizing the total distance traveled. Therefore, a feasible solution for the E-VRPTW consists of a set of routes, each one performed by an EV that starts from the depot, serves some customers and returns to the depot within a maximum time allowed.

However, over the years, several variants of the E-VRPTW have been addressed especially for properly taking into account the rapid development of battery technology, e.g., the possibility to partially recharge and a higher EVs autonomy. In addition, more realistic models have been also proposed, according to which the energy consumption depends simultaneously on several aspects. In fact, the maximum distance an EV can cover without recharging can significantly vary, for example, according to both its speed (at lower speed the EV range may be double than that at a higher speed) and the carried load (Xiao et al., 2019). To the best of our knowledge, Xiao et al. (2019) represents the first contribution in which the E-VRPTW with a realistic Energy Consumption Rate (EVRPTW-ECR) is introduced through a linear model depending on both the load and the vehicle speed, where the latter can continuously vary between a lower and an upper bound. However, such a new problem is mathematically modeled by the authors without considering possible recharges en-route. This may lead to solutions that can be substantially sub-optimal, as shown by the simple example of Fig. 1 with 6 customers located in two clusters and 3 RSs (including one at the depot), where the numbers in square brackets beside each customer indicate their time windows (TWs), the numbers beside each arc the corresponding length, and the driving range of each vehicle is assumed to be 110 km. Indeed, the optimal solution obtained not allowing recharges en-route, depicted in Fig. 1a, requires three vehicles because the limited driving range does not allow customer C5 to be served in the same route of customers C4 and C6. In fact, in order to satisfy the TWs, the customers should be visited according to the sequence Depot, C6, C4, C5, Depot whose total length is 52+5.8+10+47.43=115.23. On the other hand, in the optimal solution obtained when recharges en-route are possible, shown in Fig. 1b, C5 can be served in a longer route together with C4 and C6 (thanks to the recharge in RS2), thus requiring only two vehicles.

The contribution of this work is threefold. We model for the first time the EVRPTW-ECR through Mixed Integer Linear Programming (MILP) allowing also stops at RSs en-route. Moreover, differently from most of the models in the EVRP literature, we propose a *cloneless* formulation of the problem, i.e., we avoid cloning RSs to allow each station to be possibly visited more than once in the same route or by different vehicles. Moreover, we design a *Random Kernel Search* (RKS) matheuristic approach, based on the proposed cloneless formulation, for efficiently addressing the EVRPTW-ECR also on large-sized instances. Finally, we report the results of an analysis to evaluate the sensitivity of RKS to possible variations of some input parameters which show the robustness of the proposed matheuristic.

The rest of the paper is organized as follows. Section 2 describes the main literature contributions on the E-VRPTW and some variants. Section 3 provides the problem statement and the notation used. Section 4 introduces the cloneless MILP model formulated for the EVRPTW-ECR. Section 5 outlines the RKS matheuristic, whereas a computational comparison between the solutions provided by the MILP cloneless model and those of the RKS matheuristic is discussed in Section 6. Moreover, in Section 6.3, an analysis to evaluate the sensitivity of RKS to possible variations of some input parameters is described. Finally, Section 7 draws some conclusions.

2. Literature review

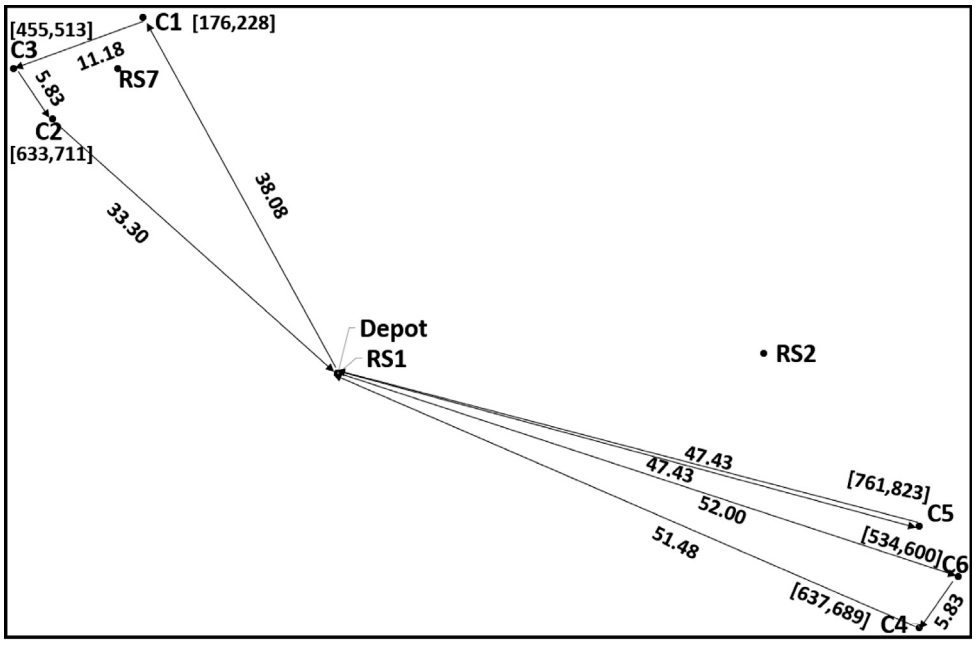
The E-VRPTW belongs to the more general class of Vehicle Routing Problems (VRPs). Readers are refereed to Mor and Speranza (2020) and Macrina et al. (2020) for very recent reviews respectively on VRPs and the E-VRPTW.

The E-VRPTW was introduced by the seminal work by Schneider et al. (2014), where the problem is formulated as a MILP on a complete direct graph, in which the set of vertices includes the depot, the customers and the clones of the RSs possibly visited en-route by the EVs for full recharges. Indeed, in order to allow using each RS more than once in the same route and/or solution, some copies of it are introduced. The ECR is assumed to be linearly proportional to the traveled distance and, differently from the Green Vehicle Routing Problem (Erdoğan and Miller-Hooks, 2012) in which the refueling time is assumed to be constant, in Schneider et al. (2014) the refueling time is considered a function of the state of charge (SOC), leading to a more complex decision problem. For addressing the E-VRPTW also on large-sized instances, the authors design a Variable Neighborhood Search (VNS) hybridized with a Tabu Search (TS) metaheuristic.

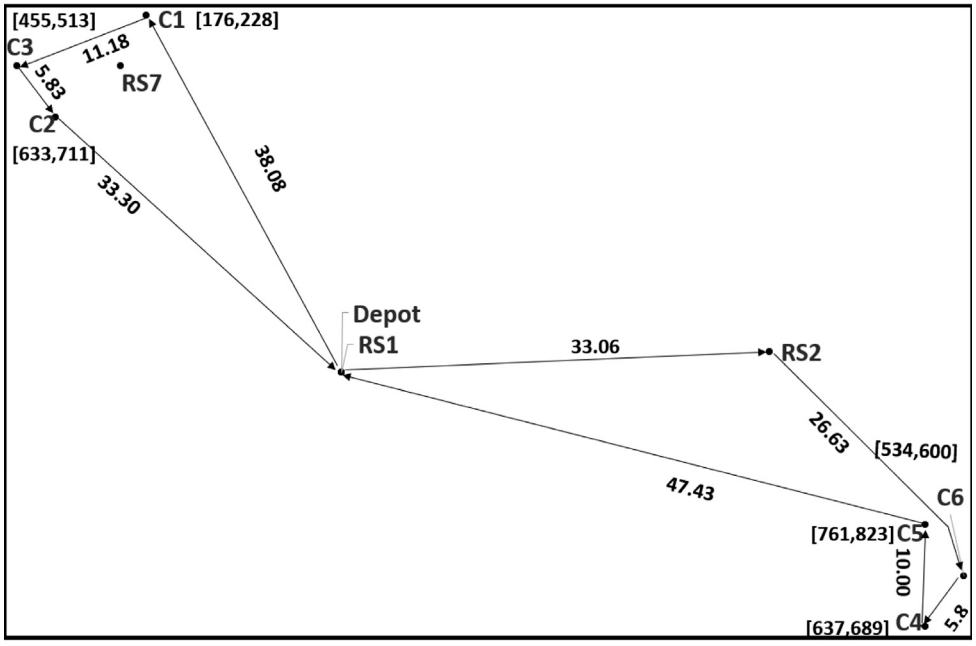
Starting from this work, several other papers have been published in the literature over the years, addressing the E-VRPTW and its variants. For example, Felipe et al. (2014) study an E-VRPTW allowing both partial recharges and multiple recharging technologies. For this new variant of the problem, they develop constructive and local search heuristics used in a non-deterministic Simulated Annealing framework. Under the assumption of partial recharges, in Bruglieri et al. (2015) the authors develop a VNS based matheuristic in which the amount recharged at each station is a variable of the decision process. In particular, the authors solve the E-VRPTW from a time-effective perspective, minimizing the total time spent by each EV outside the depot. Desaulniers et al. (2016) proposed exact branch-price-and-cut algorithms for four variants of the E-VRPTW, allowing either at most one full/partial recharge per route or multiple full/partial recharges per route. In Keskin and Çatay (2016), an E-VRPTW allowing also partial recharges at RSs is solved through an ALNS, which is shown to be effective in finding high quality solutions on the considered benchmark instances. The time-effective E-VRPTW proposed in Bruglieri et al. (2015) is addressed in Bruglieri et al. (2017) through a three-steps matheuristic able to outperform the VNS based matheuristic especially on medium-sized instances. In Goeke (2019), the E-VRPTW is addressed under the assumption that each request consists in moving a commodity from a pickup to a delivery point allowing also partial recharges at RSs. For this new variant, the author designs a granular TS. Recently, in Löffler et al. (2020), an ALNS hybridized with a granular TS is designed to solve the E-VRTW with single recharge, allowing the possibility of both full and partial recharges. The proposed approach is shown to find optimal or near-optimal solutions on the set of benchmark instances introduced by Schneider et al. (2014) with up to 100 customers.

A VRPTW with heterogeneous fleet (i.e., made up by both EVs and ICEVs) is addressed in Hiermann et al. (2016) with the aim of modeling the decisions regarding the fleet size and the route planning, including also possible stops at RSs. For this new problem, the authors propose both a branch-and-price approach and a heuristic method that combines an ALNS with an embedded local search and labeling procedure. In Shao et al. (2017), an E-VRTPW with variable travel time is solved through a dynamic Dijkstra algorithm used to find the shortest path between two nodes in the routes determined by a genetic algorithm. A VRPTW with both EVs and ICEVs is studied in Macrina et al. (2019) allowing also partial recharges and limiting the pollution emissions due to the ICEVs. This new variant of the problem is solved through an Iterated Local Search metaheuristic on the benchmark instances in Schneider et al. (2014) with up 100 customers.

Since 2017, some papers started to deal with also nonlinear charging functions, considered more realistic than the linear one (Montoya et al., 2017). In particular, in Montoya et al. (2017), a hybrid metaheuristic is



a. Optimal solution without recharges en-route as in Xiao et al. (2019).



b. Optimal solution allowing recharges en-route.

Fig. 1. Solution comparison on a simple instance.

developed, combining simple components taken from the literature and some others specifically proposed for the new variant. In Froger et al. (2019), two MILP formulations are proposed and shown to be more effective than the already existing one. In particular, the authors propose an arc-based tracking of the time and the SOC (rather than the traditional node-based tracking). Moreover, in order to avoid having dummy copies of RSs, their first formulation is indeed a path-based model. They also develop an exact labeling algorithm to find the optimal charging decisions on the given routes. In Koç et al. (2019), instead, a variant of the E-VRPTW with nonlinear charging function is introduced considering the possibility of sharing RSs, assuming that some companies may take advantage to jointly invest in such infrastructures. Therefore, the resulting new problem, solved through a multistart heuristic, aims at both deciding the location and technology of the RSs and planning the routes for each company. An efficient linearization method for the concave nonlinear charging function is presented in Zuo et al. (2019), where the problem is then modeled through MILP. In Lee (2021), an algorithm for minimizing both the total travel and charging times without any approximation of the charging time function is developed. In particular, an extended charging stations network is built such that a path on it corresponds to a route on the original one. Then, the resulting problem on this extended network is optimally solved by a branch-and-price method. An E-VRP with load-dependent discharge and non-linear charging is introduced in Kancharla and Ramadurai (2020), allowing multiple visits at the same RS by creating dummy copies of it. In addition, an ALNS is designed which outperforms the already existing solution approach in 63% of the 80 instances used. Finally, in Karakatič (2021), the author addresses a multi-depot E-VRPTW with nonlinear

charging function and cargo capacity constraints, presenting a genetic algorithm for minimizing simultaneously the driving times, the number of stops at RSs en-route and finally, the time of recharging.

