Iterated Local Search for the eBuses Charging Location Problem

César Loaiza Quintana[1(✉)], Laura Climent[2], and Alejandro Arbelaez[2]

[1] University College Cork, Cork, Ireland
c.loaizaquintana@cs.ucc.ie
[2] Universidad Autónoma de Madrid, Madrid, Spain
{laura.climent,alejandro.arbelaez}@uam.es

Abstract. Electric buses (eBuses) will be the mainstream in mass urban transportation in the near future. Thus, installing the charging infrastructure in convenient locations will play a critical role in the transition to eBuses. Taking this into account, in this paper we propose an iterated local search algorithm to optimize the location of charging stations while satisfying certain properties of the transportation system, e.g., satisfying the demand and ensuring that the limited driving range of the buses will not impact the service. The effectiveness of our approach is demonstrated by experimenting with a set of problem instances with real data from 3 Irish cities, i.e., Limerick, Cork, and Dublin. We compare our approach against a MIP-based solution. Results show that our approach is superior in terms of scalability and its anytime behavior.

Keywords: Local search · Charging location problem · Electric buses

1 Introduction

The transition to a fleet of eBuses will require a considerable up-front investment in the charging infrastructure. The actual cost of a charging station varies depending on the desired charging infrastructure. The starting cost is now estimated at €10K with additional fees proportional to the required power, e.g., €10K for a fast charging point with approximately 600 kW capacity as well as substantial upgrades and cabling costs that can certainly increase the total cost [9]. The daily power consumption of an eBus depends on multiple factors, such as bus type, weight, weather, and road conditions. A standard double-decker working 12 h daily consumes approx. 1.5 kWh per kilometer. However, the power consumption might increase or decrease due to multiple factors.

Range anxiety is one of the main concerns for transitioning to a clean and sustainable bus transportation network. Nowadays, eBuses can travel approx. up to 200–300 KM on a full charge and the charging time varies depending on the technology from a couple of minutes (e.g., with fast-charging stations) to hours (e.g., with slow overnight charging). Therefore, the charging infrastructure must be implemented in a way that the charging time and limited driving range will not impact the quality of service of the transportation system. In this paper,

G. Rudolph et al. (Eds.): PPSN 2022, LNCS 13399, pp. 338–351, 2022.
https://doi.org/10.1007/978-3-031-14721-0_24

we propose an iterated local search to efficiently tackle the charging location problem, i.e., finding the optimal location of charging stations while satisfying certain properties of the transportation system.

This paper is organized as follows: Sect. 2 describes previous work on the charging location problem for electric vehicles. Section 3 formally describes the charging location problem. Section 4 presents our new meta-heuristic solution. Section 5 presents our empirical evaluation. Finally, Sect. 6 presents some concluding remarks.

2 Related Work

A considerable amount of work has been devoted to the charging location problem for light-duty vehicles (e.g., personal use cars) rather than heavy-duty vehicles (e.g., buses). We remark that public buses operate daily schedules for a given set of routes and buses consume more energy than cars. In this line, in [4,5,14] the authors propose a set of heuristic solutions to identify suitable charging stations for electric vehicles, so that the vehicles can reach their destinations without running out of power. [15] takes into account a cost model for the charging location problem by considering different technologies, bio-diesel, bio-gas, and electric to assign a type of bus to each route.

In another line of work, the Green Vehicle Routing Problem (Green VRP) is an extension of the traditional VRP [3]. This extension aims at finding the optimal routes traveled by the fleet of vehicles to visit a set of customers while taking into account certain refueling constraints. We recall that we aim at transitioning to eBuses while maintaining the same service. [2] outlines numerous variations of the VRP for electric vehicles.

Closer to our research, [12] considered a fleet of vehicles, each one with well-defined routes within a network. This work proposes a MIP model whose objective was to minimize the cost of the charging stations, which had to be located into a subset of the stops. As the author suggests this work could be adapted to a bus transportation system by considering timetables, however, he neglects them in his validation carried out over random-generated networks.

[13] proposes a MIP model for the eBus charging location problem with fixed routes but without taking into consideration certain operational constraints, e.g., fixed timetables or overlapping constraints to prevent multiple buses using the same charging unit at the same time. In [10] the authors take into account the impact of timetables of the original routes by allowing charging when the buses are not moving (dwelling times), e.g., during the rest time of the drivers, therefore, the authors keep the original timetables. Instead, in this paper, we propose a more flexible approach to allow small enough deviations in the original timetable leading to a more general version of the problem. [1] provides more details of the described behavior of the charging location problem.

