Computers & Operations Research ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Computers & Operations Research

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

Battery swap station location-routing problem with capacitated electric vehicles

Jun Yang [a], Hao Sun [a,b,*]

[a] School of Management, Huazhong University of Science and Technology, Wuhan 430074, China
[b] School of Economics and Management, Tsinghua University, Beijing 100086, China

ARTICLE INFO

Keywords:
Electric vehicles
Battery swapping
Location-routing problem
Adaptive large neighborhood search

ABSTRACT

In this paper, we present an electric vehicles battery swap stations location routing problem (BSS–EV–LRP), which aims to determine the location strategy of battery swap stations (BSSs) and the routing plan of a fleet of electric vehicles (EVs) simultaneously under battery driving range limitation. The problem is formulated as an integer programming model under the basic and extended scenarios. A four-phase heuristic called SIGALNS and a two-phase Tabu Search-modified Clarke and Wright Savings heuristic (TS-MCWS) are proposed to solve the problem. In the proposed SIGALNS, the BSSs location stage and the vehicle routing stage are alternated iteratively, which considers the information from the routing plan while improving the location strategy. In the first phase, an initial routing plan is generated with a modified sweep algorithm, leading to the BSSs location subproblem, which is then solved by using an iterated greedy heuristic. In the third phase, the vehicle routes resulting from the location subproblem are determined by applying an adaptive large neighborhood search heuristic with several new neighborhood structures. At the end of SIGALNS, the solution is further improved by a split procedure. Compared with the MIP solver of CPLEX and TS-MCWS over three sets of instances, SIGALNS searches the solution space more efficiently, thus producing good solutions without excessive computation on the medium and large instances. Furthermore, we systematically conduct economic and environmental analysis including the comparison between basic and extended scenarios, sensitivity analysis on battery driving range and efficiency analysis about the vehicle emissions reduction when EVs are used in the logistics practice.

1. Introduction

Recognizing the environment dimensions (such as carbon emissions and demand on energy), logistics companies have special responsibility to make their operations greener. Many logistics companies are looking for new approaches to manage carbon effectively from procurement and production to distribution and product afterlife. For instance, the logistics giant DHL announced a new pilot project which would introduce electrified delivery vans for its vehicle fleet in US and Germany [45,19]. DHL will order at least 50 Renault EVs by 2015 [5]. In 2013, UPS had more than 100 electric vehicles (EVs) in its fleet operating in US [6]. However, there are several major fundamental barriers to overcome for broadening adoption of EVs. The maximum driving range may be insufficient to complete some delivery tours. A major and promising solution is to remove the existing battery that is nearly depleted and replace the battery with a fully charged one [34]. Such a method of refueling is called *battery swapping* in this study. Battery swapping does have its advantages for the logistics company. Because logistics company owns the batteries it swaps out, the sticker price of EVs that can use its network is cheaper. The main benefit associated with the swapping model is speed. The whole operation could take less than 10 min, which is fast refueling on par with conventional vehicles and much faster than even the fastest recharging stations [12,26]. Moreover, another advantage of battery swapping over fast charging or recharging is that charging the depleted batteries can be left for the night at a discounted electricity price [55]. Therefore, a battery swapping model is more appropriate than a battery recharging model since the former not only improves the productivity of vehicles but also lowers the charging cost. Due to the battery driving range limitation and the nature of battery swapping, distribution network optimization with battery swapping infrastructure could be an important part of establishing green logistics. In addition, because of the lack of unified battery standards for various EVs and the insufficiency of the public supporting infrastructure, the best

* Corresponding author at: School of Management, Huazhong University of Science and Technology, Wuhan 430074, China.
E-mail addresses: jun_yang@hust.edu.cn (J. Yang), sunh3.13@sem.tsinghua.edu.cn (H. Sun).

http://dx.doi.org/10.1016/j.cor.2014.07.003
0305-0548/

battery swapping infrastructure ownership model is the company-owned business model, which indicates that the logistics companies establish and operate the battery swap stations (BSSs) for the EVs by themselves. Hence, determining the ideal BSS location strategy and vehicle routing plan for a distribution network is mainly a question of service level and operational cost for the logistics enterprises.

The location-routing problem (LRP) integrates two decision levels with strategic level (location) and tactical level (routing). Since Perl's formulation of the LRP model [43], many variants and extensions of the LRP have been addressed in the literature (see Section 2). The most studied LRP variants focus on determining the optimal number and location of depots simultaneously with finding the optimal distribution tours. In this paper, we introduce the location-routing problem for electric vehicles distribution network (BSS–EV–LRP) with a single known depot, which incorporates the location decision of BSSs with vehicle routing and battery swapping plan. The BSS–EV–LRP considers a limited driving range and loading capacity of the EVs. The goal of the problem is to determine: (a) the location of the BSSs, (b) the allocation of customers to EVs, (c) the allocation of EVs to BSSs, and (d) the tours from the depot to serve customers with BSSs' service considering the driving range and the capacity limitation of EVs.

The remainder of the paper is organized as follows. In Section 2, the existing related literature is reviewed. Section 3 establishes a mathematical formulation of the problem (BSS–EV–LRP). Two heuristics are proposed for the solutions of medium- and large-size instances in Section 4. Computational results are conducted to assess heuristics performance in Section 5, followed by an economic and environmental analysis in Section 6. Some conclusions and future research are discussed in Section 7.

2. Literature review

There are two streams of research directly related to our paper. The first stream investigates location-routing problems (LRP). Generally speaking, facility location and vehicle routing appear to be two interrelated decisions in many applications. The LRP integrates these two decision levels with the objective of solving location and routing problems simultaneously. The variants of the LRP were commonly studied in recent literature according to considerations on single or multiple depots [32,59], capacities on depots or on vehicles [36,35], time windows [60,61], fixed cost of vehicles or uncertainty of some parameters [11,3]. Most of early related papers were shown in a survey [38]. More recently, Nagy and Salhi [41] presented a comprehensive study on the LRP work. Due to the problem's complexity, exact algorithms have been proposed to solve medium-sized and basic uncapacited or capacited vehicles instances. The first exact method was a branch and bound algorithm proposed by Laport and Norbert [30]. Laport et al. [31] developed a branch and cut algorithm for uncapacitated LRP and reported computational results for instances with 20 customers and 8 depots. Belenguer et al. [8] described a branch-and-cut algorithm for the capacitated LRP based on a zero-one linear model strengthened by new families of valid inequalities and solved instances with 5–10 potential depots and 20–88 customers. Baldacci et al. [7] described a new exact method for solving the LRP based on a set-partitioning-like formulation of the problem. Regarding possible solutions, many authors have developed heuristic methods for LRP family. Most of the heuristics divided the decision levels into subproblems and solved them sequentially or iteratively. A two-phase tabu search was proposed by Albareda-Sambola et al. [2] for a combined location-routing problem with uncapacitated vehicles. Tuzun and Burke [54] developed a two-phase tabu search (TS) for the location-routing problem with uncapacitated depots. Wu et al. [59] described a heuristic based on a combined TS and simulated annealing method for capacitated location-routing problem. Recently, a cooperative Lagrangian relaxation-granular TS heuristic that is, on average, the most effective on benchmark instances from the literature for solving the capacitated location-routing problem was developed by Prins et al. [46].

The second relevant stream is related to the supporting infrastructure network design for alternative fuel vehicles. In the last decade, this research area has received extensive attention. Most papers focused on how to locate refueling stations related to alternative fuels. Based on the flow-capturing location model (FCLM) proposed by Hodgson [24] and Berman et al. [9], Kuby and Lim [28] studied the flow-refueling location model (FRLM) which locates a given number of stations to maximize the vehicle flow on the paths with the driving range consideration. Kuby and Lim [29] extended the FRLM by adding candidate sites along arcs to improve the coverage of the network. Upchurch et al. [56] established the capacitated flow refueling location model that limits the number of vehicles refueled at each station. A recent development deals with route deviations that drivers are likely to make from their shortest paths in order to refuel their EVs when the refueling station network is sparse [27]. The recreation-oriented scooter recharge stations location problem [57] and the battery exchange stations location problem in the tourism transport [58] were proposed based on the flow-capturing model. For deploying battery swapping network infrastructure and battery management, Mak et al. [37] developed two distributionally robust optimization models for the swapping station location problem under ambiguous information on demand distribution. Apart from the service station location problem for alternative fuel vehicles, there are a few publications that consider VRPs with specific characteristics of EVs. Conrad and Figliozzi [15] established a flow-arc formulation to study the recharging vehicle routing problem (RVRP) where EVs with driving range limitation are able to recharge at customer locations mid-tour. Erdogan and Miller-Hooks [20] are the first to propose a green vehicle routing problem (G-VRP) with the possibility of refueling vehicles at the existing alternative fueling stations along the routes. Schneider et al. [50] proposed a hybrid heuristic to solve the EV routing problem with time window and recharging stations. Kang and Recker [25] developed a hydrogen refueling facility location problem with full-day scheduling and routing considerations using tours and time windows. The proposed model and a modified column generation are applied to a case study of Hydrogen Fuel Cell Vehicles refueling stations. Such works mainly focused on recharging EVs. In another related study, Mirchandani et al. [39] discussed the EV fleet scheduling and battery swapping station location issues in transforming service fleet vehicles to EVs. They defined the EV shortest route problem related to EV routing issues, which aims at finding the shortest route such that an EV with limited range can successfully reach a given end point in a network from a start point. Li [34] established a vehicle-scheduling model for electric transit buses with battery swapping at a battery station and an exact algorithm was introduced to solve the problem. Above all, these works studied either the location problem of battery service stations or the electric vehicle routing problem. There was little literature that studied station location and EV routing problem simultaneously. In fact, because the battery standard and the public battery swapping service stations are not always accessible for EVs, the joint decision of the battery swap station location and EV routing is a real problem for the logistics firms.

