Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-023-09858-x

A hybrid simulated annealing and variable neighborhood search algorithm for the close-open electric vehicle routing problem

Themistoklis Stamadianos[1] · **Nikolaos A. Kyriakakis**[1] · **Magdalene Marinaki**[1] · **Yannis Marinakis**[1]

Accepted: 9 May 2023

Abstract
Electric Vehicles (EVs) are the future of transportation, but due to their battery and charging technology they cannot yet directly replace traditional vehicles. Nonetheless, EVs are a great option for city-logistics, due to the small distances and their zero local emissions. In this paper, a novel variant of the Electric Vehicle Routing Problem (EVRP), called Close-Open EVRP (COEVRP), is presented. It considers ending EV trips at Charging Stations, as opposed to other EVRP variants that only allow for en-route charging. This new variant follows a traditional routing scheme, allowing EVs to recharge only at the end of their route. The objective is to minimize energy consumption, as well as the number of vehicles. The energy consumption function takes into account the weight of the transported items. A mathematical formulation for the problem is presented and small instances were solved using a commercial solver. To solve larger instances, a hybrid metaheuristic combining Simulated Annealing and Variable Neighborhood Search algorithm was employed and thoroughly tested.

Keywords Routing · Electric vehicles · Vehicle routing problem · Close-open

1 Introduction

Electric Vehicles (EVs) are gaining significant traction in Vehicle Routing Problems (VRPs) of all kinds. The Electric Vehicle Routing Problem (EVRP) and its variants are among the

✉ Yannis Marinakis
marinakis@ergasya.tuc.gr

Themistoklis Stamadianos
tstamadianos@tuc.gr

Nikolaos A. Kyriakakis
nkyriakakis@tuc.gr

Magdalene Marinaki
magda@dssl.tuc.gr

[1] School of Production Engineering and Management, Technical University of Crete, University Campus, Akrotiri 73100, Chania, Greece

most prominent research domains in logistics today. This trend is highly justifiable by the urgency to transition to green and sustainable means of transportation.

The current EV battery technology cannot yet directly compete with Internal Combustion Engine (ICE) vehicles in terms of the potential driving distance (range). EVs have to stop for a recharge far sooner than a conventional vehicle which might not need a refueling stop for the entirety of its daily operations. While passenger EVs have made leaps forward in the last decade, freight EVs still face many problems since they have to carry heavy payloads, which are very detrimental to energy consumption. Therefore, EVs are mainly used for urban logistics and for small payload deliveries. Small-size electric vans with a decent range are already available on the market and used in such cases, but heavily constrained to ensure they do not run out of energy.

Although vehicles do not require a full charge at each charging stop, a significant amount of time is, inevitably, spent waiting to recharge, while the driver has to be compensated for the time he spent there. Furthermore, charging stops along the way have to be on fast chargers, so drivers do not spend too much time waiting. On the other hand, frequent fast charging leads to battery degradation, and the high temperatures associated with fast charging lower their efficiency, shortening the vehicle's range. Another cost worth considering is the price of electricity used for charging, which may differ during the day. In the variant proposed in this paper, we assume that all vehicles start their journey with the maximum energy they can store.

Other uncertainties may occur too. If a vehicle visits a Charging Station (CS) with no available chargers, it may need to stay there and waste more time if its range does not allow it to travel to another one. Power outages may affect delivery schedules as well. If the vehicle visits a CS with multiple charger types or charging speeds, there is no guarantee that the desired one will be available. These are only some examples of problems that may occur in the current underdeveloped charging infrastructure.

Due to the very high recharging times for electric vehicles, en-route recharging is wasteful during drivers' working hours. Thus, the proposed approach limits the recharging at the end of the routes which is applicable to real short-haul transportation operations. Long haul, multi-day transportation EVs are still in development by truck manufacturers, therefore these applications are not within the scope of this research.

The purpose of this research is to explore the option of recharging vehicles after the completion of their deliveries, not necessarily at the depot, but at other available CSs too. The novel routing model that is proposed allows carriers to perform their deliveries as usual, and worry about charging later. It lessens the burden of time-consuming but important recharging by displacing it to the end of the route. This approach could be very beneficial for real-life applications, given the previously described state of modern charging infrastructure.

To aid the routing of EVs in the scenario above, the novel Close-Open Electric Vehicle Routing Problem (COEVRP) is introduced. This research is the first to allow open EV routing. COEVRP is a contribution to both Close-Open VRP and EVRP. Close-Open VRP has yet to attract the attention of researchers, and subsequently, is insufficiently researched. In opposition, EVRP has been extensively researched, but the Close-Open variant has not been discussed in the literature so far. Here, the first mathematical formulation for COEVRP is presented, along with a metaheuristic algorithm.

The proposed metaheuristic algorithm is a hybrid Simulated Annealing (SA) and Variable Neighborhood Search (VNS) algorithm, referred to as SA/VNS. Both algorithms have been successfully implemented in various VRPs in the past. In this hybrid version, SA replaces the mechanism that rejects moves to worse solutions, and considers such moves but with a lower probability.

A new set of instances for COEVRP was created by modifying an existing set of instances for Electric VRP (EVRP). Small instances of up to fifteen customers were solved both by the Gurobi Optimizer and SA/VNS. Larger instances were solved exclusively by SA/VNS.

The structure of the research paper is the following. A literature review of research on the Close-Open VRP and other related problems is presented in Section 2. Then, the model of the COEVRP is presented in Section 3, while Section 4 gives an insight into the solution algorithm. Section 5 discusses the results, and lastly, Section 6 gives the overall conclusions and future research.

2 Literature review

The related literature concerns Open VRPs, Close-Open VRPs and Electric VRPs. The following subsections contain brief literature reviews for all three variants.

2.1 Open vehicle routing problems

Open VRP was first presented in [1]. He argued that a company with high delivery and low pickup demands would benefit from the employment of contractors to handle part of their deliveries. In [2], the first OVRP heuristic was presented, with a clustering first and routing second method. A tabu search algorithm was developed in [3], using the nearest neighbor and an improvement phase that follows, based on previous literature. In [4], developed a Decision Support System for OVRP. In [5], a threshold-accepting methodology was presented, and in [6] a meta-heuristic employing a single parameter was developed. A tabu-search heuristic method for OVRP was developed in [7]. [8] presented a variant of OVRP in which the trips end at specific nodes, and consider delivery deadlines. A mathematical model for OVRP with Time Windows was presented in [9], along with a heuristic algorithm to solve it. In [10] a record-to-record travel algorithm for OVRP was presented, using a fixed-size neighbor list. Initial solutions were generated using a sweep algorithm. In [11], a capacitated OVRP was solved with a branch-and-cut method. A VNS algorithm for OVRP was proposed in [12]. An attribute-based hill-climbing heuristic method was presented in [13]. In [14], a destruction and repair algorithm was presented. A novel local search meta-heuristic for OVRP was developed in [15], considering two different objective functions. In [16] a hybrid evolution strategy for OVRP was presented. A variant of OVRP with a capacitated heterogeneous fleet was proposed in [17].

More recently, in [18], the authors presented a heuristic for OVRP with the objective of minimizing the length of the longest route, and in [19] an improved Bumble Bee Mating algorithm was introduced for the OVRP, where the flying equation is replaced by an Iterated Local Search (ILS) procedure to achieve better results and lower CPU times. In [20], an OVRP with demand uncertainties was solved and incorporated an optimization model to attain low transportation costs and to fulfill the demand of their customers as best as possible. These demands are not totally unknown, but they lay between certain ranges. The concept of Cross-Docking for OVRP is introduced in [21]. They explore delivery outsourcing for both the pickup and delivery parts. A Simulated Annealing algorithm provides the results. In [22] a VNS-inspired algorithm was developed for newspaper deliveries in a real-world application of OVRP. In [23], a gravitational emulation local search algorithm was developed. A multi-depot OVRP was presented in [24] and an algorithm was developed that combines the Adaptive Large Neighborhood Search with local search procedures. The same variant was solved in

[25], and in [26]. An OVRP with capacity and travel distance constraints was solved in [27], using a genetic algorithm with competitive results. The OVRP with Decoupling Points was introduced in [28], along with an ILS algorithm. The authors considered the employment of more than one carrier for deliveries and conducted a case study for a Canadian retailer. Another popular variant is the Multi-Depot OVRP (MD-OVRP). In [29], and in [30], two heuristics were proposed for the MD-OVRP. The first was a Tabu search and Multiple Neighborhood Search hybrid and the latter was a memory-based ILS.

Green OVRP is the variant most closely related to COEVRP, given their similar objectives. [31] aimed to minimize fuel emissions cost for a Green OVRP with Time Windows, assuming a hired fleet of vehicles and drivers.