Regarding the EVRPTW-ECR, to the best of our knowledge, Goeke and Schneider (2015) is the first literature contribution in which a realistic ECR is considered. In particular, the authors address an E-VRPTW with a mixed fleet, composed by both EVs and conventional internal combustion vehicles. Concerning the EVs, they consider possible stops en-route at the RSs with a linear recharging model and in their ECR model they incorporate some real factors like speed, gradient and cargo load. Moreover, they assume that the vehicle speed is constant and dependent on the arcs since they want to avoid taking into account also the traffic conditions which the vehicle speed can depend on. However, in some papers, e.g., Demir et al. (2014), the average vehicle speed on each arc is modeled as a decision variable without considering also the traffic conditions. Also Lin et al. (2016) consider possible stops en-route at the RSs with a linear recharging model and include real factors in the ECR model, e.g., the payload and the vehicle speed. The vehicle speed is still assumed to be dependent on the arc but constant, and a linear energy cost function simpler than the one already adopted in Barth et al. (2005), Barth and Boriboonsomsin (2009) and Bektaş and Laporte (2011), is used for including the vehicle load effect on the battery consumption. In Zhang et al. (2018), an E-VRPTW, aimed at minimizing the ECR, is studied and modeled as a non-linear program, in which the vehicle speed depending on the arc is constant and also dummy copies of each RS are introduced. In Joo and Lim (2018), an energy efficiency routing problem is solved through an Ant Colony Optimization approach. The authors include some real factors in the proposed ECR, e.g., the vehicle speed and the gradient of the road, resulting in a new energy consumption model that takes into account the discharge and the recuperation phase, but they do not consider possible stops en-route at RSs. In Pelletier et al. (2019), an E-VRPTW with energy consumption uncertainty is addressed, assuming that the ECR is a function of an amount of uncertainty due to both endogenous factors (e.g., the driver behavior) and exogenous factors (e.g., the weather and the road condition). This new variant is then formulated through robust MILP, and in order to solve also large-sized instances, a LNS based heuristic approach is also developed. In Basso et al. (2019), a new energy consumption model is proposed including topography, speed, acceleration and powertrain efficiency. However, the authors still maintain both dummy copies of the RSs to allow multiple visits and a constant vehicle speed. Recently, in Ferro et al. (2020) a model for EVRPTW-ECR is introduced, assuming time-of-use energy prices as well as a complex energy consumption model, which takes into account the contribution of speed, load, terrain gradient and starts and stops. In addition, the model assumes the speed of the vehicles on the arcs not fixed, but it can vary among a given discrete set of values. A MILP formulation using clones of the RS is presented which is able to solve instances up to 15 customers. Finally, in Xiao et al. (2019), an ECR is modeled as function of the vehicle speed and load. The authors formulate the resulting EVRPTW-ECR through MILP, assuming that the vehicle speed is not constant but a continuous decision variable. Two linearization methods (both inner and outer approximation) are adopted for treating the nonlinear relationship between the vehicle speed and the travel time. However, the possibility of having recharges en-route at RSs is not considered.

To the best of our knowledge, the present work represents the first contribution in which the EVRPTW is addressed by considering simultaneously several aspects:

- the need of using a realistic ECR depending on both the vehicle speed and load;
- vehicle speed as a continuous variable in the decision process;
- possible stops at the RSs are permitted en-route.

In addition, in this paper we introduce a mathematical formulation for the EVRPTW-ECR that does not contain dummy copies of the RSs, although more than one stop at the same RS is allowed in the same route and/or solution.

Regarding the proposed RKS, our matheuristic contains three main contributions, compared to the versions already proposed in the literature. First of all, the way to find an initial solution. Indeed, we do not take advantage of the solution obtained by solving the continuous relaxation of the problem since, as known, it is not useful for the VRPs (Archetti et al., 2021), in general. In fact, we design a new matheuristic, called Random k-Degree Search (RkDS), to initialize RKS, showing that it is able to efficiently find initial solutions. Secondly, at each iteration, our RKS adopts randomness for generating the buckets. Finally, RKS also adapts the bucket dimension, in particular adjusting it according to the difficulty of solving the restricted MILP, evaluated in terms of total computational time required by the solver. To the best of our knowledge, this is the first Kernel Search approach exploiting such two kinds of mechanisms for identifying the buckets.

3. Problem statement and notation

The EVRPTW-ECR consists in routing a homogeneous fleet of EVs to serve a set N of customers starting from and returning to a depot. Each customer $i \in N$ has an associated demand Q_i, a service time G_i and must be visited by a single vehicle which starts to serve i within the time window $[E_i, L_i]$. Each EV has a maximum battery capacity B and a maximum load capacity Γ. We assume that the battery charging of EVs occurs at constant current, so that the battery level increases linearly with time during charging (Kisacikoglu et al., 2011). This assumption is justified by the fact that the number of recharging cycles, and so the battery life, is significantly increased when the battery operates at constant current (Schuster et al., 2015). Hence, we assume the consequent reduction of the driving range of EVs is an acceptable trade-off with respect to the environmental and recycling costs of exhausted battery (Lander et al., 2021). The linear charging and discharging behavior of the battery can be imposed by setting an allowed range for the battery SOC. In particular, we assume that the percentage of battery capacity can range in $[\sigma, \gamma]$, where σ and γ represent the minimum and maximum allowed percentage battery capacity, respectively, that may depend on the kind of battery considered. In addition, we suppose that the EVs leave the depot with the battery charged to percentage γ. We assume that the ECR function, $ECR(v, f)$, depends on both the EV speed v and load f which are continuous decision variables. We remark that in practice an average speed on each arc can be imposed to each vehicle, e.g., thanks to a driver-assistance system (ADAS) as in Wu et al. (2015), So et al. (2020) or using an autonomous vehicle. In particular, we consider the ERC function $ECR(v, 0)$ with no load ($f = 0$) as giving the percentage of battery consumed per kilometer, computed as $ECR(v, 0) = P_B/v \cdot B$, where P_B is the battery power needed to travel one kilometer at speed v and B the maximum battery capacity. Note that P_B is computed as in Goeke and Schneider (2015) with the assumption of constant speed and disregarding the gradient of terrain. As already proposed in Xiao et al. (2019), $ECR(v, 0)$, i.e., the ECR as a function of the speed and with zero load, can be in general linearized with a set H of secant lines as $e \geq \hat{k}_h v + \hat{b}_h$, $h \in H$, where e represents the percentage of battery consumed per kilometer, and each secant h to the ECR function is defined by a pair of constants $(\hat{k}_h, \hat{b}_h)$, specifying the slope and intercept. Besides, for each additional unit of carried load, it is assumed that the ECR of the EVs increases by a constant rate, ϕ, so that the overall ECR is given by $ECR(v, f) = ECR(v, 0) + \phi \cdot f$.

A set S of RSs where EVs can recharge along their trips is available. We assume that at most a single battery recharge may be needed when traveling between two consecutive customer nodes. We know the distance D_{ij} between each pair of customers i, j, and both a lower and an upper speed limit, denoted by V_{ij}^{min} and V_{ij}^{max}, respectively.

Similarly to ECR, also the non-linear time-speed relation is linearized as in Xiao et al. (2019), where two possible methods, based on either tangent or secant lines, have been proposed and tested. In particular, we consider a set P of tangent lines, computing $t \geq D(\tilde{k}_p v + \tilde{b}_p)$, $p \in P$, being t the travel time, D the traveled distance, and each line p identified by a slope intercept pair $(\tilde{k}_p, \tilde{b}_p)$, since this kind of linearization has been shown to outperform the one based on secant lines (Xiao et al., 2019).

The objective of EVRPTW-ECR consists in minimizing the total travel cost, including the fixed cost F of using each EV, the cost of the energy consumed en-route depending on the total traveled distance through the coefficient C_E, and the drivers' wage cost considered proportional to the total duration of the routes with the coefficient C_T.

4. A cloneless mathematical model

In this section we introduce a MILP formulation for the EVRPTW-ECR. The proposed formulation is defined as *cloneless* since it avoids to use the clones of the stations to model the possibility to visit the same RS more than once in the same solution and/or route. However, a certain level of duplication exists also in our model since each link (i, j) is replaced with a set of links, each one corresponding to a possible detour to a charging station. Compared to a classical formulation with clones of the stations, our formulation has three advantages. First, the number of routing variables visiting a RS does not depend on how many times a RS can be visited by an EV. Whereas, the classical formulation with clones requires to compute an upper bound U on the number of times a RS could be visited by an EV since each RS must be cloned U times, for each EV. Second, in our formulation the number of routing variables visiting a RS does not depend on the fleet size, whereas in the classical formulation it depends on an upper bound, m, on the fleet size since each RS must be cloned mU times. Third, our formulation allows significantly reducing these variables thanks to the introduction of the concept of non dominated RSs (Section 4.1) that cannot be applied in the classical formulation with clones of the RSs.

The cloneless formulation is defined considering a multigraph $G^M = (N_0, A^M)$ derived from the road network graph $G = (N_0, A)$, where $N_0 = N \cup \{0\}$, combining each arc $(i, j) \in A$ with the subsets S_{ij} of RSs that can be reached with a feasible detour when traveling from i to j. Specifically, we assume that at most one recharge between two customers may be required, since this lends itself well to the context of the mid-haul logistics taken as reference in this work. In particular, $A^M = \{(i, j, s) : (i, j) \in A, s \in S_{ij} \cup \{n\}\}$, where (i, j, s) denotes the path connecting the two customer nodes i and j with the intermediate stop at RS s. The definition of A^M also includes a possible intermediate stop at a fictitious node n, where $n = |N_0| + 1$, to denote with (i, j, n) the direct path from i to j without any recharging stop. Moreover, we indicate by δ_j^- the set of arcs entering in node j (*backward star*), i.e., $\delta_j^- = \{(i, j) : (i, j) \in A\}$ and by δ_i^+ the set of arcs exiting from node i (*forward star*), i.e., $\delta_i^+ = \{(i, j) : (i, j) \in A\}$.

In the following we report the sets of indices, the parameters and the variables used in the cloneless formulation.

Sets

- N set of customer nodes;
- $N_0 = N \cup \{0\}$ set of customer nodes including the depot (denoted by 0);
- $A = \{(i, j) : i, j \in N_0, i \neq j\}$ set of arcs that can be traversed by EVs;
- $A' = \{(i, j) : i, j \in N, i \neq j\} \subset A$, set of arcs connecting customer nodes;
- P set of tangent lines for the times-speed linearization;
- H set of secant lines for the ECR linearization;
- S set of RS including the depot that can be used as RS;
- $S_{ij}, (i, j) \in A$ set of all intermediate RS where a vehicle can stop when traveling from i to j;
- $S'_{ij} = S_{ij} \cup \{n\}, (i, j) \in A$ set of all intermediate RS for arc (i, j), including the fictitious RS n;
- $A^M = \{(i, j, s) : (i, j) \in A, s \in S'_{ij}\}$ set of paths connecting pairs of customer nodes with a possible intermediate stop at a RS;
- $A^S = \{(i, s), (s, j) : (i, j) \in A, s \in S_{ij}\}$ set of arcs connecting customer nodes i and j with the RS s in the possible detours associated with $(i, j, s) \in A^M$, where $s \neq n$.