Contributions of this paper are summarized as follows: first, we introduce a new problem called BSS–EV–LRP to determine the location strategy of battery swap stations and the routing plan of a fleet of electric vehicles simultaneously under battery driving

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

range limitation. Second, we propose two algorithms to solve the BSS–EV–LRP and test their performance on different sizes of instances. These techniques are intended to provide decision support for a logistics company establishing and operating the EVs distribution network. Finally, on the basis of computational results, we systematically conduct an economic and environmental analysis including comparison between basic and extended scenarios, sensitivity analysis on battery driving range and efficiency analysis about the vehicle emissions reduction when EVs are used in practice.

3. Model formulation

3.1. Problem description

In this paper, we study the delivery problem in which EVs deliver customers demand from the depot. The problem has a single fixed depot, a set of customers and their demands, a feasible set of candidate locations for BSSs without capacity constraints, and a fleet of identical EVs that delivers goods from the depot to the customers. In the case where a candidate station site is at a customer node, the candidate site can be treated as a dummy node which belongs to the set of candidate BSSs, whose distance from the customer node is zero. Each customer must be visited once by one vehicle. Given that we are modeling the delivery (rather than the pickup) problem, the customer demand carried at any point of a vehicles route is referred to as its remaining load. An EV's remaining load at any node, including the depot, cannot exceed the vehicle's capacity. Each route starts and ends at the depot. It is assumed that either a BSS exists at the depot or that sufficient time and recharging ports exist at the depot so that every EV has sufficient power to start each route with a fully charged battery. There is a fixed construction cost associated with establishing a station at each candidate BSS, and a unit shipping cost associated with any route of EVs. This problem is to determine the location strategy of the BSSs and the EVs' distribution routes to minimize the sum of the location and shipping cost so that all customers' demands are satisfied.

3.2. A basic mathematical model

In this section we propose a mathematical model of this problem. To clarify the model, notations that will be used in this paper are listed as follows. Sets:

I	set of customers indexed by i
J	set of candidate BSSs indexed by j
K	set of EVs indexed $k \in K$
R	set of vehicle routes indexed by $r \in R$
$\{o\}$	the single depot
$\{o'\}$	a copy of the depot $\{o\}$
V	nodes set indexed by v, $V = I \cup J \cup \{o\} \cup \{o'\}$.

Decision variables:

y_j	binary decision variable, taking a value of 1 if a BSS is located at node j and 0 otherwise
x_{ghk}	binary decision variable, taking a value of 1 if vehicle k goes from node g to node h and 0 otherwise.

Non-decision variables and parameters:

d_{gh}	the distance from node g to node h
γ_{ghk}	the shipping cost per mile from node g to node h with vehicle k
M	a large number
q_i	the demand at node $i \in I \cup J$, where $q_i = 0, \forall i \in J$
U_k	the loading capacity of the vehicle k
u_{ik}	the remaining load when vehicles k leaves node i
Q	the battery driving range or the maximal distance a fully charged battery allows
P^1_{gk}	the maximum distance that the remaining battery power allows when vehicle k arrives at node g
P^2_{gk}	the maximum distance that the remaining battery power allows when vehicle k leaves node g
f_j	the unit BSS construction cost at node j.

The model is now formulated as follows:
(Basic Model)

$$\text{minimize} \quad Z = \sum_{j \in J} f_j y_j + \sum_{g \in V} \sum_{h \in V} \sum_{k \in K} \gamma_{ghk} d_{gh} x_{ghk} \tag{1}$$

$$\text{subject to} \quad \sum_{g \in V \backslash \{o'\}, g \neq h} \sum_{k \in K} x_{ghk} = 1 \quad \forall h \in I \tag{2}$$

$$\sum_{g \in V \backslash \{o'\}, g \neq h} \sum_{k \in K} x_{ghk} \leq M y_h \quad \forall h \in J \tag{3}$$

$$\sum_{h \in V \backslash \{o\}, h \neq g} x_{ghk} - \sum_{h \in V \backslash \{o'\}, h \neq g} x_{hgk} = 0 \quad \forall g \in V \backslash \{o, o'\}, \ k \in K \tag{4}$$

$$\sum_{h \in V \backslash \{o\}} x_{ohk} - \sum_{h \in V \backslash \{o'\}} x_{ho'k} = 0 \quad \forall k \in K \tag{5}$$

$$\sum_{h \in V \backslash \{o\}} x_{ohk} \leq 1 \quad \forall k \in K \tag{6}$$

$$\sum_{k \in K} \sum_{h \in V \backslash \{o\}} x_{ohk} \leq |K| \tag{7}$$

$$u_{hk} \leq u_{gk} - q_h x_{ghk} + U_k (1 - x_{ghk}) \quad \forall g \in V \backslash \{o'\}, \ h \in V \backslash \{o\}, \ g \neq h, \ k \in K \tag{8}$$

$$u_{ok} \leq U_k \quad \forall k \in K \tag{9}$$

$$u_{hk} \geq 0 \quad \forall k \in K, \ h \in V \backslash \{o'\} \tag{10}$$

$$P^1_{hk} \leq P^2_{gk} - d_{gh} x_{ghk} + Q(1 - x_{ghk}) \quad \forall g \in V \backslash \{o'\}, \ \forall h \in V \backslash \{o\}, \ g \neq h, \ k \in K \tag{11}$$

$$P^2_{ok} = Q \quad \forall k \in K \tag{12}$$

$$P^2_{gk} = Q y_g \quad \forall g \in J \tag{13}$$

$$P^2_{hk} = P^1_{hk} \quad \forall h \in I \tag{14}$$

$$P^1_{hk} \geq 0 \quad \forall h \in V \tag{15}$$

$$y_j, x_{ghk} \in \{0, 1\} \quad \forall j \in J, \ \forall g \in V \backslash \{o'\}, \ \forall h \in V \backslash \{o\}, \ \forall k \in K. \tag{16}$$

The objective function (1) minimizes total cost including BSSs construction cost and EVs shipping cost. Constraints (2) ensure that each customer be visited by one vehicle. Constraints (3) guarantee that the EVs only swap their batteries at a located BSS. Flow balance for each vehicle at each node is ensured through constraints (4). Constraints (5) denote that each vehicle must return to the depot. Constraints (6) assign only one trip to a vehicle when the vehicle leaves the depot. The amount of vehicles is limited to $|K|$ by constraints (7). Constraints (8) record a vehicle's remaining load level based on node sequence and type. If vehicle k visits node h right after node $g (x_{ghk} = 1)$, the first term in the right-hand-side reduces the vehicle remaining load after leaving node h based on the demand at node h (if node h is the candidate site node, the

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

demand at node h is zero). Otherwise, constraints (8) are relaxed. It should be noted that the constraints (8) in the basic model prevent vehicles from returning to a BSS they have already visited. Constraints (9) ensure that the remaining load of vehicle k is no more than its capacity U_k when it leaves the depot. The remaining load must be nonnegative in constraints (10). These three set of constraints (8)–(10), which collectively deal with the vehicle capacity limitation, also serve to eliminate the possibility of subtour formation. The logic of Constraints (8) is also applied to Constraints (11), which track an EV's battery power level based on node sequence. If node h is visited by vehicle k after node g ($x_{ghk}=1$), the maximal distance that the remaining battery power allows upon arriving at node h is reduced based on the distance between node g and node h. In both (8) and (11), the same single variable, u_{hk} or P^1_{hk} on the left-hand-side, will be constrained by $|V|-2$ instances of the constraints, respectively. The variable in each case must be less than the lowest of all the right-hand-side's. However, it is worth noting that because an EV will replace the nearly depleted battery with a fully charged one at BSSs, two important variables P^1_{gk} and P^2_{gk} are used to represent the power levels while it enters or leaves a node g, respectively. Constraints (12) and (13), respectively, reset the battery power level to Q when EVs leave the depot or a located BSS. Constraints (14) ensure that the battery power level remains unchanged while EVs visit a customer node. Constraints (15) guarantee that each EV has sufficient battery power to visit the remaining customers and return to the depot. Constraints (11)–(15) collectively handle the battery driving range limitation. Finally, the decision variables' binary nature are stated by (16).

As mentioned above, this paper presents the delivery problem, in which EVs deliver customers' demand from the depot, and as each EV continues on its route the remaining load should always be nonnegative. For the pickup problem in which a fleet of identical EVs collects the customers' goods and brings them to the depot and the remaining vehicle capacity is nonnegative, the model formulation can be identical. In the pickup problem, U_k is redefined as the remaining capacity of vehicle k; u_{ik} denotes the remaining capacity when vehicle k leaves node i. Constraints (8) record a vehicle's remaining capacity level based on node sequence and type.