3 The Charging Location Problem

Conceived in the context of a transition to eBuses, the charging location problem aims to identify, within the set of stations, the locations for installing the minimum number of electrical chargers. These electrical chargers must allow a fleet of eBuses to keep the operation of the system with minimal disruptions (deviations) from the original timetables. In this context, we assume that the charging stations are able to recharge the buses during their normal operation, i.e., in the time between the arrival and departure of the buses in the stations.

Figure 1 outlines an instance of the system for a single bus. In this example, we assume that the bus visits 12 stations. For each station we have three timestamps, i.e., actual arrival time (left), expected arrival time (middle), and departure time (right). In order to reduce the disruptions in the transition from regular diesel buses to eBuses, we fix a maximum deviation time μ from the original timetables (2 min in this example). Therefore, eBuses are not allowed to arrive more than μ minutes after (or before) the timetabled stop. We also associate each trip between two stations with the required time and power to complete the trip. The number of points sitting below each charging station denotes the amount of power (in kWh) gained at the charging station (yellow stations in this example), e.g., in station 5 the bus gains 2 kWh units of power.

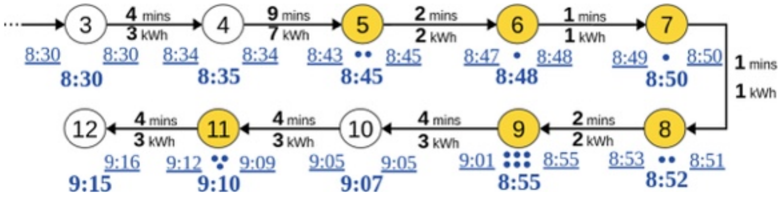


Fig. 1. Example of a solution to the problem for a single bus

Hereafter, we formally describe the charging location problem. Table 1 outlines terms and variables commonly used through the paper.

A public bus transportation system is composed of a set of buses $\mathcal{B}$ traveling across a set of stations ST, where each bus follows a sequence of ordered stops S_b visited in a workday. Thus, each stop $s_{bi} \in S_b$ has associated a station and a timetable that the bus should comply with (st_{bi} and T_{bi}).

We build our formulation of the charging location problem on top of a previous MIP model [1]. The objective function (1) is to minimize the number of installed chargers. Constraints 2–5 ensure buses are not running out of energy during their trips, by maintaining the battery within safe levels. Furthermore, Constraints 6–9 are in charge of the scheduling of the buses, which must stay within a reasonable deviation time Δt of μ with respect to the original timetables. Constraint 10 enforces a minimum reasonable charging time.

Table 1. Description of variables and constants.

Constants	
$\mathcal{B}, ST, SM$	set of buses, set of stations, and security margin between chargers
$\mathcal{S}_b$	sequence of ordered stops visited by bus b in a workday, $\mathcal{S}_b = \{s_{b1}, \dots, s_{bn}\}$, where n is the number of stops
st_{bi}	i-th station in the path of a bus b in a workday
τ_{bj}	timetabled arrival time of the bus b at its j-th stop
μ	max. time disruption allowed (considering the scheduled times)
T_{ji}, D_{ji}	time and energy needed to complete the trip between stations j and i
C_{min}, C_{max}	min. and max. capacity battery levels of the buses
ψ, μ, β, α	min. charging time, max. deviation time from original timetable, max. charging time per cycle, and charging rate in kWh per minute.
Variables	
t_{bj}	actual arrival time of bus b at the arrival to its j-th stop
Δt_{bj}	time difference between the arrival and scheduled times
c_{bj}	current battery level of bus b at the arrival to its j-th stop
e_{bj}	energy re-charged of bus b at its j-th stop
ct_{bj}	recharging time of bus b at its j-th stop
x_{bj}	Boolean var. denoting whether we recharge bus b at its j-th stop
x_i	Boolean var. denoting whether we install a charging unit at station i
Z_{bdij}	Boolean var. denoting whether buses b and d are using the same charging station ($i = j$) or not
z_{bdi}	Boolean var. indicating if bus b charges after the bus d at station i