Parameters

- F fixed cost per EV;
- B maximum battery energy capacity;
- Γ load capacity of EVs;
- C_E unitary energy cost;
- C_T unitary cost for time spent in the routes;
- $D_{ij}, (i, j) \in A \cup A^S$ length of arcs;
- $G_i, i \in N$ service time at node i;
- $Q_i, i \in N$ demand of node i;
- $[E_i, L_i], i \in N$ time window of node i;
- $\tilde{k}_p, \tilde{b}_p, p \in P$ slope and intercept of the pth tangent line of the times-speed linearization;
- $\hat{k}_h, \hat{b}_h, h \in H$ slope and intercept of the hth secant line of the ECR linearization;
- ϕ ECR for unit of carried load;
- $V_{ij}^{max}, V_{ij}^{min}, (i, j) \in A \cup A^S$ upper and lower bound for the average speed on arcs;
- γ and σ allowed maximum and minimum percentage of battery capacity;
- $n = |N_0| + 1$ index denoting a fictitious null RS.
- ρ time needed for a full battery recharge.

Variables

- $x_{ijs}, (i, j, s) \in A^M$ binary routing variables such that $x_{ijs} = 1$ if an EV travels on arc (i, j), with an intermediate stop at RS s (if $s = n$ the vehicle travels directly from i to j); $x_{ijs} = 0$ otherwise;
- $v_{ij} \geq 0, (i, j) \in A$ average speed of the EV traveling on arc (i, j) connecting directly two customers;
- $v_{ijs}^a \geq 0, (i, j) \in A, s \in S_{ij}$ average speed of the EV on arc (i, s) when traveling from node i to node j with an intermediate stop at RS s;
- $v_{ijs}^b \geq 0, (i, j) \in A, s \in S_{ij}$ average speed of the EV on arc (s, j) when traveling from node i to node j with an intermediate stop at RS s;
- $e_{ij} \in [0, 1], (i, j) \in A$ energy consumption, expressed as percentage of battery for unit of distance, of an EV traveling on arc (i, j) connecting directly two customers;
- $e_{ijs}^a \in [0, 1], (i, j) \in A, s \in S_{ij}$ energy consumption for unit of distance on arc (i, s) of an EV traveling from customer node i to customer node j with an intermediate stop at RS s;
- $e_{ijs}^b \in [0, 1], (i, j) \in A, s \in S_{ij}$ energy consumption for unit of distance on arc (s, j) of an EV traveling from customer node i to customer node j with an intermediate stop at RS s;
- $t_{ij} \geq 0, (i, j) \in A$ travel times spent on arcs (i, j) directly connecting customers i and j;
- $t_{ijs}^a \in [0, 1], (i, j) \in A, s \in S_{ij}$ travel times spent on arc (i, s) by an EV traveling from customer node i to customer node j with an intermediate stop at RS s;
- $t_{ijs}^b \in [0, 1], (i, j) \in A, s \in S_{ij}$ travel times spent on arc (s, j) by an EV traveling from customer node i to customer node j with an intermediate stop at RS s;
- $\tau_{ijs} \geq 0, (i, j) \in A, s \in S_{ij}$ time spent by an EV to recharge at RS s when traveling from customer node i to customer node j;
- $f_{ij} \geq 0, (i, j) \in A$ flow variables giving the load of an EV traveling on arc (i, j);
- $w_i \geq 0, i \in N$ waiting time at customer node i;

- $a_j \in [E_j, L_j], j \in N$ arrival time at customer node j;
- $r_j \in [\sigma, \gamma], j \in N$ percentage of battery capacity on arrival at customer node j.

Cloneless formulation

$$\min F \cdot \sum_{j \in N} \sum_{s \in S'_{0j}} x_{0js} + B \cdot C_E \cdot \sum_{(i,j) \in A} \left(D_{ij} e_{ij} + \sum_{s \in S_{ij}} (D_{is} e^a_{ijs} + D_{sj} e^b_{ijs}) \right) + C_T \cdot \left[\sum_{(i,j) \in A} \left(t_{ij} + \sum_{s \in S_{ij}} (t^a_{ijs} + t^b_{ijs} + \tau_{ijs}) \right) + \sum_{j \in N} (G_j + w_j) \right] \quad (1)$$

subject to

$$\sum_{(i,j) \in \delta^-_j} \sum_{s \in S'_{ij}} x_{ijs} = 1 \quad j \in N \quad (2)$$

$$\sum_{(i,j) \in \delta^+_i} \sum_{s \in S'_{ij}} x_{ijs} = 1 \quad i \in N \quad (3)$$

$$t_{ij} \geq D_{ij}(\tilde{k}_p v_{ij} + \tilde{b}_p) - D_{ij}(\tilde{k}_1 V^{min}_{ij} + \tilde{b}_1)(1 - x_{ijn}) \quad (i,j) \in A, p \in P \quad (4)$$

$$t^a_{ijs} \geq D_{is}(\tilde{k}_p v^a_{ijs} + \tilde{b}_p) - D_{is}(\tilde{k}_1 V^{min}_{is} + \tilde{b}_1)(1 - x_{ijs}) \quad (i,j) \in A, s \in S_{ij}, p \in P \quad (5)$$

$$t^b_{ijs} \geq D_{sj}(\tilde{k}_p v^b_{ijs} + \tilde{b}_p) - D_{sj}(\tilde{k}_1 V^{min}_{sj} + \tilde{b}_1)(1 - x_{ijs}) \quad (i,j) \in A, s \in S_{ij}, p \in P \quad (6)$$

$$\tau_{ijs} \geq \rho(\gamma - r_i + D_{is} e^a_{ijs}) - \rho(1 - x_{ijs}) \quad (i,j) \in A, i \neq 0, s \in S_{ij} \quad (7)$$

$$\tau_{0js} \geq \rho D_{0s} e^a_{0js} - \rho(1 - x_{0js}) \quad j \in N, s \in S_{0j} \quad (8)$$

$$a_j \geq E_0 + t_{0j} - (E_0 + \frac{D_{0j}}{V^{min}_{0j}})(1 - x_{0jn}) \quad j \in N \quad (9)$$

$$a_j \geq E_0 + t^a_{0js} + t^b_{0js} + \tau_{0js} - (E_0 + \frac{D_{0s}}{V^{min}_{0s}} + \frac{D_{sj}}{V^{min}_{sj}} + \rho)(1 - x_{0js}) \quad j \in N, s \in S_{0j} \quad (10)$$

$$a_j \geq a_i + t_{ij} + G_i + w_i - L_0(1 - x_{ijn}) \quad (i,j) \in A' \quad (11a)$$

$$a_j \leq a_i + t_{ij} + G_i + w_i + L_0(1 - x_{ijn}) \quad (i,j) \in A' \quad (11b)$$

$$a_j \geq a_i + t^a_{ijs} + t^b_{ijs} + G_i + w_i + \tau_{ijs} - L_0(1 - x_{ijs}) \quad (i,j) \in A', s \in S_{ij} \quad (12a)$$

$$a_j \leq a_i + t^a_{ijs} + t^b_{ijs} + G_i + w_i + \tau_{ijs} + L_0(1 - x_{ijs}) \quad (i,j) \in A', s \in S_{ij} \quad (12b)$$

$$e_{ij} \geq \hat{k}_h v_{ij} + \hat{b}_h + \Phi f_{ij} - (1 - x_{ijn}) \quad (i,j) \in A, h \in H \quad (13)$$

$$e^a_{ijs} \geq \hat{k}_h v^a_{ijs} + \hat{b}_h + \Phi f_{ij} - (1 - x_{ijs}) \quad (i,j) \in A, s \in S_{ij}, h \in H \quad (14)$$

$$e^b_{ijs} \geq \hat{k}_h v^b_{ijs} + \hat{b}_h + \Phi f_{ij} - (1 - x_{ijs}) \quad (i,j) \in A, s \in S_{ij}, h \in H \quad (15)$$

$$r_j + D_{0j} e_{0j} - (1 - x_{0jn}) \leq \gamma \quad j \in N \quad (16)$$

$$D_{0s} e^a_{0js} - (1 - x_{0js}) \leq \gamma \quad j \in N, s \in S_{0j} \quad (17)$$

$$r_j + D_{sj} e^b_{0js} - (1 - x_{0js}) \leq \gamma \quad j \in N, s \in S_{0j} \quad (18)$$

$$r_i \geq r_j + D_{ij} e_{ij} - (1 - x_{ijn}) \quad (i,j) \in A' \quad (19a)$$

$$r_i \leq r_j + D_{ij} e_{ij} + (1 - x_{ijn}) \quad (i,j) \in A' \quad (19b)$$

$$r_i \geq \sigma + D_{is} e^a_{ijs} - (1 - x_{ijs}) \quad (i,j) \in A', s \in S_{ij} \quad (20)$$

$$r_j \geq \gamma - D_{sj} e^b_{ijs} - (1 - x_{ijs}) \quad (i,j) \in A', s \in S_{ij} \quad (21a)$$

$$r_j \leq \gamma - D_{sj} e^b_{ijs} + (1 - x_{ijs}) \quad (i,j) \in A', s \in S_{ij} \quad (21b)$$

$$r_i \geq \sigma + D_{i0} e_{i0} - (1 - x_{i0n}) \quad i \in N \quad (22a)$$

$$r_i \leq \sigma + D_{i0} e_{i0} + (1 - x_{i0n}) \quad i \in N \quad (22b)$$

$$r_i \geq \sigma + D_{is} e^a_{i0s} - (1 - x_{i0s}) \quad i \in N, s \in S_{i0} \quad (23)$$

$$\sigma \leq \gamma - D_{s0} e^b_{i0s} + (1 - x_{i0s}) \quad i \in N, s \in S_{i0} \quad (24)$$

$$V^{min}_{ij} x_{ijn} \leq v_{ij} \leq V^{max}_{ij} x_{ijn} \quad (i,j) \in A \quad (25)$$

$$V^{min}_{is} x_{ijs} \leq v^a_{ijs} \leq V^{max}_{is} x_{ijs} \quad (i,j) \in A, s \in S_{ij} \quad (26)$$

$$V^{min}_{sj} x_{ijs} \leq v^b_{ijs} \leq V^{max}_{sj} x_{ijs} \quad (s,j) \in A, s \in S_{ij} \quad (27)$$

$$\sum_{(j,i) \in A} f_{ji} - \sum_{(i,j) \in A} f_{ij} = Q_i \quad i \in N \quad (28)$$

$$f_{ij} \leq \Gamma \sum_{s \in S'_{ij}} x_{ijs} \quad (i,j) \in A \quad (29)$$

$$\begin{aligned} &a_j \in [E_j, L_j], w_j \geq 0, r_j \in [\sigma, \gamma], j \in N \\ &x_{ijs} \in \{0,1\}, (i,j) \in A, s \in S'_{ij} \\ &v_{ij} \geq 0, e_{ij} \in [0,1], t_{ij} \geq 0, (i,j) \in A \\ &v^a_{ijs} \geq 0, v^b_{ijs} \geq 0, (i,j) \in A, s \in S_{ij} \\ &e^a_{ijs} \geq 0, e^b_{ijs} \geq 0, (i,j) \in A, s \in S_{ij} \\ &t^a_{ijs} \geq 0, t^b_{ijs} \geq 0, (i,j) \in A, s \in S_{ij} \\ &\tau_{ijs} \geq 0, (i,j) \in A, s \in S_{ij} \\ &f_{ij} \geq 0, (i,j) \in A \end{aligned} \quad (30)$$

Objective (1) models the total cost minimization including the total fixed costs of using EVs, the total cost of energy consumed along the trips and the total drivers' wage costs due to the total time spent in the routes, respectively. Constraints (2) and (3) impose that in a vehicle route a customer node is respectively followed and preceded by a different customer node or depot, directly or with an intermediate stop at an RS.

Constraints (4)–(6) determine the travel time of vehicles. In particular, (4) consider the time needed to directly travel between a customer node or depot to a different customer node or depot, whereas (5) and (6) determine the travel time for the two paths (upstream and downstream, respectively) connecting two nodes with an intermediate stop at an RS. All these constraints exploit the piecewise linearization of the time-speed relationship by using a set of tangent lines to inner approximate the nonlinear curve $t_{ij} = D_{ij}/v_{ij}$. Note that we consider the first tangent line, denoted by $p = 1$, since this approximates the longest travel time associated with the lowest speed.