3.3. An extended mathematical model

The basic mathematical model is formulated with the assumption that every vehicle should pass by a station or a customer only once. However, an EV may return to a previously visited BSS to swap batteries after serving several customers. Such a circumstance is called *station revisit*. It is more efficient for the EV operation network to allow for the *station revisit* of vehicles seen in Fig. 1. Let the construction cost per station be 50. In the basic scenario, three candidate BSSs including B, C and D are located. The shipping cost is 358 and the objective function value is 508.

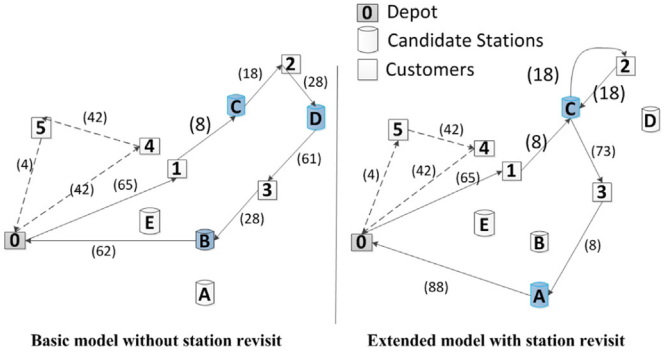

Fig. 1. Comparison between the basic and extended scenarios.

In the extended scenario with revisiting stations, one of the routes is $\{o-1-C-2-C-3-A-o'\}$ which visits the BSS C twice. Under the circumstance, only two candidate stations, A and C, are selected. The shipping cost is 366 and the objective function value is 466. The *station revisit* leads to 8% decrease in the objective function value.

As depicted in the basic model, Constraints (8) ensure that a BSS must not be revisited because a vehicle's remaining load is non-increasing and each left-hand-side variable u_{hk} in the Constraints is bounded by $|V|-2$ instances of constraints and the cost minimization results in that u_{hk} indicating the remaining load after leaving a BSS h must be equal to the lowest minimal remaining load when it leaves the last connected node ($u_{hk}=\min_g\{u_{gk}, g\in V\backslash\{o',h\}$, if h belong to $x_{ghk}=1\}$). In the formulation of the extended scenario, in order to formulate the *station revisit* of EVs, the variable u_{gk} is modified to be u_{ghk} which denotes the remaining load of the vehicle k when it reaches node g after leaving node h. Because u_{gk} denotes the remaining load at a node while u_{ghk} presents the remaining load on an arc, the constraints (8), (9) and (10) related to the vehicle capacity limitation are modified based on network flow model. The whole mathematic model of the extended scenario is presented below:

Extended model

Minimize : (1)

Subject to:

(2)−(7), (11)−(16)

$$\sum_{i\in V\backslash\{o',j\}} u_{ijk} = \sum_{i\in V\backslash\{o,j\}} u_{jik} \quad \forall j\in J,\ k\in K \tag{17}$$

$$\sum_{j\in V\backslash\{o,i\}} u_{ijk} \le \sum_{j\in V\backslash\{o',i\}} u_{jik} - q_i \sum_{j\in V\backslash\{o',i\}} x_{jik} + U_k\left(1-\sum_{j\in V\backslash\{o',i\}} x_{jik}\right) \quad \forall i\in I,\ k\in K \tag{18}$$

$$0\le u_{ijk}\le U_k x_{ijk} \quad \forall i\in V\backslash\{o'\},\ \forall j\in V\backslash\{o\},\ i\ne j,\ k\in K \tag{19}$$

In the extended model, when vehicles visit a BSS without service demand ($q_j=0,\ j\in J$), Constraints (17) represent that the sum of the remaining load of an EV entering a BSS is equal to the sum of the remaining load of an EV leaving a BSS, which guarantees the vehicle capacity balance and indicates that a BSS may be revisited more than once by relaxing constraints (8). For a customer node, the change in the remaining load level of an EV is tracked based on node sequence by constraints (18), which is identical to the constraints (8) for a customer node. Because a customer can be visited only once by constraints (2), if the vehicle k visits customer j ($\sum_{j\in V\backslash\{o'\}} x_{jik}=1$), the remaining load level is reduced by customer demand q_i. Otherwise, if the vehicle k does not visit the customer ($\sum_{j\in V\backslash\{o'\}} x_{jik}=0$), the constraints are relaxed. The constraints (17) and (18) make it possible for an EV to pass by a station more than once but visit a customer only once. In addition, the two sets of constraints are also the path elimination constraints. Constraints (19) limit the range of the remaining load level. If the vehicle k visits node j after leaving i ($x_{ijk}=1$), the maximal remaining load level u_{ijk} is at most the full vehicle capacity U_k. Otherwise, $u_{ijk}=0$.

In summary, both the basic and extended formulation of BSS–EV–LRP presented in this section are variants of a location-routing problem (LRP). Different from the previous LRP problems that seek to locate the depots and plan the routes by allocating customers to different depots, our problem (BSS–EV–LRP) aims at minimizing the total cost by locating a subset of candidate BSSs and allocating vehicles leaving from one fixed depot to swap batteries at these located BSSs such that the demand of each customer can be satisfied. Furthermore, it is obvious that BSS–EV–LRP is NP-hard: if the battery driving range is sufficiently large, it is

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

unnecessary to swap batteries and construct BSSs. This special case of BSS–EV–LRP is equivalent to the classic vehicle routing problem (VRP) which is NP-hard [28]. Therefore, BSS–EV–LRP is also NP-hard. In the following sections, we propose two hybrid heuristic algorithms for solving the BSS–EV–LRP.

4. Algorithms for the BSS–EV–LRP

4.1. The TS-MCWS heuristic

This section gives a brief description of a hybrid approach called TS-MCWS which combine Tabu Search and modified Clarke and Wright Saving algorithm to jointly solve the BSS–EV–LRP. In the hybrid algorithm, Tabu search is designed to search the location strategy and a modified Clarke–Wright Saving method undertakes the routing decision based on this location solution. Moreover, the two methods are coordinated so that an efficient solution exploration is performed.

4.1.1. Radius covering algorithm for BSS location initialization

An initial location solution to the BSS location subproblem is necessary in tabu search procedure. The TS-MCWS generates an initial location plan using a radius covering method. Let the battery driving range be Q. Define N_l to be the initial number of located stations. The main idea of the method is to select N_l BSSs which covers most of the customers within a radius of r_cQ from each candidate station, where $0<r_c\leq 1$ (1/3 in the implementation). The overall radius covering method outlines as follows.

Step 1: Create a *customers covered list* (*CCL*) that includes the number of covered customers within r_cQ radius for each candidate station. We propose two rules for customer covering. In the first and default rule, given a candidate BSS, all customers within the radius from it are recorded in CCL. In the initialization phase of the main loop, if the initial BSSs location strategy generated by using the first rule does not result in a feasible routing plan, the whole algorithm relocates N_l BSSs by using the second rule as follows: to avoid redundancy, the customers which have been covered by previous candidate stations are removed from the customer set. Rule 1 or rule 2 is repeated until each customer has been covered by a candidate stations.

Step 2: Rank all candidate stations in *CLL* in descending order.

Step 3: Return the first N_l BSSs as the initial located station.

4.1.2. Modified Clarke–Wright saving algorithm for vehicle routing subproblem

After generating a BSS location plan using radius covering or tabu search approach, we now consider solving the vehicle routing subproblem by using a modified Clarke and Wright Savings algorithm (MCWS). The Clarke and Wright Saving algorithm was originally proposed by Clarke and Wright for the classical VRP [13]. The method was further modified to a new variant of VRP called green-VRP introduced by Erdogan and Miller-Hooks [20]. We use a similar approach to create vehicle routes as a basis for comparison with another algorithm introduced later. The overall MCWS framework is outlined as follows.

Step 1 (*Initialization phase*).

Step 1.1 (*Back-and-forth routes construction*): For each customer i, create a back-and-forth route $(o-i-o')$.

Step 1.2 (*Infeasible tours repair*): For each route, check for feasibility with regard to the battery driving range limitation. If the route is feasible, add the route in the *feasible routes set* (*FS*). Otherwise, place a located BSS j_1 with the least insertion cost between $\{o\}$ and i such as $(o-j_1-i-o')$. If the route is feasible, add it in the *FS*. Otherwise, add another located BSS j_2 similarly such as $(o-j_1-i-j_2-o')$. If the route is still infeasible, discard the route and add it to the *infeasible routes set (IS)*. Otherwise, add it to the *FS*. Finally, if *IS* is empty, go to *Step* 2. Otherwise, there are some infeasible routes, return all initial routes and set the objective function value of the infeasible solution as *infinite*.

Step 2 (*Route merge phase*).

Step 2.1 (*Savings pair list generation*): For each pair of feasible routes in the *feasible routes set (FS)*, compute the saving distance. First, identify two nodes (v_k^1, v_k^2) that are adjacent to the depot $o(o')$ for each feasible tour ($k\in FS$). Second, create a saving pair $[v_1, v_2]$ that is composed of two nodes from different feasible tours k_1 and k_2. Calculate the saving value $s[v_1, v_2]$ if tour k_1 and tour k_2 are merged by this pair $[v_1, v_2]$: $s[v_1, v_2]=d(o, v_1)+d(v_2, o')-d(v_1, v_2)$. Finally, sort the savings pair in the SPL in descending order as the saving distance $s[v_1, v_2]$.