2.2 Close-open vehicle routing problems

The Close-Open VRP (COVRP) is a recent addition to VRPs and has received little study. It was introduced in [32], and later in [33], as a Close-Open Mixed VRP, with the purpose of making deliveries with a fleet of owned and hired vehicles. Owned vehicles return to the depot, while the hired ones are assumed to end their trip right after the last customer gets served. They presented a Mixed Integer Program (MIP) and a memetic algorithm to solve it. In [34] (published online in 2013) the COVRP with Time Windows was presented, along with a Variable Neighborhood Search (VNS) to solve it. They solved real-life instances and instances from the literature. Variable Neighborhood Descend (VND), a variant of VNS, and a Greedy Randomized Adaptive Search Procedure (GRASP) were also employed. In [35] a hybrid Ant Colony Optimization algorithm for the COVRP was presented with Time Windows and fuzzy constraints. In [36], the authors expanded upon the previous variants by introducing a combination of mixed fleets and multiple depots for COVRP. A MIP and a metaheuristic algorithm were presented, with the goal of cost minimization. In [37] a multi-depot COVRP with a heterogeneous fleet was presented and a mathematical formulation for the problem was proposed. The authors approached the problem from a multi-criteria perspective and focused on the solution selection process rather than solving instances of the problem. In [38], a Knowledge-Guided Neighborhood Search Algorithm was presented for COVRP. Lastly, in [39], COVRP with mixed fleets and multiple collecting centers was presented. The goal was to collect perishable products and deliver them to collection centers. Both owned and rented vehicles were included. Three solution approaches were presented and evaluated on real-world data.

2.3 Green and electric vehicle routing problems

The research paper of [40] was the first to consider a refueling scheme similar to that of EVRP. Later in [41], the Green VRP (GVRP) was introduced. In this variant, Alternative Fuel Vehicles (AFVs) were used in place of traditional ones, subsequently, EVRP can be considered a variant of GVRP that uses EVs. EVRP has received great attention since its introduction, in the form of EVRP with Time-Windows and Recharging Stations in [42]. EVRP is a variant that considers the energy expenditure of the vehicles, given they are powered by battery-stored electricity. Furthermore, the energy replenishing time is related to the amount of stored energy, while in GVRP refueling is considered to be related to the level of fuel.

In [42] linear charging and consumption rate were assumed, as well as full recharging. In [43], both the required energy and the charging infrastructure were considered. The partial recharging option was first considered in [44]. In [45] an Adaptive Large Neighborhood Search algorithm for a partial recharging variant was proposed, while in [46] a three-phase VNS Branching algorithm for the same problem was presented. Experiments to determine the best recharging strategy were carried out in [47], which was found to be the multiple and partial recharging policy. Furthermore, it was the first exact solution method proposed in the field of EVRP. In [48] the GVRP with partial refueling was studied. EVRP with three different charging speeds for the vehicles was proposed in [49]. A variant with multiple charging station technologies was solved in [50]; however, only small instances were solved by the presented algorithm. In [51] the energy consumption was minimized instead of distance and it was proven to be a more suitable objective.

Non-linear charging functions for the EVs were first considered in [52]. Most researchers tackle non-linear charging functions with piece-wise linear functions. In [53] an exact algorithm was presented for EVRPTW with non-linear charging. In [54] a variant of EVRP was studied, considering a non-linear charging function and taking into account the load of the vehicle.

Waiting time at a CS is a common issue when refueling is considered. In [55] a GVRP variant was solved with refueling waiting times, while in [56], and in [57], EVRP variants were presented. In [57], partial recharging was also considered.

A real-life EV test was presented in [58]. An EV was used by a courier service provider and proven very effective while demonstrating that en-route recharging is not always necessary. In some cases, the EV was able to complete two full days of work before requiring a recharge. It should be noted that the EV was a small van with $40kWh$ of battsery capacity and used for light parcel transportation. Furthermore, the tests were carried out in an urban environment that would favor the use of EVs.

A thorough review on the subject of EVRP was presented in [59]. They highlight the lack of realistic problems in the literature, traced to the intricate EV attributes. They examine charging strategies, and energy consumption, and they present an EVRP model of their own. The computational experiments are conducted, along with experiments that support the use of the largest possible battery size and variable vehicle speed instead of fixed values.

In the most recent Green VRP review, [60], the authors mention the range and battery-related issues of EVs and the underdeveloped infrastructure. The growing interest in this subject is also recognized. Another GVRP review, [61], was published at around the same time, reaching the same overall conclusions.

To the best of the authors knowledge, the only research on OVRP that considers AFVs is [62], in which a multi-objective variant of Open GVRP was solved. To date, no other research on Open VRP or Close-Open VRP has considered any type of AFV.

3 The close-open electric vehicle routing problem

The goal of the present research is to introduce an EVRP with a new approach to recharge, aiming to find better ways of handling EVs in the realm of logistics operations. This is achieved by incorporating the Close-Open aspect of the problem, allowing EVs to end their trips at the most convenient charging station, which may not be the depot. The novel COEVRP will be the basis for more research on realistic handling scenarios.

In this section, the main characteristics of COEVRP and its mathematical formulation are presented.

3.1 Problem description

The COEVRP is very closely related to EVRP since they both employ EVs in place of internal combustion vehicles. In this case, the EV fleet is considered to be homogeneous and has maximum capacity constraints for both the stored energy and the payload. Energy consumption is the main concern with EVs and is directly influenced by the payload the EV carries. As items get delivered to their recipients, weight is removed from the vehicle and the energy consumption gets lower. The weight of the item or items of each delivery is known a priori. No weight is carried back to the depot or to a CS. The stark difference of COEVRP to other EVRP variants is that battery recharging may only take place after the deliveries. EVs may end their trips at a Charging Station and not the depot; therefore, the problem is considered to be of the close-open type.

The problem aims at minimizing both the total energy consumption of the operation and the number of EVs used. The energy consumption minimization goal is in line with the electric nature of these vehicles. As EVs are associated with high acquisition costs, minimizing the total number of necessary vehicles is essential.

The assumptions made for COEVRP are listed below:

- EVs start with maximum energy.
- Each customer is visited only once, and no split deliveries are allowed.
- CSs are assumed to have available chargers at any time.

3.2 Mathematical formulation

The mathematical formulation for the novel COEVRP is influenced by the work of Liu et al. [33], where the close-open variant is first introduced, and the work of Jie et al. [63], where an EVRP variant with battery swapping is presented. The original close-open concept as presented in [33] has been expanded to consider the use of EVs, while the formulation of the two-echelon EVRP presented in [63], was studied and adapted for the energy consumption constraint and the payload calculation structure of the EVs.

V_D denotes the set of the depot nodes, one in this case. Set V_C includes the customer nodes. The number of customers is n_c. The Charging Stations (CS) are defined in set CS and n_{CS} of them are available. V_E contains the V_D and the CS sets that together denote the potential nodes where vehicles may end their trips. K is the set of EVs. The maximum payload capacity is Q, and the maximum energy is E. d_{ij} represents the distance between nodes i and j and q_i represents the demand of vertex i, which is 0 for the Depot and the Charging Stations.

As for the decision variables, x_{ijk} describes whether or not the k EV traversed the arc(i, j). f_{ijk} is tasked with saving the payload of k EV when it arrives at j, coming from i. This way, the payload of each vehicle at each step is easily determined and used for energy consumption calculations.

Table 1 presents a comprehensive list of all the sets, decision variables, and other parameters of the formulation it follows.

Objective Function:

$$w_1 \times (\sum_{i\in V}\sum_{j\in V}\sum_{k\in K}(1 + f_{ijk}) \times d_{ij} \times x_{ijk})/E + w_2 \times \sum_{k\in K}\sum_{j\in V} x_{0jk} \tag{1}$$

Table 1 Notation used for the COEVRP formulation

	Sets & Characteristics
V_D	Depot set, $V_D = \{v_D\}$
V_C	Customers, $V_C = \{v_{C1}, v_{C2}, ..., v_{Cn_c}\}$
CS	Charging Stations set, $CS = \{CS_1, CS_2, ..., CS_{n_{cs}}\}$
V_E	Ending nodes set, $V_E = V_D \cup CS$
V	Superset containing all the above sets, $V = V_D \cup V_C \cup CS$
d_{ij}	Distance from node i to node j
n_c	Number of customers
n_{cs}	Number of Charging Stations (CS)
q_i	payload demand of customer i
K	Set of EVs
Q	Maximum payload capacity of EVs
E	Maximum energy capacity of EVs
w_1	Weight for the energy consumption in the objective function
w_2	Weight for the number of vehicles in the objective function
	Decision Variables
f_{ijk}	Stores the payload of k EV coming from i.
x_{ijk}	When arc (i, j) is traversed, $x_{ijk} = 1$, otherwise $x_{ijk} = 0$.