The two constraints (7) and (8) allow the computation of the time spent by a vehicle at an RS to recharge the battery after visiting a customer or after leaving the depot, respectively. Note that we assume

full recharges up to the allowed maximum percentage of battery γ, and that the recharging time depends on the residual percentage of battery of the vehicle on its arrival at the RS.

The constraints (9) and (10) give the arrival time at first customer node after a direct trip from depot and when the EV stops at an intermediate RS after leaving the depot, respectively.

The two pairs of constraints (11a)–(11b) and (12a)–(12b) determine the arrival time at customer nodes when the vehicle directly travels between two customer nodes and when it stops at an intermediate RS, respectively.

Constraints (13) provides the percentage of energy consumption for unit of distance when the vehicle travels directly between two nodes (customers or depot), whereas (14) and (15) give the consumption rates for the two stretches in case of an intermediate stop at an RS.

The next set of constraints are introduced to compute the remaining energy level on the arrival at a customer node j or depot in the following different cases. In constraints (16), j is directly reached from the depot. When j is reached from the depot with an intermediate stop at RS s, (17) ensure that s is reachable from the depot and (18) provide the residual energy level at j. Constraints (19a) and (19b) model the residual energy level for the direct travel between two customer nodes. Constraints from (20) to (21b) are related to travels between two customer nodes with an intermediate stop at RS: (20) ensure that the RS is reachable from the starting node, whereas (21a) and (21b) provide the energy rate on the arrival at the ending node. Finally, (22a) and (22b) consider the direct return to the depot from a customer node, and (23) and (24) the return to the depot with an intermediate stop at a RS.

Constraints from (25) to (27) set the lower and upper bound for the speed of vehicles; in particular, (25) specify the speed bounds for the vehicle directly traveling on an arc $(i,j) \in A$, whereas (26) and (27) consider the speed bounds on the two stretches connecting two nodes with an intermediate stop at an RS. It is worth remarking that the vehicle speed on each arc represents an average value on the roads belonging to the shortest path linking two nodes. In addition, in the proposed MILP model, such paths (i,j,s), for each station s, are further distinguished into upstream and downstream stretches (i,s) and (s,j), respectively.

Constraints (28) are the flow conservation that guarantee the satisfaction of the customer demand. Constraints (29) connect flow variables with routing variables, imposing that the vehicle capacity is not exceeded. Finally, constraints (30) specify the nature of the variables.

It is worth noting that the pairwise constraints (11a)–(11b), (12a)–(12b), (19a)–(19b), (21a)–(21b) and (22a)–(22b) impose satisfying the related conditions for equality.

4.1. Computation of non-dominated RS sets S_{ij}

The proposed MILP model is based on the computation of the set $S_{ij}\ \forall (i,j) \in A$ of all the feasible RSs that may be advantageous to visit, going from i to j. In Algorithm 1, the steps for computing this set are described, where we denote by e_{ij}^{min} the minimum energy consumption for unit of distance to go from i to j, given by $e_{ij}^{min} = \tilde{k}_1 V_{ij}^{min} + \tilde{b}_1 + \Phi Q_j$ (we consider the first secant line since this approximates ECR from the lowest speed).

More specifically, the stations that cannot be reached from node i or from which it is not possible to reach node j even with the allowed full recharge are removed from S_{ij}, at step 5. Then, for each pair of stations s' and s'' in S_{ij} we check if s' dominates s'' since the former requires a lower energy consumption than that of the latter, both upstream and downstream of the station (step 9). If this is the case the station s'' is removed from S_{ij} (step 10).

If the minimum speed is the same on all the arcs, step 9 can be simplified since in this case e_{ij}^{min} does not depend on arc (i,j) and therefore the energy consumptions upstream and downstream of the stations can be compared just comparing the distances of the stations from nodes i and j, respectively. Thus, the conditional test of step 9 just becomes

$$D_{is''} \geq D_{is'} \wedge D_{s''j} \geq D_{s'j}$$

Algorithm 1 Computation of sets S_{ij} with non uniform minimum speeds

1: **for** $(i,j) \in A$ **do**
2: $S_{ij} := S$;
3: **for** $s \in S$ **do**
4: **if** $D_{is} e_{is}^{min} > \gamma - \sigma \vee D_{sj} e_{sj}^{min} > \gamma - \sigma$ **then**
5: $S_{ij} := S_{ij} \setminus \{s\}$;
6: **end if**
7: **end for**
8: **for** $s', s'' \in S_{ij} : s' \neq s''$ **do**
9: **if** $((\tilde{k}_1 V_{is''}^{min} + \tilde{b}_1 + \Phi Q_j) D_{is''} \geq (\tilde{k}_1 V_{is'}^{min} + \tilde{b}_1 + \Phi Q_j) D_{is'} \wedge (\tilde{k}_1 V_{s''j}^{min} + \tilde{b}_1 + \Phi Q_j) D_{s''j} \geq (\tilde{k}_1 V_{s'j}^{min} + \tilde{b}_1 + \Phi Q_j) D_{s'j}$ **then**
10: $S_{ij} := S_{ij} \setminus \{s''\}$;
11: **end if**
12: **end for**
13: **end for**

4.2. Pre-processing phase

Some preliminary considerations can be carried out for a pre-processing phase with the aim of a-priori removing useless arcs from the graph G and by consequence from the multigraph G^M. In particular, the arcs (i,j) that violate one of the following conditions can be removed:

$$Q_i + Q_j > \Gamma \quad \forall i,j \in N, i \neq j \tag{31}$$

$$E_i + G_i + \frac{D_{ij}}{V_{ij}^{max}} > L_j \quad \forall i,j \in N, i \neq j \tag{32}$$

$$\frac{D_{0i}}{V_{0i}^{max}} + \frac{D_{ij}}{V_{ij}^{max}} + \frac{D_{j0}}{V_{j0}^{max}} > L_0 \quad \forall i,j \in N, i \neq j \tag{33}$$

Indeed, conditions (31) assures that (i,j) is removed if and only if the sum of the demands of the two customers exceeds the vehicle cargo capacity. Conditions (32) allow to remove (i,j) if customer j cannot be reached from customer i before L_j, since the time window of customer j would be violated. Conditions (33) assures that (i,j) is removed if it is not possible to return to the depot within L_0.

5. The Random Kernel Search matheuristic

In this section we introduce a new matheuristic, called *Random Kernel Search* (RKS), developed to face the considered EVRPTW-ECR. This matheuristic is inspired by the Kernel Search Algorithm (KSA), a general purpose MILP heuristic originally proposed in Angelelli et al. (2010).

The rationale of KSA is based on the observation that, in general, only a subset of integer and/or binary variables takes a positive value in the optimal solution to MILP models. Therefore, the key idea of KSA is trying to reduce the size of the model by restricting it to a subset of integer/binary variables that are more likely to be not zero in the optimal solution. This restricted set of "promising" variables, denoted by K, is called Kernel Set (KS). Then, fixing to zero all the binary/integer variables but the ones in the KS generates a *restricted MILP* model, indicated as $MILP(K)$, of smaller size and hence easier to solve. KSA aims at identifying the KS by iteratively solving a series of restricted MILP models.

In the following, let us refer to mixed binary programming models as the one proposed for the EVRPTW-ECR. Let $\mathcal{B}$ represent the set of binary variables generically denoted as $x_j, j \in \mathcal{B}$. In the basic KSA,

an initial KS is determined from the solution of the linear relaxation of the model as $K = \{j : \hat{x}_j > 0, j \in \mathcal{B}\}$, where $\hat{x}_j$ denotes the value assumed by the binary variable x_j in the optimal solution of the linear relaxation. Then, an initial solution is found solving the restricted $MILP(K)$, so determining the initial upper bound z^H (fixing $z^H = \infty$ in the case of unfeasibility). The binary variables not in K are first sorted in non-decreasing order of their reduced cost and then partitioned into subsets B_i, $i = 1, \dots, NB$, called *buckets*, of a fixed cardinality λ_b, usually chosen $\lambda_b = |K|$. KSA then iterates the solution of restricted models $MILP(K \cup B_i)$, $i = 1, \dots, NB$, imposing as additional constraints the upper bound z^H as cutoff value for the objective, and the condition that at least one binary variable in B_i takes value one in the optimal solution. If $MILP(K \cup B_i)$ has a feasible solution, then z^H is updated and K is adjusted adding the binary variables in B_i with value one in the optimal solution.

KSA has been successfully exploited to face different combinatorial problems (e.g., Filippi et al., 2016, Labbé et al., 2019, Santos-Peñate et al., 2020 and Filippi et al., 2021), and also several variants of it have been proposed in the literature. Among the most recent ones, in Guastaroba et al. (2017) an adaptive KSA is introduced where the instances are classified as easy, normal or hard, then adjusting the size of the KS accordingly and in case of hard instances also performing some variable fixing. An iterative version of KSA is adopted in Gobbi et al. (2019) for a multi-service nurse routing problem, where two complete iterations on the set of buckets are performed, and in Hanafi et al. (2020) for facing a multi-visit team orienteering problem with precedence constraints. In particular, in Hanafi et al. (2020), the algorithm can perform multiple iterations over the set of buckets until a time limit is reached. The algorithm exploits different strategies for sorting the variables to be included in the buckets, taking into account some measures related to the considered problem. In addition, the KSA in Hanafi et al. (2020) also adopts random weights affecting the sorting and it allows the buckets to partially overlap.

One of the main issue with KSA is that in many problems the information coming from the solution of the linear relaxation is not sufficient to properly identify both the initial KS and to sort the other binary variables into the buckets. However, a few papers in the literature propose alternative initialization strategies. A KSA for multi-plant capacitated lot-sizing problem with setup carry-over is presented in Carvalho and Nascimento (2018), where the initial KS and the set of bucket are generated selecting a subset of binary variables from a pool of elite solutions produced by a Lagrangian heuristic. Recently, a multivehicle inventory routing problem has been considered in Archetti et al. (2021), where a clear analysis shows how for this problem the solution of the linear relaxation does not provide useful information to identify the promising variables and, hence, to generate the initial kernel and the set of buckets. We can observe that such an analysis can be extended to the class of vehicle routing problems in a quite evident way. Therefore, in Archetti et al. (2021), a tabu search is used to generate solutions from which to learn which binary variables are more suitable to be included in the initial kernel and in the buckets.

In this paper, we design RKS, which is described in Algorithm 2, as a new version of KSA, trying to overcome the initialization issue described above for the EVRPTW-ECR. In addition, RKS incorporates concepts as dynamic adaptation of the size of the KS and buckets, as well as randomization and overlap for the bucket generation.

The input of RKS consists of the original cloneless model (provided in Section 4), M, an initial maximum time limit for the solution of the restricted MILP models, T^{step}, the maximum time allowed for finding an initial solution, T^{ini}, the total maximum allowed time, T^{max}, the maximum number of iterations for each fixed KS, I_K^{max}, the minimum number of iterations during which a binary variable is kept in the KS, I_{KS}^{min}, and the maximum number of iterations without improvement of the current best solution, I_N^{max}. At the beginning of Algorithm 2 (step 2), an initialization matheuristic, called Random k-Degree Search (RkDS) and described in sub- Section 5.1, is used to construct a first feasible solution x^0 within the time limit T^{ini}. If no solution is found, the algorithm terminates with failure (steps 3–5), otherwise, the initial upper bound Z^H is set equal to the objective value $Z(x^0)$ (step 6). Then, the initial KS K is initialized as the subset of $\mathcal{B}$ such that $x_j^0 = 1$ (step 7).

A quantity p_j is associated with each variable $x_j \in \mathcal{B}$ whose value is used to smoothly update K. In particular, p_j is initialized equal to zero for the variables x_j in the initial K and to ∞ for all the others in $\mathcal{B} \setminus K$ (step 8). The value p_j is relevant only for the variables in K, since it allows keeping a variable in K at least for a minimum number of iterations I_{KS}^{min}. Then, a parameter β, called *bucket size factor*, is initialized equal to 2 (step 9). Such a factor is used to determine at each iteration the size of the bucket as $\beta \cdot |K|$, and it is dynamically adjusted according to how much hard it is solving the restricted MILP. The maximum time T^{sol} allowed to the MILP solver for solving M at each iteration is initialized equal to T^{step} (step 10), and the counters of the total not improving iterations and local not improving iterations, I_N and I_{LN}, respectively, are initialized to zero (step 11).