$$min \sum_{c \in ST} x_c \tag{1}$$

$$C_{min} \leq c_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{2}$$

$$c_{bi} + e_{bi} \leq C_{max} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{3}$$

$$c_{bi} \leq c_{bj} + e_{bj} - D_{ji} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}, j = i-1} \tag{4}$$

$$\alpha \cdot ct_{bi} \geq e_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}, j = i-1} \tag{5}$$

$$t_{bi} \geq t_{bj} + ct_{bj} - T_{ji} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}, j = i-1} \tag{6}$$

$$\Delta t_{bi} \geq t_{bi} - \tau_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{7}$$

$$\Delta t_{bi} \geq \tau_{bi} - t_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{8}$$

$$\Delta t_{bi} \leq \mu \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{9}$$

$$ct_{bi} \geq \psi \cdot x_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{10}$$

$$x_{st_{bi}} \geq x_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{11}$$

$$\beta \cdot x_{bi} \geq ct_{bi} \qquad \forall_{b \in B} \forall_{s_i \in \mathcal{S}_b \setminus \{s_0\}} \tag{12}$$

$$Z_{bdij} \leq x_{bi} \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{13}$$

$$Z_{bdij} \leq x_{dj} \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{14}$$

$$x_{bi} + x_{dj} \leq Z_{bdij} + 1 \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{15}$$

$$t_{bi} \geq t_{dj} + ct_{dj} - Mz_{bdi} \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{16}$$

$$t_{dj} \geq t_{bi} + ct_{bi} - Mz_{dbj} \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{17}$$

$$z_{bdi} + z_{dbj} - (1 - Z_{bdij}) \leq 1 \qquad \forall_{b,d \in B | b \neq d} \forall_{s_i \in \mathcal{S}_b}, \forall_{s_j \in \mathcal{S}_d | st_{bi} = st} \tag{18}$$

Constraint 11 determines the selected charging stations. Constraint 12 prevents overheating by limiting charging time. Finally, Constraints 13–18 prevent multiple buses from using the same charger at the same time. We refer the reader to [1] for more details of the MIP formulation.

4 The Iterated Local Search

Iterated Local Search (ILS) is a popular meta-heursitic technique to solve complex combinatorial optimization problems [7,11]. Generally speaking, the algorithm produces a single-transformation chain of solutions that iteratively improves the objective function. First, in a local search phase, the algorithm tries to improve an initial solution by performing small changes, until a local minimum ($s*$) is reached. Then, in a perturbation phase, the algorithm performs random changes to the current incumbent solution ($s*$) to escape difficult regions of the search (e.g., a plateau), and to produce a perturbed solution (s'). Next, the perturbed solution (s') is given as an input for a new local search phase that generates a new local minimum (s''). Finally, an acceptance criterion decides which will be the new incumbent solution ($s*$) to continually repeat the process from the perturbation phase. The idea is that the algorithm will primarily prefer the best solution between $s*$ and s''. However, in order to diversify the search, the algorithm selects the most recent local minimum (s'') with a certain probability. In this paper, unless stated otherwise, we use a 5% probability to update the incumbent solution.

In the context of the charging location problem, we define an open station as a location (i.e., bus station) with an installed charging unit iff at least one bus is relying on the station to recharge the battery. Therefore, during the local search phase, we aim at closing as many open stations as possible. Alternatively, during the perturbation phase, we aim at diversifying the search by randomly opening stations.

Initial Solution

We propose a simple greedy algorithm to compute an initial solution. We assume that the fleet of eBuses start operations with full capacity. The algorithm commands a given bus b to recharge the battery at a given stop $s_{bj} \in S_b$ if there is not enough power to reach the next stop s_{bj+1}. Furthermore, b will recharge the battery of the bus with, at least, the minimum required power to reach s_{bj+1}. We note that considering that b starts with full capacity, all the stations on the path of the bus might be open once the bus depletes its initial charge.

Furthermore, if b is expected to arrive at s_{bj+1} before the allowed scheduled time (see Constraints 7–9), then b uses this additional time (dwell time) to recharge the battery and get additional power. The algorithm repeats this process for each bus $b \in \mathcal{B}$ in lexicographical order. Additionally, whenever we get an overlapping conflict (i.e., two buses are attempting to use the charging station at the same time, see Constraints 13–18), we delay the arrival time of the current bus until the charging station is available.