Subject to:

$$\sum_{k \in K} \sum_{j \in V} x_{ijk} = 1, \forall i \in V_C, i \neq j \tag{2}$$

$$\sum_{j \in V} x_{ijk} = \sum_{j \in (V_C \cup V_D)} x_{jik}, \forall i \in V_C, k \in K, i \neq j \tag{3}$$

$$\sum_{j \in V_C} x_{0jk} \leq 1, \forall k \in K \tag{4}$$

$$\sum_{j \in (V_C \cup V_D)} \sum_{k \in K} f_{jik} - \sum_{j \in V} \sum_{k \in K} f_{ijk} = q_i, \forall i \in V_C, i \neq j \tag{5}$$

$$q_j \times x_{ijk} \leq f_{ijk} \leq (Q - q_j) \times x_{ijk}, \forall i \in V, j \in V, k \in K \tag{6}$$

$$\sum_{i \in V} \sum_{j \in V} (1 + f_{ijk}) \times d_{ij} \times x_{ijk} \leq E, \forall k \in K \tag{7}$$

$$x_{ijk} = 0, \forall i \in V_D, j \in V_E, k \in K \tag{8}$$

$$x_{ijk} = 0, \forall i \in CS, j \in V, k \in K \tag{9}$$

$$x_{iik} = 0, \forall i \in V, k \in K \tag{10}$$

$$x_{ijk} \in \{0, 1\}, \forall i \in (V_C \cup V_D), j \in (V_C \cup V_E), k \in K, i \neq j \tag{11}$$

The objective function (1) of this research has two goals. The first one is to minimize energy consumption, and the second is to minimize the number of vehicles employed. In this scalarized approach, each of the two objectives is weighted. When solved without applying any weights, the objective function would tend toward extreme solutions. The total energy consumption is divided by the maximum energy, in order to be in the same order of magnitude as the other part of the objective function. The energy consumption is derived from the research of [64], who solved an energy-minimizing VRP.

Constraints (2) ensure each customer gets visited by a vehicle exactly once. Constraints (3) balance the input and output flow at each node. They include the start and endpoints of each route. Constraints (4) ensure each vehicle leaves at most once from the depot. Constraints (5), state that the payload when reaching i minus the payload when leaving i must be equal to the demand at node i. Constraints (6) provide the lower and upper bound for the vehicle load. If the arc (i, j) is not traversed, then the load is 0. Constraints (7) keep in check the energy of the vehicles since a maximum energy capacity is a limiting factor. Constraints (8) to (10) restrict unwanted connections from being made. Lastly, constraints (11) enforce the binary limitations of the decision variable x.

Furthermore, the objective function and constraints (7) are non-linear. However, a piecewise-linear approximation is automatically generated by the employed commercial solver, subsequently, it has been omitted.

4 The proposed hybrid algorithm

As a variant of VRP, COEVRP is an NP-hard problem to solve. Subsequently, large instances of COEVRP cannot be solved optimally in a timely manner using commercial solvers. Heuristic and meta-heuristic methods have to be employed instead.

To solve COEVRP, the Greedy Randomized Adaptive Search Procedure (GRASP) algorithm was used to generate initial solutions. Then a hybrid Variable Neighborhood Search (VNS) algorithm combined with a Simulated Annealing (SA) algorithm is used to improve upon them. Further analysis of the algorithm is provided in the following subsections.

4.1 Initial solution construction

The Greedy Randomized Adaptive Search Procedure, or GRASP, was introduced in [65]. This heuristic is well-established in the VRP field and can provide initial solutions of great diversity. Subsequently, the initial solutions in this research are constructed using a variant of the GRASP algorithm.

In the construction phase of GRASP, a customer list called Restricted Candidate List (RCL) is created. Each of the available customers is evaluated and placed on the list. If the RCL is full, the weakest candidate is removed. When all available customers have been checked, one of the customers in the RCL is randomly selected to visit next. This construction phase is used to generate the initial solutions for COEVRP. The metric by which customers are selected for the RCL is distance.

This method allows the algorithm to further explore the solution space and has a greater chance of escaping a local minimum. An important parameter that determines the diversity of the solutions is the size of the list. A small list may store only a few possible alternatives, whereas a bigger one will tend toward more random solutions.

The pseudocode for the RCL construction method is provided in Algorithm 1. The RCL is initially empty. Each of the available customers is checked. If the customer is close enough then the customer is included in the RCL. At the end of the process, a customer is selected from the RCL at random and is added to the route.

Algorithm 1 RCL construction algorithm

```
Require: Available Customers
customer = { }
RCL = { }                                        ▷ (Restricted Customer List)
for Each available customer do
    if available customer is eligible then
        RCL.add_customer(available customer)
    end if
end for
if RCL.empty() == false then
    customer = RCL.random_customer()
end if
return customer
```

4.2 Simulated annealing

Simulated Annealing (SA) is a popular metaheuristic for VRPs. Annealing is a process in metallurgy, in which the heating and cooling of a metal part take place in a controlled manner, in order to change the structure of the atoms that comprise it and get the desired mechanical characteristics at the end of the process.

The SA algorithm simulates this process during the local search procedure. In each iteration of the algorithm, the probability of accepting a move to a new solution over the current one is calculated for each potential move. Similarly to traditional annealing, the SA algorithm starts with a high initial temperature T_{max}. In this state, the probability of choosing any solution is high. With each new iteration of the algorithm, the temperature gradually drops, affecting the probability calculations; thus, making fewer random moves. An important attribute of the algorithm is having a non-zero probability of accepting a sub-optimal move. The algorithm terminates when temperature T has reached a certain threshold (cooled down).

The probability of moving to a better solution is equal to 1. If the solution under discussion is not better, then the selection probability is calculated according to (12), where c' is the cost of the candidate solution, and c is the cost of the current solution. As the temperature drops, so does the probability. The temperature in each iteration is calculated using the (13), where $iter$ is the number of the current iteration and $iter_{max}$ is the maximum number of iterations. This cool-down rate is the most important part of SA since it has to be slow enough to allow solutions to converge.

$$P_{accept} = e^{(c'-c)/T} \tag{12}$$

$$T = 1 - (iter + 1)/iter_{max} \tag{13}$$

4.3 Integrating simulated annealing in variable neighborhood search

Variable Neighborhood Search (VNS) was introduced in [66]. It is also a popular metaheuristic in VRP applications, owing to its simple, yet effective mechanics. Three distinct phases take place, shaking, local search, and move. In the shaking phase, a random solution is selected, then local search algorithms are applied to it. If moving to another solution is beneficial then the move is applied. This is repeated until the termination condition is met, which in this case is the temperature reaching its lowest value, zero.

This integration scheme is inspired by the Variable Neighborhood Simulated Annealing (VNSA) algorithm presented in [67]. A significant difference from the SA/VNS presented in this paper is the use of Geographical Neighborhood Structures (GNSs). The local search operators differ as well, and they are employed only when the GNS cannot provide a solution. SA was also included in the VNS in [42]; however, that variant of VNS had a Tabu Search component as well.

By integrating SA in VNS, each potential move in the local search phase has a chance of being selected, even moves that result in a worse state, which would be otherwise excluded by the VNS algorithm. Nonetheless, allowing the algorithm to indefinitely accept worse results would be a bad practice; therefore, if the algorithm moves three times in a row towards a worse solution, it is forced to move to a better solution in the next iteration.

Three local search operators were used,*1-1 Inter-route Swap*, *1-1 Intra-route Swap*, and *1-0 Relocate*.

- In *1-1 Inter-route Swap*, two customers c_1 and c_2 from two different vehicles i and j, respectively, exchange their positions. The energy expenditure has to be calculated for each potential move. Both customers get selected at random following a uniform distribution. In case any of the resulting routes are infeasible, then the solution is not kept.
- In *1-1 Intra-route Swap*, two customers c_1 and c_2 from the same vehicle exchange their positions. Despite originating from the same vehicle, the energy expenditure must be calculated again to ensure feasibility, even if weight remains the same, since the distances will differ.
- In *1-0 Relocate*, a customer c_1 gets removed from the vehicle i and placed in a specific position in vehicle j. Once again, the energy expenditure has to be calculated for both vehicles.

The pseudocode of the proposed SA/VNS algorithm is presented in Algorithm 2. In an iterative process, the algorithm explores the solutions space by improving an incumbent solution step by step. Starting from an initial solution generated using the initial solution construction method of GRASP, the algorithm applies local search operators in order to move the incumbent solution within the feasible solution space. These moves are controlled using the SA acceptance criterion. The temperature is given an initial value $T_{max} = 1$. While the temperature T remains above zero ($T_{cool} = 0$), two routes are randomly selected, and potential moves are generated, evaluated, and saved to the list of potential moves, called $candidates$. Then, one of them is selected as the move to be made. At the last step of each iteration, the temperature of the SA is reduced, following the rule presented in (13). The algorithm returns the best-found solution of the search.

The SA/VNS structure may be reminiscent of the General VNS (GVNS) implementation with Union VND (U-VND); however, there are some major differences that set them apart. While the GVNS implementations have a traditional shaking procedure, in the SA/VNS the shaking intensity is static, with only one move being made in each iteration of the VNS. In addition, the U-VND method selects the best available move to make in each case, while in SA/VNS the move to make is determined by the selection probability of each move, based on the current temperature T. Thus, any of the neighbors may be selected.