The main algorithm loop (steps 12–38) iterates until the total time does not exceed T^{max} or until the total number of not improving iterations does not exceed I_N^{max}. Then, keeping fixed the current KS K, an internal loop (steps 13–29) is performed for I_K^{max} iterations. In each iteration, a new bucket B is generated as a subset of $\beta \cdot |K|$ variables randomly extracted from $\mathcal{B} \setminus K$ (step 14). Then, the reduced MILP $M(K \cup B)$, that is M where all the variables in $\mathcal{B}$ but the ones in $K \cup B$ are fixed to zero, is solved within T^{sol} (step 15), having added the *bucket constraint*, that imposes that at least one variable in B must be equal to 1, and the *objective cutoff* with the upper bound Z^H. Next, the total computation time T is updated adding the time $\tilde{\tau}$ spent to solve the restricted MILP (step 16).

If a feasible or optimal solution x^c is found and the obtained objective value $Z(x^c)$ improves the current best solution Z^H (step 17), then the current best solution x^H is updated and the two not improving iterations counters I_N and I_{LN} are reset to zero (step 18), otherwise both the counters are increased by 1 (step 20). Next, the adjustment of the size bucket factor β is performed (steps 22–28). In particular, if the MILP solver has terminated with a feasible or not feasible solution not exceeding T^{sol}, then β is increased by 1/4, otherwise β is reduced by 1/4, provided that β never becomes less than 1.

After the completion of the inner loop, a KS update phase is performed. Firstly, the value of p_j is increased by 1 for all the variables in the current KS (step 30). Then, the binary variables in K with $p_j \geq I_{KS}^{min}$ are removed from K (step 31). Note that these are the variables in K for at least $I_{KS}^{min}+1$ consecutive KS update phases without assuming value 1 in any of the solutions used to update K. Afterwards, p_j is set equal to zero for all the variables in $\mathcal{B}$ such that $x_j^H = 1$ (step 32). Finally, K is updated including the subset of binary variables in $\mathcal{B}$ such that $x_j^H = 1$ (step 33).

The KS update phase is followed by a possible adjustment of T^{sol}, which is increased in order to allow a longer maximum computation time for solving the restricted MILP models whenever the local counter of the number of iterations without improvement exceeds I_K^{max} (steps 33–37), trying in this way to diversify the solution search. In particular, if $I_{LN} > I_{max}^K$, then T^{sol} is doubled, otherwise it is reset equal to T^{step}.

The final iteration performed by the routine *ReOptimizeStations*(x^H) (step 40) aims at improving the best solution found after the end of the main loop. In fact, we observed that such a solution may include some stops at stations that are not actually needed, thus leading to an increase in the total cost. Therefore, *ReOptimizeStations* consists in running the cloneless model by maintaining fixed, for each vehicle, only the sequence of customers as in solution x^H. In particular, denoting with $x_{ijs}^H, (i,j) \in A, s \in S'_{ij}$, the values of the routing variables determined at the end of the (12)-(39) main loop, we add to the cloneless model the following constraints

$$\sum_{s \in S'_{ij}} x_{ijs} = 1 \quad \forall (i,j) \in A \,|\, \exists s \in S'_{ij}, x_{ijs}^H = 1 \tag{34}$$

Algorithm 2 Random Kernel Search (RKS)

Input A MILP model M, T^{step}, T^{ini}, T^{max}, I_K^{max}, I_{KS}^{min}, I_N^{max}
Output A feasible solution to the MILP model or *failure*.

1: $x^0 = \textbf{\textit{RkDS}}(M, T^{ini})$
2: **if** x^0 **not** $Feasible$ **then**
3: **return** *failure*
4: **end if**
5: Set the current upper bound $Z^H = Z(x^0)$;
6: Set the initial KS as $K = \{j : j \in \mathcal{B}, x_j^0 > 0\}$
7: $p_j = \infty \ \forall j \in \mathcal{B} \setminus K$ and $p_j = 0 \ \forall j \in K$
8: $\beta = 2$
9: $T^{sol} = T^{step}$
10: $T = T^{ini}$
11: $I_N = I_{LN} = 0$
12: **while** $T \leq T^{max} \wedge I_N < I_N^{max}$ **do**
13: **for** $I_K = 1$ to I_K^{max} **do**
14: $B = \textbf{\textit{RandomSelect}}(\mathcal{B} \setminus K, \beta \cdot |K|)$
15: $x^c = \textbf{\textit{Solve}}(M(K \cup B), T^{sol})$ adding constraints $\sum_{j \in B} x_j \geq 1$, $Z(x) \leq Z^H$
16: **Update** $T = T + \tilde{\tau}$, where $\tilde{\tau}$ = solution time
17: **if** (x^c is *Feasible* $\vee$ *Optimal*) $\wedge$ $Z(x^c) < Z^H$ **then**
18: $Z^H = Z(x^c)$; $x^H = x^c$; $I_N = I_{LN} = 0$
19: **else**
20: $I_N = I_N + 1$; $I_{LN} = I_{LN} + 1$
21: **end if**
22: **if** $\tilde{\tau} < T^{sol}$ **then**
23: $\beta = \beta + 0.25$
24: **else if** $\beta > 1$ **then**
25: $\beta = \beta - 0.25$
26: **else**
27: $\beta = 1$
28: **end if**
29: **end for**
30: $p_j = p_j + 1 \ \forall j \in K$
31: $K = K \setminus \{j : j \in K, p_j \geq I_{KS}^{min}\}$
32: Set $p_j = 0 \ \forall j \in \mathcal{B} : x_j^H = 1$
33: $K = K \cup \{j : j \in \mathcal{B}, x_j^H = 1\}$
34: **if** $I_{LN} > I_K^{max}$ **then**
35: $T^{sol} = 2 \cdot T^{sol}$, $I_{LN} = 0$
36: **else**
37: $T^{sol} = T^{step}$
38: **end if**
39: **end while**
40: $x^H = \textbf{\textit{ReOptimizeStation}}(x^H)$
41: **return** x^H

In practice, constraints (34) allow to solve the Fixed-Route Electric Vehicle Charging Problem (Montoya et al., 2017) associated with the solution x^H. Then, the solution found by this final re-optimization is returned at termination (step 41).

5.1. The Random k-Degree search initialization matheuristic

The RkDS matheuristic, designed to generate an initial feasible solution for the RKS, operates similarly to the Greedy Random Adaptive Search Procedure (GRASP) (Feo and Resende, 1995). In particular, it iterates the solution of restricted MILP problems in which only the routing variables associated with a subset of arcs selected from the *stars* of the nodes are not fixed to zero. RkDS first builds for each customer node $i \in N$ the *restricted forward* and *backward stars*, respectively defined as $\hat{\delta}_{ik}^+ \subseteq \delta_i^+$ and $\hat{\delta}_{ik}^- \subseteq \delta_i^-$, of cardinality up to k according to a greedy criterion. Afterwards, the algorithm iterates, randomly selecting k' arcs, half from $\hat{\delta}_{ik}^+$ and half from $\hat{\delta}_{ik}^-$. Then, it defines and solves a restricted MILP cloneless model based on the routing variables associated with the selected arcs, including also other routing variables as explained in the following. The quantities k and k' are both fixed input parameters; in particular, k' is even and $k'/2$ is never greater than k.

RkDS, outlined in Algorithm 3, receives in input the maximum allowed computation time, T^{max}, the maximum time limit for the solution of the restricted MILP models, T^{ini}, and the two parameters k and k'. At the beginning (step 1), the algorithm initializes to zero the total elapsed time T^e and the objective value of the best so far solution Z^H to infinity. It is worth noting that, in Algorithm 3, we assume that T^e is implicitly updated according to the time spent by the various steps. Finally, the set X^B of the routing variables which take value 1 in the best so far solution is initialized as empty set, and all the arcs belonging to the forward and backward star of the depot are included in the initial set of the selected arcs A_0^R.

In step 3, for each customer node $i \in N$, $\hat{\delta}_{ik}^+$ and $\hat{\delta}_{ik}^-$ are generated by the routine *GenerateRestrictedStars*, which first sorts the stars δ_i^+ and δ_i^- according to a greedy criterion and then selects the first k arcs from them. In particular, since the distance is used in this work as greedy criterion, δ_i^+ and δ_i^- are sorted in non-decreasing order respectively of D_{ij}, $(i,j) \in \delta_i^+$ and D_{ji}, $(j,i) \in \delta_i^-$, and the resulting restricted stars $\hat{\delta}_{ik}^+$ and $\hat{\delta}_{ik}^-$ include the first shortest k outgoing and incoming arcs. The routine *ComputeArcSelectionProb* (step 4) associates with each arc in the restricted stars a selection probability which is proportional to the length of the arc. In particular, considering the forward stars, this routine determines for each node $i \in N$ the longest arc in δ_i^+ as

$$(i, j^{max}) = \underset{(i,j) \in \delta_i^+}{argmax} D_{ij} \quad \forall i \in N \tag{35}$$

then obtains the selection probabilities of the arcs $(i,j) \in \delta_i^+$ as

$$P_{ij} = P_{ij^{max}} \cdot \frac{D_{ij^{max}}}{D_{ij}} \quad \forall i \in N, (i,j) \in \delta_i^+ \setminus \{(i, j^{max})\} \tag{36}$$

where

$$P_{ij^{max}} = \left(\sum_{j \neq j^{max} : (i,j) \in \delta_i^+} \frac{D_{ij^{max}}}{D_{ij}} + 1 \right)^{-1} \quad \forall i \in N \tag{37}$$

Similarly, the selection probabilities of the arcs in the backward stars are then computed for each customer node. It is worth noting that, if the cardinality of a restricted star is not greater that $k'/2$, then all its arcs are included in A_0^R. This also empties the restricted star that will not be considered in the next random selections (steps 5–10).

The main algorithm loop (steps 12–22) is iterated until a feasible solution is found or the maximum allowed computation time is exceeded (step 12). First, the set of random greedy selected arcs A^R is initialized equal to A_0^R (step 13). Afterwards, the random selection of $k'/2$ arcs from the restricted stars into the set A^R takes place for each customer node by the routine *RandomSelect* in steps 14–17. Then, the restricted model $M(X(A^R) \cup X^B)$ is solved within the T^{ini} time limit (step 18), where the set $X(A^R)$ includes all the routing variables associated with the arcs in A^R, that is, $X(A^R) = \{x_{ijs} : (i,j) \in A^R, s \in S'_{ij}\}$. If a feasible solution is found and the returned feasible solution x^c improves the best so far solution x^H, then x^H is updated and the set X^B is redefined as the set of the routing variables x_{ijs} equal to 1 in x^H (step 20). Hence, at each iteration, the restricted model M corresponds to the MILP cloneless model where only the following subsets of routing variables are not fixed to zero: the routing variables associated with the randomly selected arcs from the greedy restricted stars of the customer nodes, the routing variables in X^B, which represent the memory of the best so far solution and, finally, all the routing variables associated with the arcs of the stars δ_0^+ and δ_0^- of the depot.

At the end of the (12–22) loop, the algorithm returns x^H if a feasible solution is found, otherwise it returns failure.

6. Experimental results

The RKS matheuristic described in Section 5 has been implemented in Java (in the Eclipse environment) and the cloneless MILP model introduced in Section 4 has been solved by ILOG's CPLEX Concert

Algorithm 3 Random k-Degree Search (RkDS)

Input T^{max}, T^{ini}, k, k'
Output A feasible solution x^H with objective function z^H to the MILP model or *failure*.