Step 2.2 (*Node pair merge*): Merge the feasible routes by the saving value for each saving pair in the SPL. At first, choose the first node pair $[v_1, v_2]$ in the SPL. Add all routes visiting node v_1 and v_2 in the set R_1 and R_2, respectively. For each route $k_1\in R_1$, select route $k_2\in R_2$ in order and merge these two routes as follows: first, delete the arcs $(o, v_1)\in k_1$, $(o, v_2)\in k_2$ and connect nodes (v_1, v_2) directly such that two routes are merged. Check for the vehicle capacity constraints, if the merged route is infeasible, give up the merge and select next route in R_2. Otherwise, if the new route violates the battery driving range limitation, insert a located BSS with the least insertion cost between v_1 and v_2. If both constraints are met, add the merged route to the *feasible routes set (FS)* and remove k_1 from R_1 and k_2 from R_2 respectively. Go to next route in R_1. If R_1 or R_2 is empty, terminate the merge procedure of this node pair. Finally, remove the pair of nodes $[v_1, v_2]$ from the SPL. The process is repeated until SPL is empty.

Step 3 (*Improvement phase*).

Step 3.1 (*Redundancy elimination*): For each route with more than one BSS, consider whether it is feasible to remove one or more BSSs from the route using greedy-drop method. Then update the *feasible routes set (FS)*.

Step 3.2 (*Local search*): To further improve the routing plan, TS-MCWS performs the local search procedure (LS) [46]. LS explores two neighborhoods including *MOVE* operator for intra-route and *SWAP* operator for inter-route. The *MOVE* procedure places one node from current position to another position in the same route. And the *SWAP* procedure exchanges two nodes from two routes. In both operators, the vehicle capacity and battery driving range limitation must be respected. Finally, update the *feasible routes set*.

Step 4 (*Termination criterion*): If any two routes in the *feasible routes set* can be further merged, go to Step 2. Otherwise, stop the MCWS and return current objective function value.

4.1.3. Tabu search for BSS location subproblem

Step 1 (*Initialization phase*): For a given number of located BSSs, with the solution S_0 obtained by the radius covering or MCWS, calculate the objective function value $Z(S_0)$. Initialize the current solution $S\leftarrow S_0$ and the best-known solution $S^*\leftarrow S_0$.

Step 2 (*Neighborhood search*): Let J_l be a set of located stations and $J_u=J\backslash J_l$ be a set of unlocated stations in current solution S. Start from an empty tabu list. The tabu length *tabuList* represents the number of iterations when pair (j_l, j_u) is in the tabu list, where (j_l, j_u) means one-opt exchange between a located BSS $j_l\in J_l$ and a unlocated BSS $j_u\in J_u$. In our implementation, *tabuList*=5. The neighborhood structure is defined as follows: for each located station $j_l\in J_l$, choose an unlocated station $j_u\in J_u$ at random and replace j_l with j_u such that a set of neighboring solutions $\overline{N}(S)$ is generated.

Step 3 (*Routing phase*): For each generated neighborhood $S'\in\overline{N}(S)$, perform the MCWS to create the corresponding routes.

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

Step 4 (*Update phase*): Update the current solution S as $\text{argmin}_{S' \in \overline{N}(S)}[Z(S')]$, where S' are not in the tabu list. However, as for the aspiration criteria, if the objective function value of a neighboring solution is better than the current best-known value, the exchange is still allowed even if it is in the tabu list. If $Z(S) < Z(S^*)$, set $Z(S^*) \leftarrow Z(S)$ and $S^* \leftarrow S$. Finally, keep track of the current exchange and renew tabu state in the tabu list. If a given number of iterations has been reached, stop tabu search. Otherwise, go to Step 2.

4.1.4. Framework of TS-MCWS

Step 1: Set the initial number of BSSs $N_I = 1$ if $|J| \le N_J^0$; otherwise, $N_I = N_I^0$, where $N_I^0 > 1$.

Step 2: Generate an initial location strategy with N_i BSSs by using the radius covering algorithm. Obtain a routing plan by performing MCWS based on the initial location strategy. An initial solution S_T^0 to BSS–EV–LRP is created. Set the current solution $S_T \leftarrow S_T^0$ and the best-known solution $S_T^* \leftarrow S_T^0$.

Step 3: Call tabu search procedure on S_T to renew the current solution S_T. If $Z(S_T) < Z(S_T^*)$, update the best-known solution $S_T^* \leftarrow S_T$.

Step 4: If a predefined number of iterations without an improvement in the objective function value of the best-known solution N_{Imp} has been reached or all BSSs have been located, stop TS-MCWS. Otherwise, $N_I = N_I + 1$, go to Step 2. In our implementation, if the number of candidate BSSs $|J| \le 50, N_{Imp} = 3$, otherwise, $N_{Imp} = 5$.

4.2. The SIGALNS heuristic

This section provides a description of a hybrid heuristic called SIGALNS including four main algorithmic components: modified *S*weep heuristic, *I*terated *G*reedy, *A*daptive *L*arge *N*eighborhood *S*earch and improvement heuristic for the EV location routing problem (BSS–EV–LRP). In the next few paragraphs, we present some details of each phase for different subproblems, such as initialization, location and vehicle routing in Sections 4.2.1–4.2.3, followed by an improvement method in Section 4.2.4. Thereafter, the hybridization scheme of SIGALNS is proposed further.

4.2.1. Phase 1: Sweep heuristic for the initial solution construction

In the initialization phase, an initial solution is created using a modified sweep algorithm (MS) while the battery driving range limitation and BSS location strategy are ignored. Sweep algorithms were firstly proposed by Gillet and Miller [21] and applied to solve vehicle routing problem. In our implementation, MS contains two decomposed subproblems, namely, the customer clustering and the travelling salesman problem. The first subproblem is used to generate customer clusters as follows: first, rank customers in the ascending order of polar coordinate angles from the depot. Then beginning with the customer with the smallest angle, place customers in a single cluster sequentially as long as the total customer demand in the cluster satisfies the vehicle capacity constraints. Otherwise, create a new cluster and restart the process. If each customer is assigned to a cluster, terminate the procedure. The second subproblem determines the vehicle routes of each cluster by solving a TSP. In this paper, for all customers of each cluster along with the depot, a TSP path is constructed by using the Solomon's well-known I1 heuristic where each unserviced customer with the minimal insertion cost is added into the route sequentially [52]. The MS procedure is described in Algorithm 1.

As indicated above, because the BSS location decision is ignored, the initial solution may violate the battery driving range limitation. Therefore, this location subproblem is solved in the next section.

Algorithm 1. Modified sweep algorithm (MS).

Require: Vehicle capacity U, customer demand q_i;
Ensure An initial vehicle routing plan S_0.

```
Rank customers in the ascending order of polar angles;
Create a cluster C_k ← ∅, k ← 1;
for each i ∈ I do
  if (q_i + Σ_{t∈C_k} q_t ≤ U) then
    C_k ← C_k ⋃ {i};
  else
    Start a new cluster C_{k+1} ← ∅;
    C_{k+1} ← C_{k+1} ⋃ {i};
  end if
end for
for each C_k ∈ C do
  Solve a TSP on C_k to obtain a route R_k;
end for
S_0 ← ⋃_{k=1}^{|R|} R_k;
```

4.2.2. Phase 2: iterative greedy heuristic for the stations location

After generating a routing plan in the initialization or routing phase, the location phase selects a subset of candidate BSSs and allocates them into different routes with minimal sum of construction and allocation cost using an iterated greedy algorithm (IG). This section first analyzes the allocation cost when a BSS is inserted to a route. Then IG is proposed to determine the BSS location strategy and thus obtain an overall feasible solution to the BSS–EV–LRP.

(1) *Allocation cost analysis*: In the location phase, IG firstly removes all located BSSs in a solution to BSS–EV–LRP and then relocates BSSs in order to seek a better location decision. Removing all stations may result in violations of the constraints in terms of battery driving range limitation. Given a candidate battery station set J and a solution to the vehicle routing subproblem represented by a set of routes $R = \{r_1, \ldots, r_m, \ldots, r_n, \ldots, r_{|R|}\}$, some stations with the least cost increment must be placed in these routes to improve the feasibility of the solution. So this section analyzes allocation cost to select BSSs for insertion in the current partial routes before proposing an iterated greedy heuristic.

(a) *Breaking point*: Let $r_k = \{v_0 = o, v_1, v_2, \ldots, v_m, v_{m+1} = o'\}$ $(r_k \in R)$ be the current route. Because of battery driving range limitation, there may be some *breaking points* in r_k. A node v in r_k is called *breaking point* if it satisfies the following condition:

$$\{v | P_{vk}^1 < 0,\ v \in r_k\}, \tag{20}$$

which means that the node is unreachable because the battery has been depleted before arriving at node v. We define v^* as the *first breaking point* of route r_k, which we calculate as $v^* = \{v | P_{v^*,k}^1 < 0, P_{v',k}^1 \ge 0, \forall v' < v^*, v', v^* \in r_k\}$. v^* indicates that all nodes before v^* in route r_k are reachable.