Let us illustrate the process to compute the initial solution with Fig. 1. We assume that the initial capacity of the bus is enough to reach station 5, at this point the bus needs to recharge the battery with at least 2 kWh; otherwise, the eBus runs out of battery before reaching station 6. Therefore, the bus arrives at station 5 at 8:43 and recharges the battery for 2 min to leave the station at 8:45. Then, the bus arrives at station 6 at 8:47 to recharge the battery again for 1 min. The algorithm repeats the same process until arriving at station 9 at 8:55. Here, our greedy algorithm initially recharges the bus for 3 min (enough to reach station 10), however, by doing so, the bus would arrive at the next station at 9:02, 3 min ahead of the earliest allowed time. We recall the bus is constrained to arrive at station 10 at 9:05. This additional time is known as dwell. Therefore, our greedy construction of the initial solution uses this time to provide extra power (6 kWh). Consequently, station 10 remains closed as the bus does not need to use the station to reach station 11.

Local Search

The main purpose of the local search (Algorithm 1), given a solution s with a set of opened stations (*stations* - line 1), is to close as many as it can. Thus, the algorithm repeatedly attempts close open stations in a certain order. The selection of an open station (*SelectStation*), which is a function itself, is passed as a parameter. In particular, we explore two different selection strategies: *Random* selects a station uniformly at random and *MinActivity* selects the station with the fewest number of scheduled recharges.

Lines 4–15 form the core of the local search algorithm. Lines 5–8 select and attempt to close the station. Line 9 removes the selected station from the candidate set. After exhausting the list of candidate open stations we verify whether the algorithm reached a local minimum (Lines 10–12). Otherwise, we repeat the process with the remaining open stations in the solution. Let us recall that stations that could not be closed at a certain point of the execution may become closable after modifying the schedule of nearby stations.

The amount of charged energy remains constant during the whole process, and it is set to the minimum required amount. Therefore, when the algorithm closes or opens a charging station, it redistributes or borrows energy to/from the neighboring stations. The proportion of energy going/coming to/from previous and following stations is random.

The *CloseStation* operator (Algorithm 2) attempts to stop all charging activity of the buses in a given installed charging station. Therefore, in order to ensure the operation of the system, we redistribute the energy provided by the closing station towards adjacent open stations. We refer to these adjacent stations as *previous* and *next* open charging stations.

Algorithm 1. LocalSearch(s, *SelectStation*)

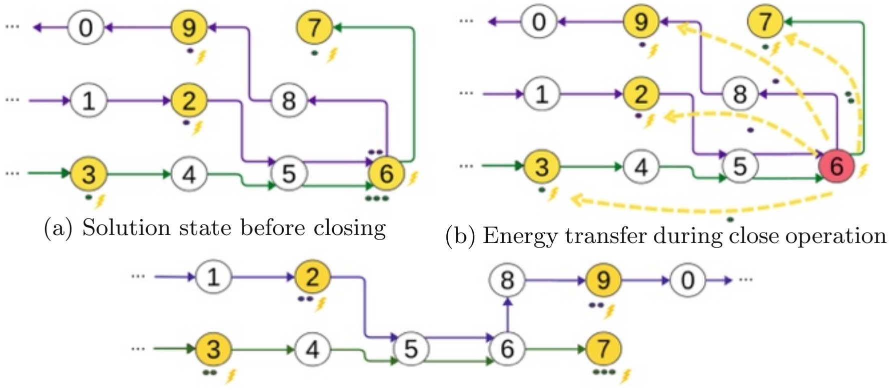
```
$stations \leftarrow \{c | c \in ST \wedge x_c = 1\}$
$bestObj, obj \leftarrow |stations|$
$isLocalMin \leftarrow false$
while not isLocalMin do
    $c \leftarrow SelectStation(stations)$
    if $CloseStation(c)$ then
        $obj \leftarrow obj - 1$
        $x_c \leftarrow 0$
    $stations \leftarrow stations - \{c\}$
    if $|stations| = 0$ then
        if $obj \geq bestObj$ then
            $isLocalMin \leftarrow true$
        else
            $bestObj \leftarrow obj$
            $stations \leftarrow \{c | c \in ST \wedge x_c = 1\}$
```

For instance, Fig. 2 represents two different bus routes (the blue and the green one). Both routes are sharing two locations but only one charging station in location 6. Thus, the charging stations are highlighted in yellow. Furthermore, the small points near the charging stations depict units of energy delivered to each bus. Figure 2a shows the green bus recharging three units of energy at charging station 6 while the blue bus gets one unit of energy at charging station 2. In Fig. 2b, when closing the charging station 6, the three units of energy provided by the charging station 6 to the green bus are redistributed as follows: two-thirds to the next station and one-third to the previous station where the bus was recharging in the original solution. This proportion is decided uniformly at randomly for each stop. In this particular example, the energy is evenly reallocated for the blue bus. Figure 2c outlines the resulting state of the solution after completing the close operation.