$T^e = 0$, $Z^H = \infty$, $X^B = \emptyset$, $A_0^R = \delta_0^+ \cup \delta_0^-$
for $i \in N$ **do**
  $[\hat{\delta}_{ik}^-, \hat{\delta}_{ik}^+] =$ ***GenerateRestrictedStars***$(\delta_i^-, \delta_i^+, k)$
  $[\hat{P}_i^-, \hat{P}_i^+] =$ ***ComputeArcSelectionProb***$(\hat{\delta}_{ik}^-, \hat{\delta}_{ik}^+)$
  if $|\hat{\delta}_{ik}^-| \leq k'/2$ **then**
    $A_0^R = A_0^R \cup \hat{\delta}_{ik}^-$, $\hat{\delta}_{ik}^- = \emptyset$
  end if
  if $|\hat{\delta}_{ik}^+| \leq k'/2$ **then**
    $A_0^R = A_0^R \cup \hat{\delta}_{ik}^+$, $\hat{\delta}_{ik}^+ = \emptyset$
  end if
end for
while $Z^H = \infty$ **and** $T^e \leq T^{max}$ **do**
  $A^R = A_0^R$
  for $i \in N$ **do**
    $A^R = A^R \cup$ ***RandomSelect***$(k'/2, \hat{\delta}_{ik}^-, \hat{P}_i^-)$
    $A^R = A^R \cup$ ***RandomSelect***$(k'/2, \hat{\delta}_{ik}^+, \hat{P}_i^+)$
  end for
  $x^c =$ ***Solve***$(M(X(A^R) \cup X^B), T^{ini})$
  if x^c is *Feasible* **and** $Z(x^c) < Z^H$ **then**
    $Z^H = Z(x^c)$, $x^H = x^c$, $X^B = \{x_{ijs} : x^c_{ijs} = 1, (i,j) \in A, s \in S'_{ij}\}$
  end if
end while
if $Z^H < \infty$ **then**
  return x^H
else
  return *Failure*
end if

Technology (version 20.1). The experiments have been carried out on a computer with a 64-bit operating system, 2.39 GHz processor and 32 GB of RAM using up to 4 threads.

Both the approaches have been tested on two sets of instances with medium and large size corresponding to 25 and 100 customers, respectively. Both the sets of instances have 21 charging stations, including also the depot. The large size instances are directly taken from the EVRPTW benchmark set introduced in the seminal work of Schneider et al. (2014), that in turn is based on the VRPTW benchmark set defined by Solomon (1987). The medium size instance set has been derived from the large size set randomly selecting 25 out of 100 customers.

The instances are divided into three classes, depending on the geographical distribution of the customer locations: random customer distribution (R), clustered customer distribution (C), and a mix of both (RC). Groups R1, C1, and RC1 have a short scheduling horizon, i.e., more vehicles are required to serve all customers than in R2, C2, and RC2, which have a long scheduling horizon. The instances within a group differ in terms of time window density and time window width. The number of the large-sized instances considered is 30, 5 for each group. The medium-sized instances are 12 and are obtained from the first 12 instances of kind R1.

The parameters introduced in Section 4 are fixed as shown in Table 1, considering an Electric Iveco Daily vehicle for the vehicle parameters, whereas the other parameters have been fixed as in Xiao et al. (2019).[1]

The parameters for the RKS matheuristic are set as shown in Table 2.

We remark that our RKS is always initialized through RkDS and that the value of T^{ini} has been chosen after some preliminary experiments since we observed that this time was large enough to find an initial solution of acceptable quality for the considered benchmark. In a similar way, the value fixed for T^{step} allows a good tradeoff between

[1] RMB indicates the Chinese currency renminbi.

Table 1
Vehicle parameters setting.

Parameter	Value
F	300 RMB
B	145 kWh
Γ	4900 kg
$C_E = 4$	4 RMB/KWh
$C_T = 0.3$	0.3 RMB/minute
$\tilde{k}_p, \tilde{b}_p \ \forall p \in P$	as in Table 3 of Xiao et al. (2019) with 0.1% accuracy
$\hat{k}_1$ $(H = \{1\})$	$8.80472 \cdot 10^{-6}$
$\hat{b}_1$	$3.2908 \cdot 10^{-3}$
ϕ	$1.60517 \cdot 10^{-6}$
V_{ij}^{min}	20 Km/h, $\forall (i,j) \in A \cup A^S$
V_{ij}^{max}	100 Km/h, $\forall (i,j) \in A \cup A^S$
σ	0.2
γ	0.8
ρ	120 min

Table 2
RKS parameters setting.

Parameter	Value
T^{step}	30 s
T^{ini}	120 s
T^{max}	1800/3600 s
I_K^{max}	4
I_{KS}^{min}	4
I_N^{max}	24
k	$\min\{24, \lvert N \rvert - 1\}$
k'	8

the ability of RKS to explore both the solution space associated with the restricted MILP, and a large number of different buckets. Moreover, the analysis reported in Section 6.3 highlights that RKS is not significantly affected by the values of these parameters, since the size of the random buckets is dynamically adjusted so that the restricted MILPs can always be solved within the allowed time limit. In similar way, also the parameters I_K^{max}, I_{KS}^{min}, I_N^{max} have been fixed through preliminary tests showing that these values ensure a good trade-off between computation time and solution quality.

We adopt a statistical method for determining how many times our RKS matheuristic has to be run on each instance to estimate the average percentage deviations with a margin of error of 0.1% and a confidence of 90%. We obtain that the number of samples needed should be at least 15 runs. Therefore, on each instance, we have run the RKS matheuristic 15 times and in the tables presented in the next sections we always report its average performance over the 15 runs. In particular, we indicate by $Dev\%$ the quantity $100 \cdot (Obj^{RKS} - Obj^{MILP})/Obj^{MILP}$, where Obj^{RKS} is the average objective function value obtained by RKS over the 15 runs and Obj^{MILP} is the objective function value obtained by CPLEX solving the cloneless formulation. Moreover, we indicate by η the number of vehicles used, by $TimeBest$ the CPU time required to obtain the best solution returned. In particular, in the case of the cloneless MILP model, it is the time at which CPLEX determines the best solution. The quantity $\bar{\eta}$ denotes the average number of the EVs used over the 15 runs of RKS. Concerning the cloneless MILP model, we indicate by Gap the final MILP gap returned by CPLEX. Finally, with regard to RKS, we also consider the results after 1800 s, which correspond to the intermediate best solutions obtained by the solver until the time of completion of the first iteration that terminates after 1800 s.

Before detailing the results obtained on the instances used, we want to draw attention to the initialization phase aimed at determining a good quality initial solution for RKS. In fact, alternatively to the proposed RkDS, a simpler method consists in properly adapting the Clark and Wright (CW) procedure (Toth and Vigo, 2002), hence starting from a solution in which each vehicle is assigned to one and only one customer. Table 3 shows a comparison, in terms of quality of both initial and final solutions, obtained by applying the two different

Table 3
Comparison between the RkDS and CW initialization methods.

Instance	CW			RkDS			RkDS vs CW	
	Initial obj	Final obj	Impr	Initial obj	Final obj	Impr	Dev% Initial	Dev% Final
c101–21	44564.283	12964.064	−70.9%	25810.599	9151.614	−61.5%	−42.1%	−29.4%
c201–21	44997.113	12409.425	−72.4%	30256.868	8851.790	−70.3%	−32.1%	−31.7%
r101–21	42581.973	10000.356	−76.5%	26599.623	9471.780	−64.4%	−40.3%	−26.9%
r201–21	42581.973	9865.911	−76.8%	37699.584	9724.960	−73.4%	−15.4%	−25.0%
rc101–21	46697.195	12593.333	−73.0%	23569.552	11773.499	−50.0%	−47.1%	−9.2%
rc201–21	46697.195	12543.189	−73.1%	44109.494	11921.191	−72.9%	−1.0%	−8.0%
Averages			−73.8%			−65.4%	−29.7%	−21.7%

Table 4
Result comparisons on medium-sized instances.

Instance	Cloneless model				RKS					
	Time limit = 7200				Time limit = 1800			Time limit = 3600		
	Obj	η	TimeBest	Gap(%)	Dev%	$\bar{\eta}$	TimeBest	Dev%	$\bar{\eta}$	TimeBest
r101-25	2972.97	5	47.80	0.00	0.00	5.0	412.0	0.00	5.0	412.0
r102-25	2890.52	5	6883.48	70.03	−7.56	4.3	1331.2	−9.88	4.1	1976.4
r103-25	2877.84	5	7163.35	81.63	−9.51	4.0	1419.2	−11.54	4.0	2485.2
r104-25	2990.92	5	7180.49	82.67	−6.65	4.1	1379.4	−12.01	3.9	3075.9
r105-25	2424.88	4	7161.36	10.61	0.24	4.0	1476.5	−0.40	4.0	2168.7
r106-25	3417.14	6	7177.09	83.90	−28.97	3.7	1550.3	−31.38	3.6	2595.3
r107-25	2913.99	5	7144.38	85.76	−11.42	3.8	1246.8	−13.96	3.7	2367.0
r108-25	2818.50	4	6820.03	86.04	−7.28	3.8	1146.6	−13.23	3.5	2320.7
r109-25	2463.61	4	7186.19	65.57	4.45	4.1	1644.2	−0.22	4.0	2822.5
r110-25	2606.53	4	7095.04	82.29	−2.16	4.0	1402.3	−4.54	3.9	2447.9
r111-25	2989.96	5	7124.32	81.49	−11.91	4.0	1208.2	−13.75	4.0	2123.8
r112-25	3070.25	5	7099.59	87.52	−6.11	4.3	1227.0	−9.80	4.1	2534.6

initialization strategies. Such a comparison is presented on a selection of 2 instances for each group C, R and RC of large-size instances. The second, third and fourth columns in Table 3 indicate respectively the value of the objective function of the initial solution found by CW, that of the final solution found by RKS starting from the solution of CW and then, the percentage improvement passing from the initial to the final solution, computed as $100 \cdot (Initial\ obj - Final\ obj)/Initial\ obj$. The fifth, sixth and seventh columns have the same meaning but with reference to RkDS. It is worth noting that RKS is able to significantly improve the initial solutions generated by the two methods. On the average the improvement obtained for the CW initial solutions is greater (73.8%) than the one for the RkDS initial solutions (65.4%). However, this is due to the lower quality of the CW solutions, as shown by the $Dev\%\ Initial$ column, that highlights that the objective of the RkDS initial solutions is on the average 29.7% smaller than the one of the CW initial solutions. The $Dev\%\ Final$ column then reports the deviations of the objectives of the solutions finally generated by RKS when initialized with RkDS with respect to the ones when initialized with CW. We can observe that the RkDS initialization allows to improve the RKS result of 21.7% on average, so justifying the use of this initialization method. We also applied the Friedman non-parametric test that confirmed the statistical significance of this result with a p-value equal to $2.7265 \cdot 10^{-6}$.

6.1. Results on medium-sized instances

In this section, we discuss the comparison of the results between the cloneless MILP model and RKS on the instances with 25 customers, this last initialized through RkDS. On these instances, the cloneless MILP model found the optimal solution only for instance r101-25 out of 12, within the time limit of 7200 s. Moreover, the average TimeBest, respectively with and without considering the first instance solved to optimality, is 6506.9 s and 7094.12 s with standard deviations, respectively, equal to 1950.82 and 118.74. These results remark that the MILP solver is able to exploit the available CPU time on this group of instances. This may be due to the fact that they contain only 25 customers and therefore can be considered easier than the large-sized ones.

Table 4 shows that the average Gap produced in the time limit of 7200 s by the MILP solver for the cloneless model is equal to 68.13%. Considering RKS, the average percentage deviation $Dev\%$ after 1800 s is equal to −7.24, whereas after 3600 s the solution quality is improved, being the average percentage deviation equal to −10.06. The average time to best of RKS is equal to 2277.5 s that is by far less than the average computational time required by the model (6506.9 s).

6.2. Results on large-sized instances

This section compares the results on the large-sized instances with 100 customers (Schneider et al., 2014) of the cloneless MILP model and of RKS, the latter starting from the solution of the RkDS.

On these instances, the cloneless MILP model always reached the time limit of 2 hours. Table 5 shows that RKS, within the time limit of 1800 s, found solutions on the average always better than the feasible ones determined by the MILP solver, with the only exception on the two instances, i.e., c101-21 and r101-21. Indeed, the $Dev\%$ is on average equal to −52.79. However, after one hour, RKS found solutions that are better on average than that of the model of about 2.88% on c101-21 and worse than that of the model of about 0.07% on r101-21. The $Dev\%$ is on average equal to −61.68 after one hour. These results highlight the effectiveness of RKS, taking also into account that they have been obtained within time limits that are respectively one fourth and half the one allowed to the MILP solver.