5 Computational study

To test the quality of the proposed solution algorithm computational experiments were carried out on the instances described in the next subsection. The results of the algorithm are compared to the results provided on the mathematical model for the COEVRP, presented

Algorithm 2 SA/VNS outline

Require: $instance\_data, vns_{iter}, iter_{max}, T_{cool}, T_{max}$
$S \leftarrow generate\_initial\_solution(instance\_data)$
$S_{best} \leftarrow S$
$T \leftarrow T_{max}$ ▷ *Initial temperature for SA
$iter \leftarrow 0$
while $T > T_{cool}$ **do**
    $iter \leftarrow iter + 1$
    $candidates \leftarrow \{\}$ ▷ *Save possible moves
    for vns_{iter} **do**
        $\{route1, route2\} \leftarrow random\_routes(S)$
        $candidates.add\_to\_list(intra\_route\_swap(route1, T))$
        $candidates.add\_to\_list(inter\_route\_swap(route1, route2, T))$
        $candidates.add\_to\_list(reloc\_1\_0(route1, route2, T))$
    end for
    if $candidates.empty() == false$ **then**
        $S \leftarrow select\_move(candidates)$ ▷ *Based on P_{accept}
        if $S_{best}.cost < S.cost$ **then**
            $S_{best} \leftarrow S$
        end if
    end if
    $T = 1 - (iter + 1)/iter_{max}$
end while
return Solution S_{best}

in Section 3, by the commercial solver Gurobi Optimizer (version 9.1.2), on an academic license.

The proposed solution algorithm was implemented in C++ (Standard C++20) and compiled in Microsoft Visual Studio 2019 (Visual C++ 14.20). The mathematical formulation was solved by Gurobi Optimizer using Python 3.9.1. All tests were carried out on an Intel(R) Core(TM) i3-8130u @2.20GHz with 6 GB of DDR4 RAM @2400Mhz.

5.1 Problem instances

To create new instances, appropriate for COEVRP, the instances introduced in [42] were used, which were Based on the instances for VRP with Time Windows (VRPTW), presented in [68]. The intention is to use instances from the literature and adapt them to work for COEVRP.

Besides the Time Window data which were not applicable to this research paper, the instances include the coordinates of the charging stations, the depot, and the customers along with the customer demand. Furthermore, the characteristics of the electric vehicles are described, but not the total number of available vehicles, therefore, it was determined by performing various tests for each instance. A preliminary test for both Gurobi and SA/VNS was carried and out of their solutions, the lowest number of vehicles was kept for each instance. If the two methods used a different amount of vehicles, then the results would not be comparable.

These instances belong to three different types, “r”, “c”, and “rc”, corresponding to random customer distribution, clustered, and a combination of both. To be able to use these instances for testing the COEVRP mathematical model, and the heuristic algorithm, the parameter referring to the energy capacity of the EVs is altered, depending on the type of problem. Problems of type “c” and “rc” were given vehicles with 2333 units of energy capacity, while problems of type "r" were given vehicles with 1818 units of energy capacity. These numbers are about thirty times the original values of energy capacity, provided in [42]. In their research, the energy calculation method took into account only the traveled distance, while in this research, the energy consumption is affected both by the distance and the weight which necessitates the use of different maximum energy limits for the EVs. The names of the COEVRP instances are kept the same as in [42]. The instance names of the small instances include the number of customers as "C5", "C10", and "C15", denoting 5, 10, and 15 customers, respectively.

Following experimentation with the instances solved in this paper, it was determined that $w_1 = 10$, and, $w_2 = 2$ for the objective function provided acceptable solutions for all instance sizes. Acceptable solutions are considered those that achieve a balance between energy consumption and the number of vehicles used and are not heavily dependent on either part of the objective function.

5.2 Computational experiments

5.2.1 SA/VNS parameters

The maximum number of initial solutions for each instance was one of the first to be considered. For small instances, twenty times the number of the nodes of the instance was selected, as they are more than enough. For large instances, it was set equal to half of the number of customers. These initial solutions differ from each other since they are generated using the solutions generation algorithm of GRASP.

The number of SA/VNS iterations is a highly important parameter. It is commonly referred to as cool-down time. One of the first and most important tasks is to ensure a slow enough cool-down time for the SA to perform optimally. A quick cool-down would provide worse results, even compared to a simple VNS. Since the possibility of moving to worse solutions exists, if the algorithm is not given enough time to recover from it, its performance will end up being worse overall, despite prohibiting more than three sequential bad moves to occur. This is very evident in large instances.

To further investigate this aspect of COEVRP, fifteen-customer instances were solved using three different cool-down times. The results of this experiment are presented in Table 2. The first row presents the average energy of the initial solutions, as obtained in line 1 of Algorithm 2. The following rows present the solutions obtained by different values of parameter $iter_{max}$. Three fifteen-customer instances were selected, one of each type. The presented results are the average of ten runs. As observed, the number of iterations is vitally important for ensuring high-quality results. Following extensive testing, a hundred iterations of SA/VNS proved to be adequate for five and ten-customer instances, while for fifteen-customer instances three-thousand iterations were necessary.

Another essential part of any VNS variant is the number of possible LS moves generated, vns_{iter}. In each iteration of SA/VNS, up to 75 possible moves may be generated. Values larger than 75 did not yield better results and slowed down the algorithm. Table 3, presents the results of testing the SA/VNS with a different number of vns_{iter} moves per iteration. The first column presents the instance name, followed by the energy consumption, the vns_{iter} moves, and finally, the execution time in seconds. As shown, $vns_{iter} = 75$ was the ideal number of iterations. The instances from the previous test were used here too. The presented results are the average of ten runs. The difference in execution time could be more noticeable for larger instances.

Table 2 SA/VNS Iterations Test

Instance	$iter_{max}$	$Energy_{avg}$	$Gap_{BFS}\%$
c208C15	-	9864.52	16.77
c208C15	30	9425.09	12.89
c208C15	300	8433.51	2.64
c208C15	3000	8210.68	0
r209C15	-	7895.17	14.84
r209C15	30	7835.06	7.45
r209C15	300	7692.33	573
r209C15	3000	7251.46	0
rc204C15	-	13321.56	15.82
rc204C15	30	12738.92	11.97
rc204C15	300	11749.11	4.56
rc204C15	3000	11213.68	0

5.2.2 LS operators

To further emphasize the importance of each included LS operator a number of tests were carried out, the results of which are presented in Table 4. The first column presents the instance name, the following column presents the LS operators employed, then the energy consumption, execution time, and gap from the best energy consumption value. The presented results are the average of ten runs. The operators were tested both in isolation and, also, coupled with a second one. The results indicate that the combination of *1-1 Intra-route Swap* and *1-0 Relocate* has the same performance as the combination of all three LS operators, in an almost identical execution time. While, in some cases, it may be possible to achieve the same results, adding the *1-1 Inter-route Swap* operator does not harm performance and provides better results in some cases. This is demonstrated in Table 5. The first column contains the instance and the second the number of vehicles. The average energy consumption achieved using all three of the LS operators is presented in the third column, followed by the average

Table 3 vns_{iter} Test

Instance	Energy	vns_{iter}	Time (s)
c208C15	8210.68	225	48.3
c208C15	8210.68	150	41.2
c208C15	8210.68	75	35.8
c208C15	8210.68	38	39.1
r209C15	7251.46	225	59.3
r209C15	7251.46	150	46.1
r209C15	7251.46	75	40.3
r209C15	7251.46	38	48.3
rc204C15	11213.68	225	45.0
rc204C15	11213.68	150	41.8
rc204C15	11213.68	75	38.5
rc204C15	11213.68	38	39.1

Table 4 Comparison of all possible LS combinations for SA/VNS

Instance	LS operators	Energy	Time (s)	Gap_{best} %
rc204C15	Intra Swap	13074.93	27.3	14.24
rc204C15	Inter Swap	11648.32	28.2	3.73
rc204C15	Relocate	12744.41	28.6	12.01
rc204C15	Intra & Inter Swap	11779.91	34.1	4.81
rc204C15	Intra Swap & Relocate	11213.68	38.3	0.00
rc204C15	Inter Swap & Relocate	11513.13	41.0	2.6
rc204C15	All	11213.68	38.5	0.00

energy consumption achieved using only Intra Swap & Relocate, and the gap between the solutions. A small, but noticeable, gap of 0.70% can be observed.

In the following subsections, the results of the COEVRP tests on the modified instances from [42] are presented.

5.2.3 Small instances

For instances of up to fifteen customers, both the results of the Gurobi Optimizer and the proposed algorithm are given.