Algorithm 2. *CloseStation*(selectedStation)

```
$openStopsPerStation \leftarrow \{s_{bi} | \forall_{b \in B} \forall_{s_{bi} \in S_b}, st_{bi} = selectedStation \wedge x_{bi} = 1\}$
for $s_{bi}$ in openStopsPerStation do
    if $FeasibleCloseStop(s_{bi})$ then $CloseStop(s_{bi})$ else return false
return true
```

Algorithm 2 (*CloseStation*) checks whether a charging station can be closed or not under certain circumstances and apply the operation. To this end, *FeasibleCloseStop* verifies if a given bus b can keep operating once the power gotten from recharging at s_{bi} is moved to the *Previous* and the *Next* open stations in the path of b. *CloseStop* completes the close operations and updates the arrival times, charging times, and the capacity of the batteries of the buses as prescribed



(a) Solution state before closing

(b) Energy transfer during close operation

(c) Solution state after closing

Fig. 2. Example of the close operation applied to station 6 (Color figure online)

by the *FeasibleCloseStop* algorithm. Taking this into account, *CloseStation* indicates whether the operation is successful for the entire fleet of buses relying on the input charging station, i.e., the system is capable of redistributing the power provided by the station without violating the operational constraints. We remark that the close operation might be performed partially as once the algorithm finds a non-closable stop, it might have already closed multiple stops for certain buses. We decided not to undo partial closes as it increases the complexity of the operation.

FeasibileCloseStop (Algorithm 3) verifies if the power provided by the m-th station in the path of b can be redistributed to adjacent stations. To this end, Line 1 calculates an ordered set with the open stations in the path of b. Line 2 calculates ρ the proportion of energy that goes to the next open stop (1-ρ goes to the previous one). Lines 3 and 4 define some variables that verify the feasibility of the constraints. Lines 5 and 6 calculate the index of the next (n) and previous (k) charging stations in the path of the bus. We remark that the next and previous charging stations are not always the same as we constantly open and close stations.

Lines 7–15 verify the feasibility of re-distributing energy to n. ne denotes the actual amount of power that b gains at n. nct denotes the additional time that the bus needs to gain the extra power at n. Lines 10–11 check whether the current capacity of the battery is enough to reach n and the new charging time is within the limits. Lines 12–15 check the overlapping constraints and prevent multiple buses of using the same charging station at the same time. We mark that in this scenario there is no need to check for potential time delays as the bus arrives at n earlier than expected.

Lines 16–26 verify the feasibility of re-distributing energy backwards to k. Whereas the risk of running out of power is inexistent the bus could experiment delays in the timetable. The new extra charging time at k could potential delay