It is worth noting that, on almost all the instances, the best solution of the cloneless MILP model was indeed found after spending a lot of time for solving its continuous relaxation at the root node. Moreover, the standard deviation of the TimeBest is 2104.70 s for the MILP solver whereas it is 41.5 s and 50.70 s for the RKS after 1800 s and 3600 s of CPU time, respectively. On the contrary of the cloneless MILP model, the RKS found the best solution on average after 1756.00 s and 3520.90 s (depending on the two different CPU time limits) with a by far lower standard deviation, thus revealing to be more robust.

An example of the average behavior of RKS compared to the MILP solver applied to the cloneless model for instance c201-21 is given in Fig. 2. In this figure, the line associated with the RKS corresponds to the average result over 15 runs obtained for instance c201-21, whereas

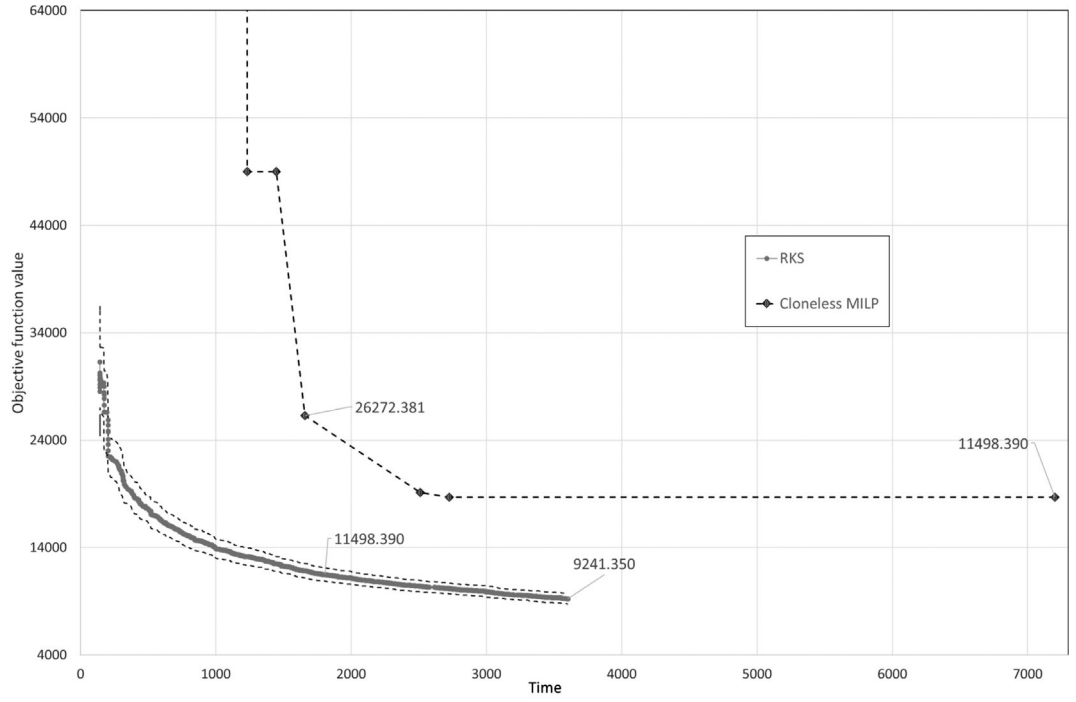


Fig. 2. An example of the searching behaviors of RKS and MILP solver.

Table 5
Result comparisons on large-sized instances.

Instance	Cloneless model				RanKS					
	Time limit = 7200				Time limit = 1800			Time limit = 3600		
	Obj	η	TimeBest	Gap(%)	Dev%	$\bar{\eta}$	TimeBest	Dev%	$\bar{\eta}$	TimeBest
c101-21	8482.263	12	7181.9	53.47	8.92	13.3	1791.6	−2.88	12.5	3489.1
c102-21	49204.285	100	2296.2	96.68	−71.54	16.3	1765.7	−75.27	15.2	3538.2
c103-21	49254.51	100	2731.3	97.19	−68.14	16.6	1771.3	−74.59	14.9	3530.4
c104-21	19082.92	37	1169.8	93.33	−20.81	16.7	1813.3	−34.93	14.4	3503.6
c105-21	17232.461	30	3325.3	86.25	−15.08	14.5	1782.0	−29.90	13.7	3518.4
c201-21	18674.189	32	6987.7	91.64	−41.49	7.3	1792.2	−54.63	6.7	3556.8
c202-21	49020.704	100	4231.0	97.34	−66.81	10.0	1783.7	−74.13	8.1	3567.7
c203-21	49020.704	100	4533.8	97.39	−66.21	10.4	1747.9	−74.38	7.7	3532.3
c204-21	30364.331	60	5695.3	95.84	−53.65	9.9	1643.0	−63.78	7.1	3497.3
c205-21	49020.704	100	2465.1	97.36	−75.13	8.9	1783.3	−81.36	7.1	3562.4
r101-21	9160.489	15	7219.5	24.39	11.00	17.7	1766.5	0.07	16.7	3531.4
r102-21	45835.469	100	1400.3	97.53	−70.44	23.5	1800.2	−73.55	21.3	3532.8
r103-21	296125.785	100	3457.7	99.64	−94.68	24.4	1760.9	−95.63	21.3	3539.3
r104-21	45939.895	100	3241.7	97.71	−59.87	29.5	1753.9	−68.75	23.0	3532.9
r105-21	18466.117	36	6746.5	90.48	−33.64	19.0	1754.2	−39.10	17.6	3533.5
r201-21	45935.872	100	2603.4	97.52	−73.93	9.1	1802.4	−79.02	7.4	3564.7
r202-21	45935.872	100	5419.5	97.66	−70.67	10.6	1728.9	−77.54	7.7	3554.5
r203-21	14665.021	27	7587.0	92.76	−1.17	10.5	1744.7	−24.55	7.9	3489.5
r204-21	14761.009	28	6693.3	92.9	−8.00	10.0	1622.2	−27.99	7.7	3408.7
r205-21	37187.986	79	5065.2	97.06	−65.70	8.9	1769.5	−74.32	7.3	3563.3
rc101-21	16574.141	30	4064.8	84.81	−19.44	20.3	1774.9	−27.76	18.9	3506.9
rc102-21	683432.596	100	1949.0	99.81	−97.89	21.9	1766.3	−98.07	20.5	3343.7
rc103-21	893141.008	100	3002.7	99.86	−98.22	24.4	1754.5	−98.41	21.8	3478.8
rc104-21	51495.042	100	7083.1	97.63	−56.89	30.5	1770.2	−69.23	23.6	3573.9
rc105-21	52297.614	100	1417.0	97.29	−69.50	22.3	1730.8	−74.14	20.2	3433.9
rc201-21	52383.019	100	2299.6	97.37	−69.80	9.9	1773.1	−77.48	8.7	3575.8
rc202-21	52383.019	100	4637.8	97.58	−65.63	10.9	1758.6	−75.86	9.8	3519.7
rc203-21	52383.019	100	7105.2	97.65	−63.56	13.0	1691.3	−74.13	9.7	3544.4
rc204-21	25089.783	46	908.6	95.13	−37.98	11.7	1718.1	−52.02	9.0	3529.2
rc205-21	52383.019	100	2741.3	97.56	−67.70	10.7	1765.1	−76.95	9.1	3573.2

Table 6
The combinations of parameters values analyzed.

k	k'	T^{ini}	T^{step}
24	8	120	30
24	8	480	120
60	20	120	30
60	20	480	120
60	40	120	30
60	40	480	120

Table 7
The statistics for the deviations for the initial solutions varying T^{ini}.

		Deviation from Z^*_{ini}		*p-value*
		Average	StDev	
T^{ini}	120	2.248	1.371	6.02E−09
	480	0.740	0.798	

the dashed lines show the 95% confidence interval with respect to the average.

6.3. Sensitivity analysis

This section illustrates the analysis performed to evaluate the sensitivity of RKS to a possible variation of the values of a subset of the input parameters. In particular, the maximum number of iterations for a fixed value of k, the minimum number of iterations during which a variable remains in the kernel after entering it, and the maximum number of iterations without improvements have been kept fixed to $I_K^{max} = 4$, $I_{KS}^{min} = 4$ and $I_N^{max} = 24$, respectively. In fact, from a set of preliminary tests, we observed that these values ensure a good trade-off between the computation time and the solution quality. Therefore, we decided to focus the analysis on both the parameters that influence the RkDS procedure and T^{step}. In addition, to reduce the computational burden, we performed this analysis on the three instances c201-21, r104-21 and rc205-21, respectively selected from the three classes C, R and RC and we run five repetitions of RKS. The time limit was set to 1800 and 3600 s on each instance for the combinations of the values of the parameters k, k', T^{ini} and T^{step} as shown in Table 6.

From Tables 7 and 8, we can first observe the influence of the parameters k, k' and T^{ini} on the quality of the starting solution found by RkDS. Both such tables report the Average and the Standard Deviation (StDev) of the $Dev\%$ values obtained with the different combinations of the parameters specified in the rows. Such deviations are computed with respect to the best initial solution found for each instance over the whole set of tests. In particular, the deviations are computed as $100 \cdot (Z_{ini}(i,p) - Z^*_{ini}(i))/Z^*_{ini}(i)$, where $Z^*_{ini}(i)$ is the objective value of the best initial solution found for the instance i, whereas $Z_{ini}(i,p)$ is the objective for i produced by RkDS with the set p of parameters fixed to the values as specified in the rows of the tables. In addition, in order to assess the significance of the differences between the obtained average values, we report in the tables also the p-value returned by the Friedman non-parametric test: if p-value is smaller than 1%, then we can reject the null hypothesis that the compared combinations generated not significantly different results.

Observing Table 7, we can hence conclude that, as might be expected, the initial solutions produced with $T^{ini} = 480$ s are significantly better than the ones with $T^{ini} = 120$ s.

Table 8 summarizes the results when the value of k and k' are changed. In this case, we can observe that the p-value from the Friedman test indicates the differences in the averages as not statistically significant.

Tables 9 and 10 report the results obtained analyzing the effects of the different combinations of the parameters values on the final solutions found by RKS. Also in this case, the deviations are given by $100 \cdot (Z(i,p) - Z^*(i))/Z^*(i)$, where $Z^*(i)$ is the objective value of the best final solution found for the instance i over the whole set of tests, whereas $Z(i,p)$ the objective found for i fixing the set of parameters p as in the tables rows. In particular, Table 9 considers the two combinations of T^{ini} and T^{step} tested to evaluate if increasing them could improve the quality of the RKS solutions. It is worth noting that, since the differences in the average values computed with both the T^{max} limits are not statistically significant, RKS appears sufficiently robust with respect to these two parameters.

Table 8
The statistics for the deviations for the initial solutions varying k and k'.

k	k'	Deviation from Z^*_{ini}		*p-value*
		Average	StDev	
24	8	1.226	1.092	
60	20	1.783	1.822	0.067
60	40	1.472	0.901	

An analogous conclusion can be drawn observing the p-value returned by the Friedman tests in Table 10, which reports the statistics on deviations from the final solutions when changing the values of k and k'. Therefore, the outcomes of the analysis performed remarks that RKS is not significantly sensitive both to the value of the parameters affecting the RkDS procedure and to the initial time limit for the restricted MILP models.

7. Conclusions and future works

In this paper, we studied the *Electric Vehicle Routing Problem with Time Windows and realistic Energy Consumption Rate* (EVRPTW-ECR). Compared to EVRP-TW, it includes some more real-life factors in the energy consumption model, e.g., the payload and the vehicle speed. Differently from (Xiao et al., 2019), where the EVRPTW-ECR was first introduced, in this paper, we also consider possible stops en-route at the Recharging Stations (RSs).