(b) *Node feasibility state*: Let P_{vk}^1 be the maximal distance that the remaining battery power allows when vehicles in r_k arrives at node v. In order to further evaluate the feasibility improvement after locating a BSS, the *feasibility state* of a node v, denoted by q_{vk}, is defined as follows:

$$q_{vk} = \min\{P_{vk}^1, 0\}, \quad v \in r_k. \tag{21}$$

Obviously, $q_{vk} \le 0$. And $q_{vk} = 0$ indicates that the node v in route r_k is reachable $(P_{vk}^1 \ge 0)$. Otherwise, $q_{vk} < 0$ and the route is infeasible. The *node feasibility state* measures the extra required battery power to reach the node and also indicates whether a BSS is needed to swap a battery before arriving there. In addition, we compute the *worst node feasibility state* in route r_k as $q_k^* = \min_{v \in r_k \setminus \{o\}} q_{vk}$. The smaller q_k^* is, the worse the solution feasibility becomes.

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

(c) *Allocation cost*: When a candidate BSS $j \in J$ is selected and placed at position $\overline{v}$ after the node v in the route r_k, allocation cost $c_{j,k}^{\overline{v}}$ is used to evaluate the solution feasibility improvement and the objective function value increment. To minimize the total allocation cost, such a solution where more *breaking points* are eliminated or the whole *feasibility state* of all nodes gains a larger improvement is preferred. More precisely, let $\overline{v}$ be an insertion position for a BSS after node v, $g_{j,k}^{\overline{v}}$ be the *insertion gain*, $l_{j,k}^{\overline{v}}$ be the *insertion loss* and $e_{j,k}^{\overline{v}}$ be the *extra penalty*, the allocation cost after locating and inserting a BSS $j \in J$ at node $\overline{v}$ after the node v in the route r_k, denoted by $a_{j,k}^{\overline{v}}$, is defined as

$$a_{j,k}^{\overline{v}} = \beta_1(-g_{j,k}^{\overline{v}}) + \beta_2 l_{j,k}^{\overline{v}} + \beta_3 e_{j,k}^{\overline{v}}, \quad j \in J, \ \ v \in r_k, \ \ r_k \in R, \tag{22}$$

where

$$\beta_1 + \beta_2 + \beta_3 = 1, \quad \beta_1, \beta_2, \beta_3 \geq 0.$$

$$g_{j,k}^{\overline{v}} = \left(\sum_{v \in r_k \setminus \{o\}} (q_{vk'} - q_{vk}) \right) \left(\frac{1 + |q_k^*|}{1 + |q_k^{*\prime}|} \right), \tag{23}$$

$$l_{j,k}^{\overline{v}} = d_{v_n,\overline{v}} + d_{\overline{v},v_{n+1}} - d_{v_n,v_{n+1}}, \quad v_n, v_{n+1} \in r_k, \tag{24}$$

$$e_{j,k}^{\overline{v}} = M|q_{\overline{v},k}|, \tag{25}$$

The *insertion gain* $g_{j,k}^{\overline{v}}$ is calculated in Eq. (23). $g_{j,k}^{\overline{v}}$ measures the feasibility improvement of route r_k when station j is inserted at position $\overline{v}$ after node v. As defined above, q_{vk} and q_k^* denote the feasibility state in node v and the *worst node feasibility state* in route r_k before insertion, respectively. $q_{vk'}$ and $q_k^{*\prime}$ represent that after insertion. The first component on the right estimates the total quantity improved of feasibility states for all the nodes in route r_k. If $q_k^* < q_k^{*\prime} \leq 0$, which indicates that the *worst node feasibility state* is improved, then the second component is greater than 1 and thus magnifies the aggregate improvements in the first component.

The *insertion loss* $l_{j,k}^{\overline{v}}$ is defined in Eq. (24), where v_n and v_{n+1} is, respectively, the predecessor and successor node at position $\overline{v}$. $l_{j,k}^{\overline{v}}$ approximates the travelling distance increment resulting from the insertion of station j after node v.

The *extra penalty* $e_{j,k}^{\overline{v}}$ is set equal to Eq. (25), where M is a user-defined big number and $M = 10^4$ in our paper. $q_{\overline{v},k}$ denotes the feasibility state of station j at position $\overline{v}$. If the station j itself is a *breaking point*, the station is unreachable and the infeasibility insertion is penalized. Furthermore, let $P_{v,k}^2$ be the maximal distance that the remaining battery power allows when vehicles k leave node v, for each node v in route r_k, there is a *reachable BSSs set*, denoted by $J_v^k \subseteq J$, which is computed as $J_v^k = \{j \in J | P_{v,k}^2 \geq d_{v,j}\}$ $(v \in r_k)$. For any BSS j in J_v^k, if it is located and the vehicle k visits it after leaving node v, the extra penalty $e_{jk} = 0$.

Allocation cost is the main criteria to select stations for location and insertion among customers. Based on this rule and the above definitions, we propose the following algorithm IG to decide location strategy.

(2) *An iterated greedy algorithm* (*IG*): This section proposes an iterated greedy algorithm (IG) to solve the battery swap station location subproblem. The basic idea is to search for or maintain the feasibility of a solution by selecting and inserting stations in each route iteratively. The whole procedure contains three phases, namely feasibility identification, station location and postoptimization. More precisely, the infeasible routes are selected in the first phase. Next, the best stations and their optimal insertion positions with the least allocation cost are determined. To overcome local optima, a random selection scheme is introduced. The first two phases are repeated until all routes are feasible. Finally, the solution is further improved by applying a local search. An overview of the heuristic procedure is presented in Algorithm 2.

Algorithm 2. An iterated greedy algorithm.

Require: A solution S_0, location cost f_0 of a station.
Ensure A feasible BSS–EV–LRP solution S.

```
1:  Initialize construction cost with f_0 and set φ ← 0.
2:  Delete all located stations in S_0;
3:  S ← S_0;
4:  while (φ = 0) do
5:    for each (r_k ∈ R) do
6:      Compute the worst feasibility state q_k^*;
7:      if (q_k^* < 0) then
8:        Choose and insert the best stations;
9:        Update node feasibility state in r_k;
10:       Update station construction cost;
11:     end if
12:   end for
13:   if all routes are feasible then
14:     φ ← 1;
15:   end if
16: end while
17: Apply local search procedure;
```

Let f_0 be the construction cost of a station, ϕ indicates whether a solution is feasible. The unit construction cost of all BSSs is equal to f_0 initially. IG first declares that the input solution is infeasible $(\phi = 0)$ in Step 1. Given an initial solution S_0 to BSS–EV–LRP with a set of vehicle routes denoted by $R = \{r_1, \cdots, r_m, \ldots, r_n, \ldots, r_{|R|}\}$, all located stations in S_0 are removed in Step 2 and IG initializes the current solution S as S_0 in Step 3. In the feasibility identification phase seen in Step 6, the feasibility states of all nodes are calculated and then the worst feasibility state q_k^* is determined. If q_k^* is negative, which means the route is infeasible, IG goes to the station location phase in Step 8.

In the station location phase, for each route r_k, a segment of nodes $L_k \in r_k$, called the *search zone*, is created as follows: starting from the predecessor node of the first breaking point v^*, all nodes before v^* are added in L_k until reaching a BSS or the depot. All nodes in L_k are potential station insertion positions. Next, for each node $v \in L_k$, generate a *reachable BSSs set* J_v^k and the sum of construction cost and allocation cost $a_{j,k}^{\overline{v}}$ is recorded for each candidate station $j \in J_v^k$. Importantly, under the basic scenario where a route can only visit a station at most once, the stations which route r_k has visited are removed from J_v^k. A node with an empty J_v^k is removed from L_k. In our computational experiments, we focus on the extended scenario, but a comparison between these two scenarios is conducted afterwards. To choose a station insertion position, nodes in L_k are ranked in ascending order of the total cost and the position $\overline{v}$ after the node v indexed $\lfloor \epsilon_1^{\rho_1} |L_k| \rfloor$ is selected, where ϵ_1 is a random number between 0 and 1, $\rho_1 \geq 1$ is a parameter to introduce randomness and equal to 10 in our implementation. Similarly, to find the inserted station in the set J_v^k at the chosen position $\overline{v}$, sort stations in J_v^k in ascending of total cost and station $\bar{j}$ indexed $\lfloor \epsilon_2^{\rho_2} |J_v^k| \rfloor$ is located, where $\epsilon_2 \in (0,1)$ is also a random number, $\rho_2 \geq 1$ is a deterministic parameter equal to 10 in our paper. After inserting station $\bar{j}$ after node v, Step 9 updates the *node feasibility states* in r_k. Because a station may serve multiple routes, Step 10 sets the construction cost of each located station as 0. An example is used to illustrate the process of locating a station. In Fig. 2, route $r_k = \{o, 1, 2, A, 3, 4, 5, 6, 7, o'\}$ and station A has been located. Let battery driving range be 100, numbers in bracket be the maximal distance the remaining battery power allows when vehicles leave a node. The *first breaking point* is node 6, so the *search zone* $L_k = \{5, 4, 3, A\}$. Because the *reachable BSSs set* of node 5 is empty, customer 5 is removed from search zone. After ranking