Algorithm 3. $FeasibleCloseStop(selectedStop_{bm})$

```
1: $openStops_b \leftarrow \{s_{bi} | \forall_{s_{bi} \in S_b}, x_{bi} = 1\}$
2: $\rho \leftarrow Random(0, 1)$
3: $notOverflow, notDelay, notDeplete \leftarrow true$
4: $notExtraPCt, notExtraNCt, notOverlap \leftarrow true$
5: $n \leftarrow NextIndex(openStops_b, selectedStop_{bm})$
6: $k \leftarrow PreviousIndex(openStops_b, selectedStop_{bm})$
7: if $n$ is not nil then
8:     $ne \leftarrow$ if $k$ is not nil then $(1 - \rho) \cdot e_{bm}$ else $e_{bm}$
9:     $nct \leftarrow \frac{ne}{\alpha}$
10:    $notDeplete \leftarrow c_{bn} - ne \geq C_{min}$
11:    $notExtraNCt \leftarrow ct_{bn} + nct \leq \beta$
12:    for d in $\mathcal{B}$ $|d \neq b$ do
13:        for $s_{di}$ in $S_d$ $|st_{bi} = st_{bn} \wedge x_{di} = 1$ do
14:            $notOverlapTmp \leftarrow t_{bn} + ct_{bn} + nct \leq t_{di} \vee t_{bn} + ct_{bn} + nct \geq t_{di} + ct_{di}$
15:            $notOverlap \leftarrow notOverlap \wedge notOverlapTmp$
16: if $k$ is not nil then
17:    $pe \leftarrow$ if $n$ is not nil then $\rho \cdot e_{bm}$ else $e_{bm}$
18:    $pct \leftarrow \frac{pe}{\alpha}$
19:    $max\Delta t_{bm} \leftarrow max(\{\Delta t_{bk+1} ... \Delta t_{bm}\})$
20:    $notDelay \leftarrow \mu \geq max\Delta t_{bm} + pct$
21:    $notOverflow \leftarrow c_{bk} + e_{bk} + pe \leq C_{max}$
22:    $notExtraPCt \leftarrow ct_{bk} + pct \leq \beta$
23:    for d in $\mathcal{B}$ $|d \neq b$ do
24:        for $s_{di}$ in $S_d$ $|st_{bi} = st_{bk} \wedge x_{di} = 1$ do
25:            $notOverlapTmp \leftarrow t_{bk} + ct_{bk} + pct \leq t_{di} \vee t_{bk} + ct_{bk} + pct \geq t_{di} + ct_{di}$
26:            $notOverlap \leftarrow notOverlap \wedge notOverlapTmp$
27: $notExtraCt \leftarrow notExtraPCt \wedge notExtraNCt$
28: $isFeasible \leftarrow notExtraCt \wedge notOverflow \wedge notDelay \wedge notDeplete \wedge notOverlap$
29: return isFeasible
```

all the arrival times of the bus in the stations between the closing station and the previous charger. Let us define *pct* and $max\Delta_{bi}$ (Lines 18 and 19). *pct* denotes the additional time that the bus needs to gain the power at k and $max\Delta_{bi}$ denotes the maximum delay a bus is currently experiencing during the trip between the previous station and the closing one. Line 20 ensures that the extra time spent in the previous charger is not disrupting the timetables beyond the threshold μ. Besides checking a battery overflow in Line 21, there is no need to further inspect that the energy is within the limits. Line 22 checks that the charging time is within the limit. Lines 23–26 check for overlapping charging events.

Let us illustrate the verification process with Fig. 2, and let us assume that the bus recharges one unit of power per minute. Firstly, we recall that the close operation acts on a single bus stop at a time. Thus, the sequence of open stops of the green bus (Algorithm 3 - line 1) includes the stops of the bus at stations 3, 6, and 7. Furthermore, since the selected stop corresponds to the stop at station

6 in that new sequence, the previous and next functions return the stops at stations 3 and 7 respectively.

Now, let us consider the first case to transfer energy towards the previous open stop at stop 3. In the current solution (before the operation) the bus departs from station 3 with at least 1 unit of power. After the close operation, the bus departs with an additional unit of power (we depict this behavior in Fig. 2c). Thus, it is impossible to run out of power during the trip between stations 3 and 6. On the other hand, in the new solution, the bus is spending more time recharging at station 3. This extra time introduces a delay when arriving at station 4, 5, and 6. Therefore, it is paramount to check that the delay of a minute is not violating the maximum allowed disruption time (in the timetable) in any of those stops (checking the most delayed timetable is enough).

Similarly, it can be seen graphically that the amount of power available for traveling between stations 6 and 7 is reduced after applying the operation. Therefore, we must ensure that the trip can be completed. Lastly, since the bus is saving two recharging minutes before departing from station 6, there is no way the bus is delayed when arriving at 7.

Perturbation

The perturbation phase aims at escaping from difficult regions (i.e., local minimum) while maintaining an appropriate balance between diversification and intensification. The operator iteratively tries to open a given station *st* by attempting to recharge the battery of certain buses while stopping at the station. To this end, we randomly redistribute the gained power at adjacent charging stations of *st* (i.e., *Next* and *Previous*). Thus, we label *st* as open iff at least one bus relies on the station to recharge the bus. We attempt to open an additional percentage of the opened stations of the incumbent solution, which determines the level of perturbation.