In Tables 6, 7 and 8, the results of the five, ten, and fifteen-customer instances are presented, respectively. In each table, the first column contains the instance name, as in [42]. The second column contains the number of vehicles employed. The number of vehicles is the same for both methods; thus, it is not reported separately for each one. The third column presents the Best found Solution (BFS) among Gurobi and SA/VNS. The BFS refers only to energy consumption. The fourth column presents the energy consumption of the Gurobi solution, while the fifth column presents its gap from the BFS as a percentage. The final

Table 5 Comparison of two LS combinations for SA/VNS

Instance	Vehicles	All LS operators $Energy_{avg}$	Intra Swap & Relocate $Energy_{avg}$	Gap %
c103C15	7	8666.84	8711.91	0.52
c106C15	4	5465.16	5465.16	0.00
c202C15	6	7891.61	7891.61	0.00
c208C15	6	8210.68	8210.68	0.00
r102C15	6	5062.99	5113.62	1.00
r105C15	5	5826.06	5848,78	0.39
r202C15	7	8001.84	8112.29	2.63
r209C15	7	7251.46	7279.02	0.38
rc103C15	5	7560.69	7562.96	0.03
rc108C15	7	10440.30	10538.44	0.94
rc202C15	6	8770.42	8770.42	0.00
rc204C15	7	11213.68	11491.78	2.48
Average	N/A	7863.48	7924.72	0.70

Table 6 5-customer instances

Instance	BFS *Vehicles*	BFS *Energy*	Gurobi *Energy*	Gurobi $Gap_{BFS}\%$	SA/VNS $Energy_{avg.}$	SA/VNS $Energy_{best}$	SA/VNS $Gap_{BFS}\%$
c101C5	3	3662.75	3662.75	0.00	3662.75	3662.75	0.00
c103C5	3	2698.09	2698.09	0.00	2698.09	2698.09	0.00
c206C5	3	2704.75	2704.75	0.00	2704.75	2704.75	0.00
c208C5	3	4779.94	4779.94	0.00	4779.94	4779.94	0.00
r104C5	3	2134.89	2134.89	0.00	2134.89	2134.89	0.00
r105C5	2	1381.10	1381.10	0.00	1381.10	1381.10	0.00
r202C5	3	1568.07	1568.07	0.00	1568.07	1568.07	0.00
r203C5	3	2828.08	2828.08	0.00	2828.08	2828.08	0.00
rc105C5	4	5019.44	5019.44	0.00	5019.44	5019.44	0.00
rc108C5	3	5579.50	5579.50	0.00	5579.50	5579.50	0.00
rc204C5	4	5100.98	5100.98	0.00	5100.98	5100.98	0.00
rc208C5	3	2465.94	2465.94	0.00	2465.94	2465.94	0.00
Average	-	3326.96	3326.96	0.00	3326.96	3315.71	0.00

three columns concern the SA/VNS algorithm. More specifically, the average results of ten SA/VNS runs, as well as the best SA/VNS solution, and again, the gap from the BFS as a percentage.

While the objective function of the mathematical model contains both the number of vehicles and the energy consumption, since the number of vehicles does not change, BFS refers solely to the energy consumption of the solution. Nonetheless, it is important to include the number of vehicles in the objective function, since the solutions that minimize the energy consumption are not necessarily the same with solutions that minimize the number of EVs, as using an additional vehicle may have potential energy savings. Whether the energy savings

Table 7 10-customer instances

Instance	BFS *Vehicles*	BFS *Energy*	Gurobi *Energy*	Gurobi $Gap_{BFS}\%$	SA/VNS $Energy_{avg.}$	SA/VNS $Energy_{best}$	SA/VNS $Gap_{BFS}\%$
c101C10	5	7754.12	7754.12	0.00	7754.12	7754.12	0.00
c104C10	5	5776.82	5776.82	0.00	5776.82	5776.82	0.00
c202C10	5	6853.08	6853.08	0.00	6853.08	6853.08	0.00
c205C10	4	6004.50	6004.50	0.00	6004.50	6004.50	0.00
r102C10	5	4516.26	4516.26	0.00	4516.26	4516.26	0.00
r103C10	4	3082.25	3082.25	0.00	3082.25	3082.25	0.00
r201C10	5	3844.80	3844.80	0.00	3844.80	3844.80	0.00
r203C10	4	3511.19	3511.19	0.00	3511.19	3511.19	0.00
rc102C10	5	6451.89	6451.89	0.00	6451.89	6451.89	0.00
rc108C10	4	5371.85	5371.85	0.00	5371.85	5371.85	0.00
rc201C10	4	5376.41	5376.41	0.00	5376.41	5376.41	0.00
rc205C10	5	5841.67	5841.67	0.00	5841.67	5841.67	0.00
Average	-	5365.40	5365.40	0.00	5365.40	5365.40	0.00

Table 8 15-customer instances

	BFS		Gurobi		SA/VNS		
Instance	*Vehicles*	*Energy*	*Energy*	$Gap_{BFS}\%$	$Energy_{avg.}$	$Energy_{best}$	$Gap_{BFS}\%$
c103C15	7	8666.84	8666.85	0.00	8666.84	8666.84	0.00
c106C15	4	5465.16	5465.16	0.00	5465.16	5465.16	0.00
c202C15	6	7891.61	8051.07	2.02	7891.61	7891.61	0.00
c208C15	6	8210.68	8329.90	1.45	8210.68	8210.68	0.00
r102C15	6	4934.03	5235.34	6.11	5062.99	4934.03	0.00
r105C15	5	5826.06	5826.06	0.00	5826.06	5826.06	0.00
r202C15	7	8001.84	8001.84	0.00	8001.84	8001.84	0.00
r209C15	7	7251.46	7251.46	0.00	7251.46	7251.46	0.00
rc103C15	5	7560.69	7726.32	2.19	7560.69	7560.69	0.00
rc108C15	7	10235.80	10555.89	3.13	10440.30	10235.80	0.00
rc202C15	6	8770.42	8791.27	0.24	8770.42	8770.42	0.00
rc204C15	7	11213.68	11213.68	0.00	11213.68	11213.68	0.00
Average	-	7835.69	7926.24	1.26	7863.48	7835.69	0.00

commend the purchase of an additional vehicle or not, is up to the Decision Maker in each situation. This is true for large instances as well.

For all of the five, and, ten-customer instances, both SA/VNS and Gurobi achieved the same energy consumption. In all cases Gurobi finished before reaching the time limit (1800 seconds), subsequently, the results are optimal.

For fifteen-customer instances, Gurobi reached the time limit in nine out of the twelve instances. For those instances, optimality can not be guaranteed. On the other hand, SA/VNS outperformed Gurobi in half of the twelve instances and provided better solutions in a fraction of the time. In two out of those six instances, SA/VNS had a small gap between the best and the average results.

The average gap between Gurobi and the BFS was 1.26%. This gap should not be viewed in solidarity, but, in conjunction with the execution time. SA/VNS managed to achieve the presented results in a small fraction of the time it took Gurobi.

A visual comparison of the results instance c202C15 is provided in Fig. 1. In the center of the figure is the depot, designated with a **0**. The charging stations are represented by **1**, **2**, **3**, and **4**. The rest of the points which are not in bold are the customers. The difference between the two displayed solutions is extremely small, as the only dissimilarity is the removal of node 12 from route {0, 12, 15, 6, 1} and its addition to route {0, 19, 14, 4}.

The most significant difference between the two methods, was the time it took to achieve these results, as execution time differed substantially between Gurobi and SA/VNS. The recorded times for five, ten, and fifteen-customer instances are presented in Table 9. The time is reported in seconds. Five-customer instances were solved in a matter of seconds from both Gurobi and SA/VNS, albeit a bit faster by SA/VNS. The difference became more noticeable for ten-customer instances, where Gurobi took 100 seconds on average, while SA/VNS took less than eight on average. For fifteen-customer instances, Gurobi reached the 1800-second (30 minutes) time limit, in all but three cases. Subsequently, optimality cannot be guaranteed for the instances that reached the time limit. On the other hand, SA/VNS took on average less than forty seconds, with the worst execution time being about fifty-one seconds for instance rc202C15 and the quickest being almost sixteen seconds for r202C15,

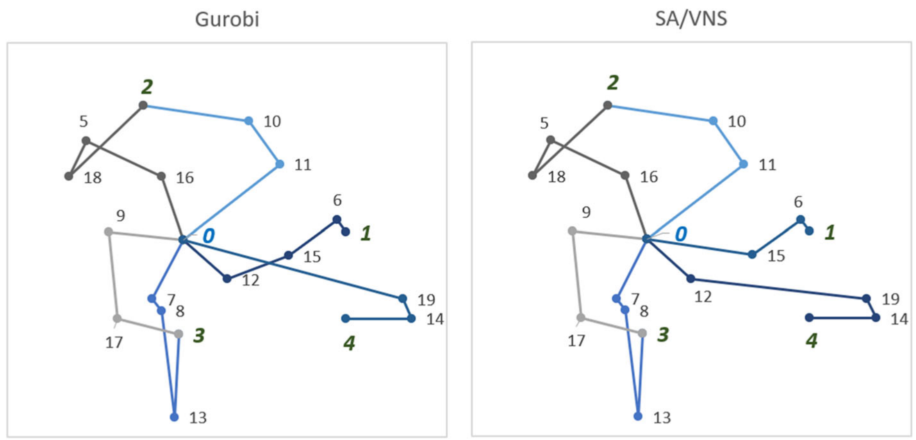

Fig. 1 Comparison of the results for instance c202C15 (CSs in bold)

coinciding with the quickest solve time from Gurobi, at a hundred and thirty seconds. This extremely high difference in execution time and the gap of Gurobi indicate that Gurobi is not suitable for larger problems.