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

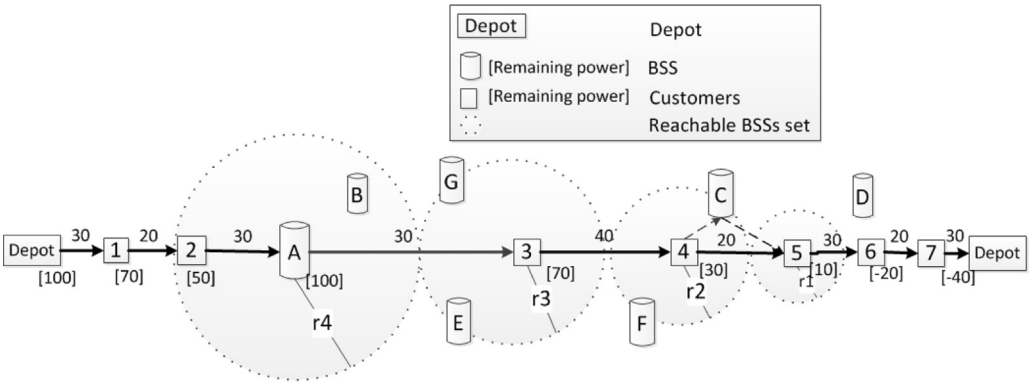

Fig. 2. An example of station location.

all nodes in L_k in ascending of the least sum of construction and allocation cost in each *reachable BSSs set*, node 4 is chosen at random. If candidate station C is chosen by the random scheme, it is inserted after node 4 and its construction cost becomes 0. In Steps 13–15, if all routes are feasible, set $\phi = 1$ and terminate the second phase. Otherwise, continue the location process.

At the end of the IG, a feasible solution to BSS–EV–LRP ia available. In the third improvement phase, it is further improved by a local search in Step 17 based on two neighborhood search operators in the following order: (1) *EXCHANGE*. One located station is replaced by another nearest located station in the same route or in a different route. (2) *MOVE*. One located station is closed and another nearest unlocated station opens. Each operator is performed at every located station and a new location strategy is accepted if it is still feasible and the objective function value is better.

The heuristic SIGALNS proposed to solve BSS–EV–LRP carries out a cooperative method by exchanging information iteratively between location and routing phase: In IG indicated above, all located stations are removed from a solution first and the new location strategy strongly depends on the vehicle routes; on the other hand, the routing plan is also strongly affected by the located stations. The vehicle routing subproblem is solved by the following adaptive large neighborhood search heuristic (ALNS).

4.2.3. Phase 3: adaptive large neighborhood search heuristic for the vehicle routing

ALNS was first proposed by Ropke and Pisinger [49] for the pickup and delivery problem. Its new developments and recent applications can be found in the recent literature [47,16,40]. The main idea of ALNS is to search the large neighborhood by removing a given number of customers from a solution and then reinserting them to construct a new solution. At each iteration, the removal and insertion operators are chosen adaptively based on their historical success [17]. Considering the characteristics of the BSS–EV–LRP, we propose several new removal and insertion operators in our ALNS procedure.

(a) *Framework of ALNS*: We now describe the main elements of our ALNS for BSS–EV–LRP according to the framework presented by Laporte et al. [33], which contain *large neighborhood*, *removal and insertion operators*, *adaptive search mechanism*, *penalized objective function*, *acceptance and termination criteria* as follows.

The first element is the *large neighborhood*. At each iteration, n_q customers are removed from the routes and put into a request bank *RB*, which is a set of removed customers, by calling a removal operator. Next, these customers are reinserted into the routes by using an insertion operator. The n_q is randomly chosen within the interval $[\delta_1|I|, \delta_2|I|]$, where $\delta_1, \delta_2 \in (0,1)$ and $|I|$ is the number of customers.

The second element is the *removal and insertion operators*. The removal operators remove n_q customers and put them into the request bank. In our ALNS, apart from the basic removal heuristics in the current review such as *random removal*, *worst removal*, *related removal* and *request graph removal*, we introduce four new operators including *single point removal*, *two point removal*, *binary removal* and *station based removal*. The insertion operators reinsert the customers in the request bank into the selected route to create a new solution. We not only use two traditional insertion operators such as *basic greedy insertion* and *regret-k insertion* but also propose two new insertion operators called *advanced greedy insertion* and *advanced regret-k insertion*.

The third element is the *adaptive search mechanism*. The adaptive search mechanism consists of the adaptive selection of removal–insertion operators and the adaptive adjustment of operators' weights. The search is divided into a number of *segments*. Each segment contains ϖ (ϖ=50) iterations. At each iteration, a removal or insertion operator is chosen by a roulette-wheel mechanism. Let $w_{i,j}$ be the weight of the operator i at segment j, operator i is selected with probability $p_{i,j} = w_{i,j} / \sum_{h \in H} w_{h,j}$, where H is the set of removal operators or insertion operators. As for the adjustment of weights, all operators' weights are 10 initially. Operators are updated at the end of each segment j adaptively as follows: if $\varepsilon_{i,j} \neq 0$, $w_{i,j+1} = (1-\theta)w_{i,j} + \theta\pi_{i,j}/\varepsilon_{i,j}$; otherwise, $w_{i,j+1} = w_{i,j}$, where $\varepsilon_{i,j}$ and $\pi_{i,j}$ are the number of times operator i has been used in the jth segment, and the score of operator i in the jth segment, respectively. θ is the so-called reaction factor and we use $\theta = 0.3$ in our study. The score $\pi_{i,j}$ is computed as follows: the score is set to 0 initially and incremented by $\sigma_{i,h}$ depending on the historical performance of operator i at an iteration h; if a new global best solution is found by a pair of removal–insertion operators, they obtain score $\sigma_{i,h} = 50$; if the current solution is improved, they obtain score $\sigma_{i,h} = 20$; if the current solution is not improved but the new solution is accepted, they obtain score $\sigma_{i,h} = 10$.

The fourth element is the *penalized objective function*. Instead of restricting the search in a feasible region, we allow violation of the battery driving range limitation by using a penalized objective function:

$$Z_{penalized} = \sum_{j \in J} f_j y_j + \sum_{g \in V} \sum_{h \in V} \sum_{k \in K} \gamma_{ghk} d_{gh} x_{ghk} - M \sum_{k \in K} \sum_{v \in r_k} q_{vk} \tag{26}$$

where M is a very large number like 10^4 and $q_{vk} \leq 0$ is the *node feasibility state* proposed in the previous section.

The last element is the *acceptance and termination criteria*. We implement the element by following the criteria introduced

by [1]. A new solution is accepted based on a simulated annealing (SA) criterion. The criterion always accepts a better new solution. At the same time, it accepts a worse new solution with probability $e^{-(Z(s')-Z(s))/T}$, where s and s' are the current and new solution, respectively. The current temperature T is set to T_0 initially and is updated by $T_n = cT_{n-1}$ at each iteration. T_0 is set equal to 10,000 and the cooling rate is fixed to 0.995 in our implementation. The whole ALNS terminates when the maximal number of iterations $ITER^{ALNS}$ is reached. Let S_0, S, S' and S^* be the initial, current, neighborhood and the global best solution, respectively. The pseudocode of ALNS is presented in Algorithm 3.

Algorithm 3. Adaptive large neighborhood search algorithm.

Require: An initial solution S_0 obtained from location phase;
Ensure The global best solution S^*;
$S \leftarrow S_0$, $S^* \leftarrow S$;
$iter \leftarrow 1$
while $iter < ITER^{ALNS}$ **do**
  select a pair of removal and insertion operators;
  $S' \leftarrow S$;
  apply the removal operator to S';
  apply the insertion operator to S';
  if the acceptance is satisfied **then**
    $S \leftarrow S'$;
  end if
  if $Z(S) < Z(S^*)$**then**
    $S^* \leftarrow S$;
  end if
  update the score and weight of each operator;
  $iter \leftarrow iter + 1$;
end while

(b) *Basic removal operators*: This subsection introduces four basic removal operators which can be found in the current literature.

Random removal (*RaR*): This operator removes n_q customers from the current solution at random and puts them into the request bank.

Request graph removal (*RGR*): This operator was first proposed by Pisinger and Ropke [44]. It creates a complete and undirected graph, in which a node denotes a customer and the weight of each edge (i,j) is the number of times the edge is passed in the N_{best} best-known solution. The bigger the weight of two customers, the more similar they are. So the removal mechanism is the same with the related removal operator introduced as follows. A more detailed description of this operator is given by Ribeiro and Laporte [47].

Basic worst removal (*BWR*): This operator removes customers with high *removal gain*, namely the difference between the cost when a customer is in the current solution and the cost when it is removed [23], which is defined as $Z(S) - Z_{-i}(S)$. Because removing a customer only decreases the traveling distance of a route and does not change the location plan, the *removal gain* is equal to $d_{i-1,i} + d_{i,i+1} - d_{i-1,i+1}$, where $d_{i-1,i}, d_{i,i+1}$ and $d_{i-1,i+1}$ are the distances between two nodes, $i-1$ and $i+1$ are the predecessor and successor of customer i, respectively. Sort all customers in descending order of the *removal gain*. The customer indexed $\epsilon_3^{\rho_w}|I|$ is removed from the current solution, where ϵ_3 is a random number between 0 and 1, ρ_w is a predefined deterministic parameter and $|I|$ is the number of customers in the current solution.