Let us illustrate the process of opening a station with Fig. 2 by looking at the example the other way around. Then Fig. 2c denotes the incumbent solution and attempting to open station 6. Furthermore, let us assume that the yellow arrows in Fig. 2b are pointing in the opposite direction. In this context, the blue bus will attempt to transfer a random portion of the gained power at stations 9 and 2 into station 6. Therefore, in order to open station 6 (blue bus), it is important to verify that the redistribution of the power lost at station 2 leaves the bus with enough power to complete the trip between stations 2 and 6 and that any potential delay (derived from charging at station 6) is not impacting the quality of service.

5 Evaluation

In this paper we use a real dataset with the operations of three Irish cities, i.e., Limerick, Cork, and Dublin.[1] We implemented the MIP model with CPLEX

[1] The GPS location of the bus stations and timetables are available at https://buseireann.ie/ and http://www.dublinbus.ie/.

12.10 and conducted our experiments on a 2.5 GHz Intel Xeon W-2175 processor with 64 GB of memory running Ubuntu 18.04.5. We used CPLEX with its default parameters, including the parallel optimizer with 12 threads and two time limits (10 and 120 min). Furthermore, we executed our ILS algorithm on each instance 10 times (each time with a different random seed) and reported the median number of open stations (out of the 10 executions) with a 10-minute timeout. All series of experiments had a standard deviation of less than 1.

Table 2 outlines the performance of our ILS algorithm with two selection strategies (i.e., *Random* and *MinActivity*). We recall that *MinActivity* attempts to close the open station with the fewest number of charging events. The first column displays the number of buses and the total number of bus stops per city. We simulate multiple scenarios varying the C_{max} value and timetable disruptions (i.e., max μ). Moreover, in all our experiments, we establish that the charging rate is 1 kWh per minute (i.e., $\alpha = 1$), and the buses consume 1 kWh per km. Additionally, we limit the charging time per cycle of the buses (β) to the time needed to recharge the batteries to up to 80% of the max. capacity. Furthermore, we set the level of perturbation to 20% after trying other values (10% and 40%) that did not suggest a significant difference. Bold numbers indicate the method was able to find the optimal solution, whereas the highlighted cells point out the method with the best performance. Lastly, it is worth noticing that we generate initial solutions for Cork and Limerick with up to 455 charging stations, whereas for Dublin we need 1666 stations. Also, the number of local minima found during an execution varies in the order of tens of thousands for Dublin, hundreds of thousands for Cork, and millions for Limerick.

In these experiments we observe that our ILS algorithm finds the optimal solution for small-size (i.e., Limerick) and mid-size instances (i.e., Cork) with C_{max} = 180 kWh and 200 kWh. Notably, ILS dominates the performance for the remaining instances with at least the same performance as CPLEX with 12 times more computational resources.

Furthermore, it is remarkable that CPLEX is unable to find reasonable solutions, within 10 min, for 6 instances (out of 10) for our Cork and Dublin datasets. Interestingly, CPLEX with 120 min recommends the installation of more than two hundred of charging units for one of the most constrained instances (i.e., Cork with 120 kWh and max μ = 4), whereas our ILS algorithm quickly computes a solution with only 6 charging units, Similarly, our ILS algorithm finds a solution with approx. 40% fewer stations (15 vs. 28 charging stations) for the most constrained Dublin dataset, i.e., 200 kWh with max μ = 6. Finally, in general our *MinActivity* selection strategy performs slightly better than *Random*. We attribute this to the fact that this heuristic favors stations where the close operator is more likely to succeed.

Table 2. Empirical evaluation for Limerick, Cork, and Dublin. Bold numbers denote the optimal solution and highlighted cells outline the best performance.

		μ	CPLEX		ILS (10 mins)	
City	$C_{\max}$	(mins)	10 mins	120 mins	Random	MinActivity
Limerick 23 buses 8,417 stops	120	4	5	**4**	**4**	**4**
	150		**4**	**4**	**4**	**4**
	180		**2**	**2**	**2**	**2**
	200		**2**	**2**	**2**	**2**
	120	6	**4**	**4**	**4**	**4**
	150		**4**	**4**	**4**	**4**
	180		**2**	**2**	**2**	**2**
	200		**2**	**2**	**2**	**2**
Cork 65 buses 29,724 stops	120	4	565	220	7	6
	150		-	5	5	4
	180		5	**3**	**3**	**3**
	200		**3**	**3**	**3**	**3**
	120	6	-	8	7	6
	150		567	6	5	4
	180		4	3	3	3
	200		**3**	**3**	**3**	**3**
Dublin 163 buses 36,609 stops	200	6	2636	28	17	15
	200	10	-	17	17	15