5.2.4 Large instances

The larger instances consist of 100 customers and 21 CSs, including the depot. To display how each component of the SA/VNS improves the solutions, they get compared to results obtained from VNS, as presented in Algorithm 2, but without the SA. In essence, only moves that improve the solution may be added to the list of candidates. The maximum number of

Table 9 Execution time for small instances (in seconds)

5-customers			10-customers			15-customers		
Instance	Gurobi	SA/VNS	Instance	Gurobi	SA/VNS	Instance	Gurobi	SA/VNS
c101C5	0.8	0.4	c101C10	37.2	7.8	c103C15	1800.0	27.9
c103C5	1.1	0.3	c104C10	19.1	5.3	c106C15	1800.0	37.8
c206C5	1.6	0.2	c202C10	604.3	6.5	c202C15	1800.0	41.5
c208C5	3.9	0.2	c205C10	97.0	8.6	c208C15	1800.0	35.8
r104C5	0.9	0.2	r102C10	4.0	4.9	r102C15	1800.0	39.0
r105C5	0.8	0.2	r103C10	29.0	11.1	r105C15	1340.0	57.7
r202C5	0.9	0.5	r201C10	4.0	8.0	r202C15	130.6	15.7
r203C5	1.1	0.4	r203C10	204.0	11.6	r209C15	440.6	40.3
rc105C5	1.1	0.2	rc102C10	4.0	1.1	rc103C15	1800.0	50.1
rc108C5	1.8	0.3	rc108C10	19.0	7.5	rc108C15	1800.0	24.3
rc204C5	1.2	0.2	rc201C10	109.0	10.2	rc202C15	1800.0	51.0
rc208C5	0.8	0.5	rc205C10	126.0	6.9	rc204C15	1800.0	38.5
Average	1.3	0.3	Average	104.7	7.5	Average	1509.26	38.3

initial solutions generated was equal to half of the number of customers of the problems, which were fifty in all cases. The SA/VNS iterations were set to five thousand.

Table 10 presents the results of testing. The values displayed are the average of 10 runs. The first column contains the instance name. The following two columns contain the number of vehicles (Veh.) and energy consumption (Energy) of the BFS. The following three columns contain the energy consumption, execution time (in seconds), and the gap of energy consumption from the energy consumption of the BFS, for the VNS solutions. The last three columns contain the same information as the previous three, but, for the SA/VNS solutions. Once again, BFS refers only to energy consumption, since the number of vehicles remains the same. If the number of vehicles differed, then the results would not be comparable.

As indicated in Table 10, SA/VNS outperformed the VNS algorithm; demonstrating the value of the hybridization. All BFS energy values were derived from SA/VNS, subsequently, the SA/VNS gap from the BFS is 0%. The average gap of the VNS (w/o SA) from the BFS was 5.71%. The lowest gap it was able to achieve was 3.55%, for instance, c101, while its worst gap was 8.33% for instance r111.

Computational times were on average marginally higher for SA/VNS, compared to VNS. The average time for SA/VNS was 92.9 seconds and 91.0 seconds for VNS. The instance that took SA/VNS the longest to solve was c205 at 146.2 seconds, while instance r108 took only 54.1 seconds.

6 Conclusions

The novel Close-Open Electric Vehicle Routing Problem (COEVRP) was introduced in this paper, along with a mathematical formulation for it. The main limitations of electric vehicles, energy and payload capacity, were considered. The main goal of the presented COEVRP is to provide an alternative routing methodology, in which EVs are not allowed to recharge between deliveries, but only after all the deliveries have been made. It is safe to estimate that such a model would be a great fit for city logistics.

The proposed model is a more realistic approach for the adoption of EVs in supply chains and especially last-mile logistics since the battery technology cannot yet compete with internal combustion engines. Furthermore, the charging stations' infrastructure currently available is not robust enough, since they are few in number, may suffer failures more often than a gas station, may be fully occupied, or not support the desired charging speed. Factors such as the above would pose a significant risk and could compromise routing operations.

The initial solutions of COEVRP were generated using the construction algorithm inspired by the construction phase of the GRASP algorithm. To improve upon them, a hybrid algorithm combining SA and VNS was developed to solve COEVRP. While the original VNS algorithm would only allow moves that immediately improve the solution, SA/VNS gives an opportunity for non-optimal moves to be selected, providing further diversification and exploration of the solution space. The SA/VNS hybrid algorithm was tested on modified instances from the literature. For small problems, both SA/VNS and Gurobi Optimizer were employed and compared, while larger instances were only solved using SA/VNS. Three Local Search operators were included, *1-1 Intra-route Swap*, *1-1 Inter-route Swap*, and *1-0 Relocate*.

The energy consumption results of SA/VNS and Gurobi for five and ten-customer instances coincided, while SA/VNS outperformed Gurobi in half of the fifteen-customer instances. The most important difference between the two methods is that they differed significantly in execution time. While it was not so significant for five-customer instances,

Table 10 100-customer instances

Instance	Veh.	BFS	VNS			SA/VNS		
		Energy	Energy	Time	$Gap_{BFS}\%$	Energy	Time	$Gap_{BFS}\%$
c101	16	68836.8	71283.0	75.8	3.55	68836.8	78.8	0.00
c102	16	69113.8	71659.8	69.8	3.68	69113.8	69.4	0.00
c103	17	68366.7	71314.6	67.6	4.31	68366.7	67.5	0.00
c104	16	69178.8	71809.0	65.6	3.80	69178.8	69.4	0.00
c105	16	69219.2	72106.4	68.3	4.17	69219.2	70.2	0.00
c106	17	68392.7	71261.7	59.0	4.19	68392.7	71.2	0.00
c107	16	68832.4	71972.0	70.7	4.56	68832.4	72.2	0.00
c108	16	68936.4	72138.3	67.3	4.64	68936.4	74.9	0.00
c109	16	69068.2	71706.5	73.4	3.82	69068.2	75.9	0.00
c201	17	71201.6	75133.1	131.1	5.52	71201.6	131.8	0.00
c202	17	71342.6	74944.2	116.4	5.05	71342.6	125.4	0.00
c203	17	71648.6	75454.5	141.0	5.31	71648.6	131.3	0.00
c204	17	71384.2	74901.6	130.9	4.93	71384.2	130.3	0.00
c205	16	72197.7	75900.4	137.0	5.13	72197.7	146.6	0.00
c206	17	71491.2	75106.0	128.9	5.06	71491.2	117.1	0.00
c207	17	71391.6	74858.5	122.7	4.86	71391.6	141.3	0.00
c208	17	72008.2	74885.6	131.6	4.00	72008.2	139.9	0.00
r101	14	53334.3	56729.1	51.8	6.37	53334.3	60.2	0.00
r102	13	54713.9	58509.0	63.5	6.94	54713.9	69.2	0.00
r103	13	54460.6	58572.6	60.3	7.55	54460.6	63.3	0.00
r104	13	54306.9	58489.8	60.7	7.70	54306.9	62.7	0.00
r105	13	54437.4	58577.4	58.0	7.61	54437.4	65.0	0.00
r106	13	54305.6	58023.9	59.4	6.85	54305.6	61.7	0.00
r107	13	54468.5	58792.9	68.6	7.94	54468.5	64.6	0.00
r108	14	53323.3	56402.1	52.5	5.77	53323.3	54.1	0.00
r109	13	54305.3	57653.8	62.7	6.17	54305.3	70.0	0.00
r110	14	53082.0	56398.0	49.0	6.25	53082.0	56.3	0.00
r111	13	54049.2	58551.7	59.4	8.33	54049.2	63.9	0.00
r112	13	54279.8	58130.0	62.2	7.09	54279.8	63.6	0.00
r201	14	53451.0	56765.3	76.2	6.20	53451.0	96.5	0.00
r202	13	54215.9	58469.4	108.0	7.85	54215.9	100.1	0.00
r203	13	54702.3	58386.1	108.7	6.73	54702.3	103.6	0.00
r204	14	53801.6	57011.3	83.0	5.97	53801.6	96.4	0.00
r205	14	52960.3	57107.3	84.3	7.83	52960.3	85.1	0.00
r206	13	54661.2	58673.1	100.1	7.34	54661.2	112.8	0.00
r207	13	54516.5	58189.6	101.6	6.74	54516.5	94.2	0.00
r208	13	54560.2	58900.1	99.5	7.95	54560.2	105.8	0.00
r209	13	54542.8	58403.0	107.4	7.08	54542.8	110.8	0.00
r210	13	54490.3	58794.9	104.7	7.90	54490.3	108.9	0.00
r211	13	54633.8	58391.5	112.5	6.88	54633.8	101.8	0.00
rc101	17	74572.2	78098.7	80.9	4.73	74572.2	85.9	0.00