First of all, we modeled this problem as a cloneless Mixed Integer Linear Program where the vehicle speed is a continuous variable that can vary between a minimum and a maximum value. Moreover, the proposed formulation allows using more than once the same RS without introducing dummy copies of it. To this purpose, for each pair of customers i and j, the set S_{ij} is pre-computed, containing all the RSs that may be advantageous to visit, going from i to j. Moreover, through a pre-processing phase, we removed a-priori all the links, between two customers, that are infeasible with regard to either the vehicle cargo capacity, or the time windows or also the maximum time to return to the depot.

In order to efficiently address large-sized instances of the problem, a *Random Kernel Search* (RKS) matheuristic approach was also designed, inspired by the Kernel Search Algorithm of Angelelli et al. (2010). Moreover, in order to efficiently generate an initial solution for RKS, a *Random k-Degree Search* matheuristic (RkDS) was also introduced.

Since EVRPTW-ECR shares several aspects with the EVRP-TW, we used the benchmark instance sets with 100 customers introduced by Schneider et al. (2014), and we also derived a medium-sized one with 25 customers. On all the sets, we tested both the cloneless model and RKS.

The effectiveness of RKS was confirmed by the results obtained for both the sets. Indeed, RKS was able to improve on average, even in half an hour, the solutions found by solving the cloneless model in two hours of computations. We can also observe that RKS always greatly improved the initial solutions found by the RkDS matheuristic. However, RkDS was a fundamental component of RKS to overcome the possible computational difficulty in finding a feasible starting solution on the large-sized instances.

Finally, an analysis to evaluate the sensitivity of RKS to possible variation of some input parameters was also performed, showing that it is robust since it is not significantly sensitive both to the parameters

Table 9
The statistics for the deviations for the final solutions varying T^{ini} and T^{step}.

T^{ini}	T^{step}	T^{max} = 1800			T^{max} = 3600		
		Deviation from best			Deviation from best		
		Average	StDev	*p-value*	Average	StDev	*p-value*
120	30	0.6624	0.5428	0.8763	0.4348	0.3563	0.2359
480	120	0.4217	0.3895		0.3740	0.2878	

Table 10
The statistics for the deviations for the final solutions varying k and k'.

k	k'	T^{max} = 1800			T^{max} = 3600		
		Deviation from best			Deviation from best		
		Average	StDev	*p-value*	Average	StDev	*p-value*
24	8	0.477	0.370		0.365	0.248	
60	20	0.531	0.560	0.049	0.372	0.365	0.121
60	40	0.619	0.502		0.477	0.339	

of the RkDS procedure and to the initial time limit for the restricted MILP models.

Future works may concern the solution of further extension of EVRPTW-ECR where for instance the maximum speed on the arcs is time dependent to model the evolution of the traffic conditions along the day or where the maximum speed is known real time.

Data availability

Data will be made available on request.

References

Angelelli, E., Mansini, R., Speranza, M.G., 2010. Kernel search: A general heuristic for the multi-dimensional knapsack problem. Comput. Oper. Res. 37 (11), 2017–2026.

Archetti, C., Guastaroba, G., Huerta-Muñoz, D.L., Speranza, M.G., 2021. A kernel search heuristic for the multivehicle inventory routing problem. Int. Trans. Oper. Res. 28 (6), 2984–3013.

Barth, M., Boriboonsomsin, K., 2009. Energy and emissions impacts of a freeway-based dynamic eco-driving system. Transp. Res. D 14 (6), 400–410.

Barth, M., Younglove, T., Scora, G., 2005. Development of a heavy-duty diesel modal emissions and fuel consumption model. Technical Report UCB-ITS-PRR-2005-1, California PATH Program, Institute of Transportation Studies, University of California at Berkeley.

Basso, R., Kulcsár, B., Egardt, B., Lindroth, P., Sanchez-Diaz, I., 2019. Energy consumption estimation integrated into the electric vehicle routing problem. Transp. Res. D 69, 141–167.

Bektaş, T., Laporte, G., 2011. The pollution-routing problem. Transp. Res. B 45 (8), 1232–1250.

Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., Suraci, S., 2017. A three-phase matheuristic for the time-effective electric vehicle routing problem with partial recharges. Electron. Notes Discrete Math. 58, 95–102.

Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S., 2015. A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electron. Notes Discrete Math. 47, 221–228.

Carvalho, D.M., Nascimento, M.C., 2018. A kernel search to the multi-plant capacitated lot sizing problem with setup carry-over. Comput. Oper. Res. 100, 43–53.

Demir, E., Bektaş, T., Laporte, G., 2014. A review of recent research on green road freight transportation. European J. Oper. Res. 237 (3), 775–793.

Desaulniers, G., Errico, F., Irnich, S., Schneider, M., 2016. Exact algorithms for electric vehicle-routing problems with time windows. Oper. Res. 64 (6), 1388–1405.

Erdoğan, S., Miller-Hooks, E., 2012. A green vehicle routing problem. Transp. Res. E 48 (1), 100–114.

Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G., 2014. A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transp. Res. E 71, 111–128.

Feo, T.A., Resende, M.G., 1995. Greedy randomized adaptive search procedures. J. Global Optim. 6 (2), 109–133.

Ferro, G., Paolucci, M., Robba, M., 2020. Optimal charging and routing of electric vehicles with power constraints and time-of-use energy prices. IEEE Trans. Veh. Technol. 69 (12), 14436–14447.

Filippi, C., Guastaroba, G., Huerta-Muñoz, D., Speranza, M., 2021. A kernel search heuristic for a fair facility location problem. Comput. Oper. Res. 132, 105292.

Filippi, C., Guastaroba, G., Speranza, M., 2016. A heuristic framework for the bi-objective enhanced index tracking problem. Omega 65, 122–137.

Franke, T., Rauh, N., Günther, M., Trantow, M., Krems, J.F., 2016. Which factors can protect against range stress in everyday usage of battery electric vehicles? Toward enhancing sustainability of electric mobility systems. Human Factors 58 (1), 13–26.

Froger, A., Mendoza, J.E., Jabali, O., Laporte, G., 2019. Improved formulations and algorithmic components for the electric vehicle routing problem with nonlinear charging functions. Comput. Oper. Res. 104, 256–294.

Gobbi, A., Manerba, D., Mansini, R., Zanotti, R., 2019. A kernel search for a patient satisfaction-oriented nurse routing problem with time-windows. IFAC-PapersOnLine 52 (13), 1669–1674.

Goeke, D., 2019. Granular tabu search for the pickup and delivery problem with time windows and electric vehicles. European J. Oper. Res. 278 (3), 821–836.

Goeke, D., Schneider, M., 2015. Routing a mixed fleet of electric and conventional vehicles. European J. Oper. Res. 245 (1), 81–99.

Guastaroba, G., Savelsbergh, M., Speranza, M.G., 2017. Adaptive kernel search: A heuristic for solving mixed integer linear programs. European J. Oper. Res. 263 (3), 789–804.

Hanafi, S., Mansini, R., Zanotti, R., 2020. The multi-visit team orienteering problem with precedence constraints. European J. Oper. Res. 282 (2), 515–529.

Hiermann, G., Puchinger, J., Ropke, S., Hartl, R.F., 2016. The electric fleet size and mix vehicle routing problem with time windows and recharging stations. European J. Oper. Res. 252 (3), 995–1018.

Joo, H., Lim, Y., 2018. Ant colony optimized routing strategy for electric vehicles. J. Adv. Transp. 2018.

Kancharla, S.R., Ramadurai, G., 2020. Electric vehicle routing problem with non-linear charging and load-dependent discharging. Expert Syst. Appl. 160, 113714.

Karakatič, S., 2021. Optimizing nonlinear charging times of electric vehicle routing with genetic algorithm. Expert Syst. Appl. 164, 114039.

Keskin, M., Çatay, B., 2016. Partial recharge strategies for the electric vehicle routing problem with time windows. Transp. Res. C 65, 111–127.

Kisacikoglu, M.C., Ozpineci, B., Tolbert, L.M., 2011. Reactive power operation analysis of a single-phase EV/PHEV bidirectional battery charger. In: 8th International Conference on Power Electronics-ECCE Asia. IEEE, pp. 585–592.

Koç, Ç., Jabali, O., Mendoza, J.E., Laporte, G., 2019. The electric vehicle routing problem with shared charging stations. Int. Trans. Oper. Res. 26 (4), 1211–1243.

Labbé, M., Martínez-Merino, L.I., Rodríguez-Chía, A.M., 2019. Mixed integer linear programming for feature selection in support vector machine. Discrete Appl. Math. 261, 276–304.

Lander, L., Cleaver, T., Rajaeifar, M.A., Nguyen-Tien, V., Elliott, R.J., Heidrich, O., Kendrick, E., Edge, J.S., Offer, G., 2021. Financial viability of electric vehicle lithium-ion battery recycling. Iscience 24 (7), 102787.

Lee, C., 2021. An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. J. Oper. Res. Soc. 72 (7), 1461–1485.

Lin, J., Zhou, W., Wolfson, O., 2016. Electric vehicle routing problem. Transp. Res. Procedia 12, 508–521.

Löffler, M., Desaulniers, G., Irnich, S., Schneider, M., 2020. Routing electric vehicles with a single recharge per route. Networks 76 (2), 187–205.

Macrina, G., Pugliese, L.D.P., Guerriero, F., 2020. The green-vehicle routing problem: A survey. In: Modeling and Optimization in Green Logistics. Springer, pp. 1–26.

Macrina, G., Pugliese, L.D.P., Guerriero, F., Laporte, G., 2019. The green mixed fleet vehicle routing problem with partial battery recharging and time windows. Comput. Oper. Res. 101, 183–199.

Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G., 2017. The electric vehicle routing problem with nonlinear charging function. Transp. Res. B 103, 87–110.

Mor, A., Speranza, M.G., 2020. Vehicle routing problems over time: a survey. 4OR 18 (2), 129–149.

Pelletier, S., Jabali, O., Laporte, G., 2019. The electric vehicle routing problem with energy consumption uncertainty. Transp. Res. B 126, 225–255.

Santos-Peñate, D.R., Campos-Rodríguez, C.M., Moreno-Pérez, J.A., 2020. A kernel search matheuristic to solve the discrete leader-follower location problem. Netw. Spat. Econ. 20 (1), 73–98.

Schiffer, M., Klein, P.S., Laporte, G., Walther, G., 2021. Integrated planning for electric commercial vehicle fleets: A case study for retail mid-haul logistics networks. European J. Oper. Res. 291 (3), 944–960.

Schneider, M., Stenger, A., Goeke, D., 2014. The electric vehicle-routing problem with time windows and recharging stations. Transp. Sci. 48 (4), 500–520.

Schuster, S.F., Bach, T., Fleder, E., Müller, J., Brand, M., Sextl, G., Jossen, A., 2015. Nonlinear aging characteristics of lithium-ion cells under different operational conditions. J. Energy Storage 1, 44–53.

Shao, S., Guan, W., Ran, B., He, Z., Bi, J., 2017. Electric vehicle routing problem with charging time and variable travel time. Math. Probl. Eng. 2017.

So, K.M., Gruber, P., Tavernini, D., Karci, A.E.H., Sorniotti, A., Motaln, T., 2020. On the optimal speed profile for electric vehicles. IEEE Access 8, 78504–78518.

Solomon, M., 1987. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper. Res. 35 (2), 254–265.

Toth, P., Vigo, D., 2002. The Vehicle Routing Problem. SIAM.

Wu, X., He, X., Yu, G., Harmandayan, A., Wang, Y., 2015. Energy-optimal speed control for electric vehicles on signalized arterials. IEEE Trans. Intell. Transp. Syst. 16 (5), 2786–2796.

Xiao, Y., Zuo, X., Kaku, I., Zhou, S., Pan, X., 2019. Development of energy consumption optimization model for the electric vehicle routing problem with time windows. J. Clean. Prod. 225, 647–663.

Zhang, S., Gajpal, Y., Appadoo, S., Abdulkader, M., 2018. Electric vehicle routing problem with recharging stations for minimizing energy consumption. Int. J. Prod. Econ. 203, 404–413.

Zuo, X., Xiao, Y., You, M., Kaku, I., Xu, Y., 2019. A new formulation of the electric vehicle routing problem with time windows considering concave nonlinear charging function. J. Clean. Prod. 236, 117687.