Related removal (*ReR*): This operator aims at removing those customers that are similar to the customer that has been removed [51]. The first seed customer is chosen randomly. Customers i and j are said to be similar if the similarity measure $sim(i,j)$ is small: $sim(i,j) = \alpha_1 d_{i,j} + \alpha_2 |q_i - q_j| + \eta_{i,j}$, where α_1 and α_2 are weights between 0 and 1 ($\alpha_1 + \alpha_2 = 1$), $d_{i,j}$ is the distance between customers i and j, $|q_i - q_j|$ is the difference between their demand q_i and q_j. $\eta_{i,j} = 1$ if these two customers are in the same route, otherwise, $\eta_{i,j} = 0$. Like BWR, the random selection of a related customer to be removed is controlled by a deterministic parameter ρ_r.

(c) *Basic insertion operators*: Removal operators remove some customers from a current solution and they are reinserted using one of the insertion operator in the next step. Two basic insertion operators are used in our algorithm: *greedy insertion* and *regret-k insertion*. Each operator terminates when all the customers in the request bank are inserted to the routes.

Basic greedy insertion (*BGI*). This operator inserts the removed customer with the least objective function value increment at its best inserting position from *request bank RB* repeatedly. In detail, for customer i and route r_k, $\Delta Z_{i,k} = Z_{i,k} - Z_{i-,k}$ denotes the added cost after inserting i into route r_k at its best inserting position. Like BWR, $\Delta Z_{i,k}$ can be calculated as $d_{i-1,i} + d_{i,i+1} - d_{i-1,i+1}$. The selected customer i^* is determined as: $i^* = \text{argmin}_{i \in U}\{\min_{r_k \in R} \Delta Z_{i,k}\}$.

Basic regret-k insertion (*BRkI*): This operator was originally proposed by Ropke and Pisinger [49]. It reinserts every removed customer with the largest regret value at its best inserting position repeatedly. Let $\Delta Z_{i,j}$ denote the change in the objective function value by inserting customer i into the jth best route in its best position with regard to added distance as defined in BGI. Specially, $\Delta Z_{i,1}$ presents the cost after inserting the customer into the best route. The selected customer i^* to be reinserted in *regret-k* insertion is computed as $i^* = \text{argmax}_{i \in U}\{\sum_{j=2}^{k}(\Delta Z_{i,j} - \Delta Z_{i,1})\}$. In our study, we implement *regret*-2 called *Basic regret*-2 *insertion* (*BR2I*) and *regret*-3 called *Basic regret*-3 *insertion* (*BR* 3*I*).

(d) *ALNS applied to BSS–EV–LRP*: According to the characteristics of the BSS–EV–LRP, we propose five new removal operators and two new insertion operators as follows.

Advanced worst removal (*AWR*): In the basic worst removal, the *removal gain* is defined in terms of distance, namely $Z(S) - Z_{-i}(S) = d_{i-1,i} + d_{i,i+1} - d_{i-1,i+1}$. However, removing a customer not only decreases the travelling distance but also may improve the feasibility of a solution. In the advanced worst removal (AWR), the *removal gain* is modified as follows: $Z(S) - Z_{-i}(S) = \lambda_1(d_{i-1,i} + d_{i,i+1} - d_{i-1,i+1}) + \lambda_2(\sum_{v \in V_k} q_{vk} - \sum_{v \in V_k'} q_{vk}')$, where $\lambda_1 + \lambda_1 = 1$, q_{vk} and q'_{vk} are the node feasibility states before and after removal, respectively.

Station-based removal (*SBR*): This operator removes all customers connected to a random selected station repeatedly until n_q customers have been removed. Fig. 3 illustrates an example of the SBR. This operator chooses a located station at random such as station *A*. Then the customers {2, 3, 7, 8} connected to the station *A* are removed.

Single point removal (*SPR*): Having swapped a battery at the depot or a BSS, a vehicle has to reach all next customers without depleting batteries until arriving at another BSS or returning back to the depot. Therefore, the partial routes between two BSSs or a BSS and the depot, called *BSS service zone*, are strongly determined by the location of BSSs. Given the current station location strategy, this operator attempts to destroy the *BSS service zones* and construct a new routing plan while maintaining the battery driving range limitation. At first, this operator chooses a route randomly. Then it generates a removal position in this route at random and the customers between the position and the depot $\{o\}$ or the dummy depot $\{o'\}$ are removed, as depicted in Fig. 3. In the example, the *service zone* of the depot o is 1,2,3 and the *service zone* of station *A* is $\{4,5,6,7\}$. When removal position is 3, the customers between the removal position and the depot such as $\{1,2,3\}$ are removed.

Two points removal (*TPR*): This operator aims at destroying two *BSS service zone* simultaneously. The general idea is to choose two

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

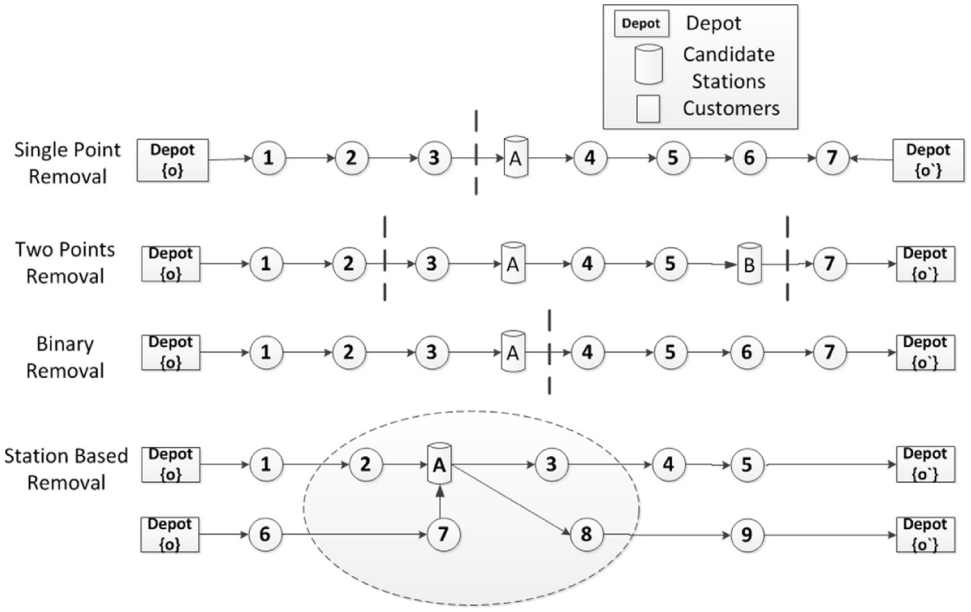

Fig. 3. Advanced removal operators.

removal position in a given route at random. The customer between them are removed, seen in Fig. 3. The two removal points are 3 and *B*. The customer 3 and the customers in the *service zone* of station *A*, namely {3, 4, 5} are removed.

Binary removal (*BiR*): This operator is a special case of the *SPR*. Different from generating a removal position randomly, *BiR* sets the removal position as the middle point of a route. For example in Fig. 3, the middle position is *A* and the customers between the depot and station *A* {1, 2, 3} are removed.

In the *basic greedy insertion* (*BGI*), the objective function value increment is calculated as added travelling distance, namely $d_{i-1,i}+d_{i,i+1}-d_{i-1,i+1}$. Similarly, the *Basic regret-k insertion (BRkI)* determines the regret value only based on added distance. However, inserting a customer not only increases the travelling distance but also changes the feasibility of the current solution. In the new insertion operators, the change in the objective function value ΔZ^{adv} after inserting customer i in the optimal position of route r_k is defined as follows:

$$\Delta Z^{adv} = \lambda_3(d_{i-1,i}+d_{i,i+1}-d_{i-1,i+1})+\lambda_4\left(\sum_{v\in V_k} q_{vk}-\sum_{v\in V_{k'}} q_{vk'}\right)+c_{penalty} \tag{27}$$

$$c_{penalty} = M_c \max\left\{\left(\sum_{v\in r_k} q_v - U\right), 0\right\} \tag{28}$$

where $\lambda_3+\lambda_4=1$, q_{vk} and $q_{vk'}$ are the node feasibility state before and after insertion, and $c_{penalty}$ is the penalty of exceeding vehicle capacity. The modified cost guides the operator to choose insertion which satisfies the battery driving range limitation and the vehicle capacity constraints. The weighted cost is applied to the *basic greedy insertion* operator as $\Delta Z_{i,k}=\Delta Z^{adv}$, called *advanced greedy insertion (AGI)*. In the *regret-k* insertion operator, regret value $\Delta Z_{i,j}=\Delta Z^{adv}$, called *advanced regret-k insertion (ARkI)*. In our study, we choose *advanced regret-2 insertion (AR2I)* and *advanced regret-3 insertion* (*AR3I*).