Table 10 continued

Instance	Veh.	BFS Energy	VNS Energy	VNS Time	VNS $Gap_{BFS}\%$	SA/VNS Energy	SA/VNS Time	SA/VNS $Gap_{BFS}\%$
rc102	17	74132.9	78155.6	82.8	5.43	74132.9	86.9	0.00
rc103	17	74675.6	78812.9	83.8	5.54	74675.6	93.6	0.00
rc104	18	73519.7	77770.0	74.8	5.78	73519.7	78.7	0.00
rc105	17	74918.4	77982.1	85.7	4.09	74918.4	87.5	0.00
rc106	17	74436.9	78307.5	84.2	5.20	74436.9	87.4	0.00
rc107	17	74253.5	78265.9	81.2	5.40	74253.5	83.3	0.00
rc108	18	73430.2	77108.8	74.5	5.01	73430.2	71.6	0.00
rc201	17	74585.2	78400.5	116.1	5.12	74585.2	134.9	0.00
rc202	17	74108.2	77916.7	116.7	5.14	74108.2	124.2	0.00
rc203	18	73240.6	76708.1	112.4	4.73	73240.6	105.5	0.00
rc204	17	74243.8	78361.3	134.9	5.55	74243.8	124.8	0.00
rc205	17	74514.4	77956.5	133.6	4.62	74514.4	125.9	0.00
rc206	17	74459.7	77794.3	127.1	4.48	74459.7	130.1	0.00
rc207	17	74650.9	78467.7	136.5	5.11	74650.9	113.6	0.00
rc208	17	74759.5	77982.3	121.3	4.31	74759.5	83.0	0.00
Average	N/A	64762.8	68365.1	91.0	5.71	64762.8	92.9	0.00

ten-customer instances took more than ten times longer for Gurobi and fifteen-customer instances took about forty times longer, on average. Subsequently, the large, one hundred-customer instances were solved using SA/VNS only. To demonstrate the significance of the SA component of the SA/VNS hybrid algorithm, the results were compared to VNS without the SA component. SA/VNS lowered the energy consumption compared to VNS by 5.71%.

Moreover, various tests were carried out on the operators and variables of SA/VNS. The number of initial solutions generated was set to twenty times the number of nodes of each instance. The cool-down time of the SA, determined by the number of SA/VNS iterations was tested next. Four tests were carried out, none, thirty, three-hundred, and three-thousand iterations respectively. Clearly, the last test with three-thousand iterations had the best results. Next, the number of maximum LS moves generated in each iteration was considered. After testing, a maximum of seventy-five moves was proven to be ideal. Lastly, the LS operators of SA/VNS were tested. Evidently, all of the LS operators proved to be necessary to achieve the desired results.

Future research should focus on adding even more realistic elements to the COEVRP model and account for more parameters. Alternative solution acceptance criteria could be explored in the hybrid algorithm implementation, such as threshold acceptance, hill-climbing, and record-to-record travel. Different energy consumption functions could be used as well. However, a balance between energy expenditure and the number of vehicles should be kept, as EVs are still an expensive investment. A small-scale real-life test would also be a valuable addition to the literature, paving the way for more accurate energy consumption estimates. Furthermore, the COEVRP model could be adapted for use in Multi-Depot problems, using an objective function that would also cater to vehicle distribution.

Acknowledgements The research work was supported by the Hellenic Foundation for Research and Innovation (HFRI) under the HFRI PhD Fellowship grant (Fellowship Number: 6334.)

Funding Open access funding provided by HEAL-Link Greece.

Data Availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request.

Declarations

Competing Interests The authors declare that they have no competing interests.

References

1. Schrage, L.: Formulation and structure of more complex/realistic routing and scheduling problems. Networks **11**(2), 229–232 (1981)
2. Sariklis, D., Powell, S.: A heuristic method for the open vehicle routing problem. Journal of the Operational Research Society **51**(5), 564–573 (2000)
3. Brandão, J.: A tabu search algorithm for the open vehicle routing problem. European Journal of Operational Research **157**(3), 552–564 (2004)
4. Tarantilis, C.D., Diakoulaki, D., Kiranoudis, C.T.: Combination of geographical information system and efficient routing algorithms for real life distribution operations. European Journal of Operational Research **152**(2), 437–453 (2004)
5. Tarantilis, C.D., Ioannou, G., Kiranoudis, C.T., Prastacos, G.P.: A threshold accepting approach to the open vehicle routing problem. RAIRO-Operations Research **38**(4), 345–360 (2004)
6. Tarantilis, C.D., Ioannou, G., Kiranoudis, C.T., Prastacos, G.P.: Solving the open vehicle routeing problem via a single parameter metaheuristic algorithm. Journal of the Operational research Society **56**(5), 588–596 (2005)
7. Fu, Z., Eglese, R., Li, L.Y.: A new tabu search heuristic for the open vehicle routing problem. Journal of the operational Research Society **56**(3), 267–274 (2005)
8. Aksen, D., Özyurt, Z., Aras, N.: Open vehicle routing problem with driver nodes and time deadlines. Journal of the Operational Research Society **58**(9), 1223–1234 (2007)
9. Repoussis, P.P., Tarantilis, C.D., Ioannou, G.: The open vehicle routing problem with time windows. Journal of the Operational Research Society **58**(3), 355–367 (2007)
10. Li, F., Golden, B., Wasil, E.: The open vehicle routing problem: Algorithms, large-scale test problems, and computational results. Computers & operations research **34**(10), 2918–2930 (2007)
11. Letchford, A.N., Lysgaard, J., Eglese, R.W.: A branch-and-cut algorithm for the capacitated open vehicle routing problem. Journal of the Operational Research Society **58**(12), 1642–1651 (2007)
12. Fleszar, K., Osman, I.H., Hindi, K.S.: A variable neighbourhood search algorithm for the open vehicle routing problem. European Journal of Operational Research **195**(3), 803–809 (2009)
13. Derigs, U., Reuter, K.: A simple and efficient tabu search heuristic for solving the open vehicle routing problem. Journal of the Operational Research Society **60**(12), 1658–1669 (2009)
14. Salari, M., Toth, P., Tramontani, A.: An ilp improvement procedure for the open vehicle routing problem. Computers & Operations Research **37**(12), 2106–2120 (2010)
15. Zachariadis, E.E., Kiranoudis, C.T.: An open vehicle routing problem metaheuristic for examining wide solution neighborhoods. Computers & Operations Research **37**(4), 712–723 (2010)
16. Repoussis, P.P., Tarantilis, C.D., Bräysy, O., Ioannou, G.: A hybrid evolution strategy for the open vehicle routing problem. Computers & Operations Research **37**(3), 443–455 (2010)