6 Conclusions

In this paper we have proposed an efficient ILS algorithm to tackle the Charging Location Problem for eBuses. Our proposed algorithm relies on two simple operations, i.e., opening and closing charging stations in a given solution. Furthermore, our approach assumes that the eBuses recharge a constant amount of energy during a workday, therefore, the energy is locally distributed (e.g., when closing a station), among the adjacent stations of the ones affected by the operators. The effectiveness is demonstrated by experimenting with a set of real instances from three Irish cities (i.e., Limerick, Cork, and Dublin). We compared our algorithm against a MIP-based solution and our ILS solution is notably better than CPLEX in terms of the number of installed charging stations, generating results with up to 40% fewer stations in less than 10% of the time for some of the most difficult instances.

In the future, we plan to extend our ILS with robustness to tackle the Charging Location Problem, so that the system is resilient to failures in the transportation network. Also, we plan to perform an extensive parameter tuning using tools like ParamILS [8] or Calibra [6].

Acknowledgements. This work received funding from the Sustainable Energy Authority of Ireland (SEAI) under the RDD 2019 programme - Grant No 19/RDD/519. The authors would like to thank the anonymous reviewers for their comments and suggestions which helped to improve the paper.

References

1. Arbelaez, A., Climent, L.: Transition to eBuses with minimal timetable disruptions. In: International Symposium on Combinatorial Search (SoCS). Association for the Advancement of Artificial Intelligence (2020)
2. Erdelić, T., Carić, T.: A survey on the electric vehicle routing problem: variants and solution approaches. J. Adv. Transp. **2019**(54), 1–48 (2019)
3. Erdoğan, S., Miller-Hooks, E.: A green vehicle routing problem. Transp. Res. Part E Logistics Transp. Rev. **48**(1), 100–114 (2012)
4. Funke, S., Nusser, A., Storandt, S.: Placement of loading stations for electric vehicles: no detours necessary! In: AAAI, pp. 417–423. AAAI Press (2014)
5. Funke, S., Nusser, A., Storandt, S.: Placement of loading stations for electric vehicles: allowing small detours. In: ICAPS 2016, pp. 131–139. AAAI Press (2016)
6. Hoos, H.H.: Automated algorithm configuration and parameter tuning. In: Hamadi, Y., Monfroy, E., Saubion, F. (eds.) Autonomous Search, pp. 37–71. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21434-9_3
7. Hoos, H.H., Stützle, T.: Stochastic Local Search: Foundations and Applications. Elsevier, Amsterdam (2004)
8. Hutter, F., Hoos, H.H., Leyton-Brown, K., Stützle, T.: ParamILS: an automatic algorithm configuration framework. J. Artif. Intell. Res. **36**, 267–306 (2009)
9. Karlsson, E.: Charging infrastructure for electric city buses: an analysis of grid impact and costs (2016)
10. Kunith, A., Mendelevitch, R., Goehlich, D.: Electrification of a city bus network-an optimization model for cost-effective placing of charging infrastructure and battery sizing of fast-charging electric bus systems. Int. J. Sustain. Transp. **11**(10), 707–720 (2017)
11. Stützle, T., Ruiz, R.: Iterated local search. In: Martí, R., Pardalos, P.M., Resende, M.G.C. (eds.) Handbook of Heuristics, pp. 579–605. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-07124-4_8
12. Wang, I.L., Wang, Y., Lin, P.C.: Optimal recharging strategies for electric vehicle fleets with duration constraints. Transp. Res. Part C Emerg. Technol. **69**, 242–254 (2016)
13. Wang, X., Yuen, C., Hassan, N.U., An, N., Wu, W.: Electric vehicle charging station placement for urban public bus systems. IEEE Trans. Intell. Transp. Syst. **18**(1), 128–139 (2016)
14. Wang, Y.W., Lin, C.C.: Locating road-vehicle refueling stations. Transp. Res. Part E Logistics Transp. Rev. **45**(5), 821–829 (2009)
15. Xylia, M., Leduc, S., Patrizio, P., Kraxner, F., Silveira, S.: Locating charging infrastructure for electric buses in Stockholm. Transp. Res. Part C Emerg. Technol. **78**, 183–200 (2017)