17. Li, X., Leung, S.C., Tian, P.: A multistart adaptive memory-based tabu search algorithm for the heterogeneous fixed fleet open vehicle routing problem. Expert Systems with Applications **39**(1), 365–374 (2012)
18. López-Sánchez, A., Hernández-Díaz, A.G., Vigo, D., Caballero, R., Molina, J.: A multi-start algorithm for a balanced real-world open vehicle routing problem. European Journal of Operational Research **238**(1), 104–113 (2014)
19. Marinakis, Y., Marinaki, M.: A bumble bees mating optimization algorithm for the open vehicle routing problem. Swarm and Evolutionary Computation **15**, 80–94 (2014)
20. Cao, E., Lai, M., Yang, H.: Open vehicle routing problem with demand uncertainty and its robust strategies. Expert Systems with Applications **41**(7), 3569–3575 (2014)
21. Vincent, F.Y., Jewpanya, P., Redi, A.P.: Open vehicle routing problem with cross-docking. Computers & Industrial Engineering **94**, 6–17 (2016)
22. Şevkli, A.Z., Güler, B.: A multi-phase oscillated variable neighbourhood search algorithm for a real-world open vehicle routing problem. Applied Soft Computing **58**, 128–144 (2017)
23. Hosseinabadi, A.A.R., Vahidi, J., Balas, V.E., Mirkamali, S.S.: Ovrp gels: solving open vehicle routing problem using the gravitational emulation local search algorithm. Neural Computing and Applications **29**(10), 955–968 (2018)
24. Lahyani, R., Gouguenheim, A.-L., Coelho, L.C.: A hybrid adaptive large neighbourhood search for multi-depot open vehicle routing problems. International Journal of Production Research **57**(22), 6963–6976 (2019)
25. Shen, L., Tao, F., Wang, S.: Multi-depot open vehicle routing problem with time windows based on carbon trading. International journal of environmental research and public health **15**(9), 2025 (2018)
26. Lalla-Ruiz, E., Mes, M.: Mathematical formulations and improvements for the multi-depot open vehicle routing problem. Optimization Letters **15**(1), 271–286 (2021)
27. Ruiz, E., Soto-Mendoza, V., Barbosa, A.E.R., Reyes, R.: Solving the open vehicle routing problem with capacity and distance constraints with a biased random key genetic algorithm. Computers & Industrial Engineering **133**, 207–219 (2019)
28. Atefi, R., Salari, M., Coelho, L.C., Renaud, J.: The open vehicle routing problem with decoupling points. European Journal of Operational Research **265**(1), 316–327 (2018)
29. Soto, M., Sevaux, M., Rossi, A., Reinholz, A.: Multiple neighborhood search, tabu search and ejection chains for the multi-depot open vehicle routing problem. Computers & Industrial Engineering **107**, 211–222 (2017)
30. Brandão, J.: A memory-based iterated local search algorithm for the multi-depot open vehicle routing problem. European Journal of Operational Research **284**(2), 559–571 (2020)
31. Niu, Y., Yang, Z., Chen, P., Xiao, J.: Optimizing the green open vehicle routing problem with time windows by minimizing comprehensive routing cost. Journal of Cleaner Production **171**, 962–971 (2018)
32. Liu, R., Jiang, Z., Hu, H., Yao, S.: A memetic algorithm for the close-open mixed vehicle routing problem. In: 2010 IEEE International Conference on Industrial Engineering and Engineering Management, pp. 728–732 (2010). IEEE
33. Liu, R., Jiang, Z.: The close-open mixed vehicle routing problem. European Journal of Operational Research **220**(2), 349–360 (2012)
34. Brito, J., Expósito, A., Moreno, J.A.: Variable neighbourhood search for close-open vehicle routing problem with time windows. IMA Journal of Management Mathematics **27**(1), 25–38 (2016)
35. Brito, J., Martínez, F.J., Moreno, J., Verdegay, J.L.: An aco hybrid metaheuristic for close-open vehicle routing problems with time windows and fuzzy constraints. Applied Soft Computing **32**, 154–163 (2015)
36. Azadeh, A., Farrokhi-Asl, H.: The close-open mixed multi depot vehicle routing problem considering internal and external fleet of vehicles. Transportation Letters **11**(2), 78–92 (2019)
37. Tavakkoli-Moghaddam, R., Meskini, M., Nasseri, H., Tavakkoli-Moghaddam, H.: A multi-depot close and open vehicle routing problem with heterogeneous vehicles. In: 2019 International Conference on Industrial Engineering and Systems Management (IESM), pp. 1–6 (2019). IEEE
38. Sun, G.-J.: Knowledge-guided neighborhood search algorithm for closeopen vehicle routing problem. In: Proceedings of the Seventh International Forum on Decision Sciences, pp. 157–163 (2020). Springer
39. Fernando, M., Thibbotuwawa, A., Perera, H.N., Ratnayake, R.C.: Closeopen mixed vehicle routing optimization model with multiple collecting centers to collect farmers' perishable produce. In: 2022 International Conference for Advancement in Technology (ICONAT), pp. 1–8 (2022). IEEE
40. Conrad, R.G., Figliozzi, M.A.: The recharging vehicle routing problem. In: Proceedings of the 2011 Industrial Engineering Research Conference, p. 8 (2011). IISE Norcross, GA
41. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transportation research part E: logistics and transportation review **48**(1), 100–114 (2012)

42. Schneider, M., Stenger, A., Goeke, D.: The electric vehicle-routing problem with time windows and recharging stations. Transportation Science **48**(4), 500–520 (2014)
43. Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G.: A heuristic approach for the green vehicle routing problem with multiple technologies and partial recharges. Transportation Research Part E: Logistics and Transportation Review **71**, 111–128 (2014)
44. Bruglieri, M., Pezzella, F., Pisacane, O., Suraci, S.: A variable neighborhood search branching for the electric vehicle routing problem with time windows. Electronic Notes in Discrete Mathematics **47**, 221–228 (2015)
45. Keskin, M., Çatay, B.: Partial recharge strategies for the electric vehicle routing problem with time windows. Transportation Research Part C: Emerging Technologies **65**, 111–127 (2016)
46. Bruglieri, M., Mancini, S., Pezzella, F., Pisacane, O., Suraci, S.: A threephase matheuristic for the time-effective electric vehicle routing problem with partial recharges. Electronic Notes in Discrete Mathematics **58**, 95–102 (2017)
47. Desaulniers, G., Errico, F., Irnich, S., Schneider, M.: Exact algorithms for electric vehicle-routing problems with time windows. Operations Research **64**(6), 1388–1405 (2016)
48. Yavuz, M.: An iterated beam search algorithm for the green vehicle routing problem. Networks **69**(3), 317–328 (2017)
49. Keskin, M., Çatay, B.: A matheuristic method for the electric vehicle routing problem with time windows and fast chargers. Computers & Operations Research **100**, 172–188 (2018)
50. Ceselli, A., Felipe, Á., Ortuño, M.T., Righini, G., Tirado, G.: A branchand-cut-and-price algorithm for the electric vehicle routing problem with multiple technologies. In: Operations Research Forum, vol. 2, pp. 1–33 (2021). Springer
51. Chakraborty, N., Mondal, A., Mondal, S.: Intelligent charge scheduling and eco-routing mechanism for electric vehicles: A multi-objective heuristic approach. Sustainable Cities and Society **69**, 102820 (2021)
52. Montoya, A., Guéret, C., Mendoza, J.E., Villegas, J.G.: The electric vehicle routing problem with nonlinear charging function. Transportation Research Part B: Methodological **103**, 87–110 (2017)
53. Lee, C.: An exact algorithm for the electric-vehicle routing problem with nonlinear charging time. Journal of the Operational Research Society **72**(7), 1461–1485 (2021)
54. Kancharla, S.R., Ramadurai, G.: Electric vehicle routing problem with non-linear charging and load-dependent discharging. Expert Systems with Applications **160**, 113714 (2020)
55. Poonthalir, G., Nadarajan, R.: Green vehicle routing problem with queues. Expert Systems with Applications **138**, 112823 (2019)
56. Keskin, M., Laporte, G., Çatay, B.: Electric vehicle routing problem with time-dependent waiting times at recharging stations. Computers & Operations Research **107**, 77–94 (2019)
57. Keskin, M., Çatay, B., Laporte, G.: A simulation-based heuristic for the electric vehicle routing problem with time windows and stochastic waiting times at recharging stations. Computers & Operations Research **125**, 105060 (2021)
58. Iwan, S., Nürnberg, M., Jedliński, M., Kijewska, K.: Efficiency of light electric vehicles in last mile deliveries-szczecin case study. Sustainable Cities and Society **74**, 103167 (2021)
59. Xiao, Y., Zhang, Y., Kaku, I., Kang, R., Pan, X.: Electric vehicle routing problem: A systematic review and a new comprehensive model with nonlinear energy recharging and consumption. Renewable and Sustainable Energy Reviews **151**, 111567 (2021)
60. Asghari, M., Al-e, S.M.J.M., et al.: Green vehicle routing problem: A state-of-the-art review. International Journal of Production Economics **231**, 107899 (2021)
61. Moghdani, R., Salimifard, K., Demir, E., Benyettou, A.: The green vehicle routing problem: A systematic literature review. Journal of Cleaner Production **279**, 123691 (2021)
62. Kar, S., Dutta, J., Barma, P.S., Mukherjee, A., De, T.: A hybrid multi-objective evolutionary algorithm for open vehicle routing problem through cluster primary-route secondary approach. International Journal of Management Science and Engineering Management, 1–15 (2022)
63. Jie, W., Yang, J., Zhang, M., Huang, Y.: The two-echelon capacitated electric vehicle routing problem with battery swapping stations: Formulation and efficient methodology. European Journal of Operational Research **272**(3), 879–904 (2019)
64. Kara, I., Kara, B.Y., Yetis, M.K.: Energy minimizing vehicle routing problem. In: International Conference on Combinatorial Optimization and Applications, pp. 62–71 (2007). Springer
65. Feo, T.A., Resende, M.G.: Greedy randomized adaptive search procedures. Journal of Global Optimization **6**(2), 109–133 (1995)
66. Mladenović, N., Hansen, P.: Variable neighborhood search. Computers & operations research **24**(11), 1097–1100 (1997)
67. Xiao, Y., Zhao, Q., Kaku, I., Mladenovic, N.: Variable neighbourhood simulated annealing algorithm for capacitated vehicle routing problems. Engineering Optimization **46**(4), 562–579 (2014)

68. Solomon, M.M.: Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations Research **35**(2), 254–265 (1987)