4.2.4. Phase 4: solution improvement

Phase 4 aims at further improving the new solution found by the former heuristics. For all located stations in each route r_k, the heuristic first selects a BSS in order and then removes the station from the route. If removal results in a driving range violation, the route is split at the position with the least *cost* $\psi_{i,k}= d_{i,o}+d_{o,i+1}-d_{i,i+1}$, where i is the split position and o is the depot. If a better solution is found, the current solution is updated. The procedure is repeated until the number of vehicles reaches the predefined maximum number of vehicles or the solution cannot be improved further.

4.2.5. Algorithmic framework

As depicted above, the framework of our heuristic called SIGALNS consists of four phases. After initialization in the first phases, the location and routing subproblems are solved successively. Finally, the solution is improved in the fourth improvement phase. A sketch of the hybrid algorithm is outlined in Algorithm 4. In Step 1, a starting solution S_0 is created using a modified sweep procedure (MS). Then, the current solution S and the global best solution S^* are initialized to be S_0 in Step 2. More importantly, instead of initializing the selection weight of each operator in ALNS, we finish it in Step 3 so that the historical success of operators can be recorded globally which will be beneficial to improve the performance of ALNS. Thereafter, in each iteration, all located stations are removed first in Step 7 to restart locating BSSs depending on the results of vehicle routing. Next, in Steps 8 and 9, the station location subproblem and vehicle routing subproblem are solved successively. In addition, the weights of all operators are updated in ALNS. The new neighborhood solution is improved further in Step 10. Similar to ALNS, given a neighborhood solution S' and the current solution S, the acceptance criterion from simulated annealing (SA) is used in Steps 11–13. In the end, the global best solution S^* is updated. The whole procedure of SIGALNS is repeated for $ITER^{SIGALNS}$ iteration.

Algorithm 4. Framework of SIGALNS.

Require: Distance matrix D, customers' demand q_i, vehicle capacity U, battery driving range Q
Ensure A solution to BSS–EV–LRP
1: Apply MS to generate an initial solution S_0;
2: $S, S^* \leftarrow S_0$;
3: Initialize weights of removal and insertion operators;
4: $iter \leftarrow 1$

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

```
while iter < ITER^SIGALNS do
  S' ← S;
  Remove all located stations in S';
  Apply IG to S' to set up the best stations;
  Call ALNS to S' to determine the vehicle routes;
  Improve S' by an improvement procedure
  if the acceptance criterion is satisfied then
    S ← S';
  end if
  if (Z(S) < Z(S*)) then
    S* ← S;
  end if
  iter ← iter + 1;
end while
```

5. Computational experiments

This section presents the computational experiments to evaluate the performance of two algorithms: *TS-MCWS* and *SIGALNS*. The algorithms were implemented in Java and run on a Dell PC Inspiron 545S with Intel Core Duo, 2.93 GHz processor and 2.00 GB of RAM. Section 5.1 describes several sets of instances and the parameter setting. Section 5.2 compares all removal and insertion heuristics in ALNS and selects a best operators combination. Finally, the computational results on benchmark instances are listed in Section 5.3

5.1. Test instances and parameter setting

In our studies, we use four sets of data applied to classic CVRP in small-, medium- to large-size. The data are publicly available at http://neo.lcc.uma.es/vrp/vrp-instances/. In the small-size sets, Augerat et al. [4] introduced three sets of instances, of which we used *Set P* in our experiments. It contains 24 instances with 16–101 customers and 2–15 vehicles. Another set generated by Rinaldi and Yarrow [48] contains only one instance with 48 customers. For the medium-size, the data set proposed by Taillard [53] contains 23 instances. Specifically , the number of customers is either 75, 100 or 150 and they are generated using four kinds of customer spatial distribution. The large-size set, with 20 instances with up to 480 customers, was taken from Golden et al [22]. In the computational experiments, it is assumed that all nodes are candidate battery swap stations, and Euclidean distances $d_{ij}=\sqrt{(x_i-x_j)^2-(y_i-y_j)^2}$ are used in all instances. In order to compare with the best results of the previous study on the conventional VRP, the shipping cost per unit distance $\gamma_{ghk}=1, \forall k\in K,\ g\in V\backslash\{o'\},\ h\in V\backslash\{o\},\ g\neq h$. The battery driving range is set to $Q=\lceil 1.2d_{max}\rceil$, where d_{max} is the maximal Euclidean distance between any two points in the network, and the construction cost per station is set to $\lceil 0.5Q\rceil$.

Based on the results of several preliminary experiments, the parameter setting is determined as follows. For the TS-MCWS method, $N_I^0=15$ and $N_J^0=50$. For the IG, the weights of allocation cost, β_1,β_2 and β_3, are set to 0.08, 0.92 and 0. For the ALNS, the percentages of customers to remove, δ_1 and δ_2, are set to 0.1 and 0.3 respectively. The parameters to avoid determinism in the worst removal and related removal operators, ρ_w and ρ_r, are both set to 10. The number of best unique solutions in the request graph removal, N_{best}, was set to 30. In the related removal operator, the weights of similarity measure $\alpha_1=0.6$ and $\alpha_2=0.4$. In the advanced worst removal operators and the advanced insertion operators, weights $\lambda_1=\lambda_3=0.9$ and $\lambda_2=\lambda_4=0.1$. The maximal number of iterations $ITER^{ALNS}$ is set as follows: if $|I|\leq 50$, $ITER^{ALNS}=40$, otherwise, $ITER^{ALNS}=50$. For the whole SIGALNS, we have used the following strategy: if $|I|\leq 50$, $ITER^{SIGALNS}=200$, else if $|I|\leq 200$, $ITER^{SIGALNS}=300$, otherwise, $ITER^{SIGALNS}=400$. Therefore, for different customer size, the number of iterations in which the historical success of each operator in ALNS are recorded is 8000, 10,000 and 15,000.

5.2. ALNS configurations

This section compares the performance of different removal and insertion operators used in ALNS. Finally, a combination of removal and insertion operators is selected based on the following criteria: the operator performs better than the average level in at least half of the test instances with regard to different evaluation indexes. When evaluating the behavior of removal operators, only one of them was used in the experiments but all insertion operators are involved. Similarly, in the comparison of insertion operators, ALNS only selected one insertion operator but chose removal operators adaptively. Because the ALNS includes random elements, each instance is solved five times using the default configuration ($ITER^{ALNS}=20, ITER^{SIGALNS}=400$) and report the average results. Tables 1–2 show a summary of this experiment.

Table 1 provides a comparison of removal operators. The column "*Param.*" lists the evaluation parameters: "*Call*" denotes the number of times a operator has been used, and "*W.*" is the mean value of the final weights. The column "*Instance*" describes the problem instance. Columns 3–12 present the computational results of all operators. Column "*Avg.*" reports the average performance. Row-wise results better than the average are indicated in bold. We see that *related removal (ReR)* performs best and the next best is *random removal (RaR)* and *single point removal (SPR)*. The *worst removal* (*BWR* and *AWR*) gives the worst performance in terms of two evaluation indexes, which is different from the conclusion given by Ropke and Pisinger [49]. Moreover, in order to improve the solution quality of ALNS, five operators are selected: according to the selection criteria, three removal operators including *random removal* (*RaR*), *related removal* (*ReR*) and *single point removal* (*SPR*) are chosen. However, because of the excellent performance of *RGR* and *BiR* with respect to the number of called times ("*Call*"),

Table 1
Performance comparison of removal operators for the instances (Set P) proposed by Augerat et al. [4].

Param.	Instance	RaR	RGR	BWR	ReR	AWR	SBR	SPR	TPR	BiR	*Avg.*
Call	P-n16-k8	**1305**	**1354**	997	**1341**	817	1079	**1352**	**1285**	**1238**	1196
	P-n21-k2	**1379**	**1320**	978	**1435**	1161	1121	**1329**	946	972	1182
	P-n40-k5	**1502**	**1479**	724	**1758**	577	1130	**1421**	1139	1189	1213
	P-n50-k7	**1650**	**1463**	232	**1730**	422	1166	**1531**	1168	**1299**	1185
W.	P-n16-k8	**8.70**	8.35	7.40	**11.52**	4.24	**10.29**	**9.41**	**8.65**	7.81	8.48
	P-n21-k2	**13.00**	9.04	8.61	**10.42**	7.13	**11.00**	**10.37**	8.94	7.53	9.56
	P-n40-k5	**10.42**	8.31	4.99	**12.37**	2.79	**11.62**	**13.34**	**13.83**	5.03	9.19
	P-n50-k7	**13.02**	**10.71**	0.09	**13.86**	1.78	7.18	**8.63**	6.02	**11.75**	8.12

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

Table 2
Performance comparison of insertion operators for the instances (Set P) proposed by Augerat et al. [4].

Param.	Instance	BGI	BR2I	BR3I	AGI	AR2I	AR3I	*Avg.*
Call	P-n16-k8	1941	1944	**2034**	1972	**2105**	**2003**	2000
	P-n21-k2	1925	**2040**	**2003**	1949	**2001**	**2081**	2000
	P-n40-k5	1652	**2266**	**2056**	1776	**2227**	**2023**	2000
	P-n50-k7	1009	**2485**	**2348**	1244	**2540**	**2374**	2000
W.	P-n16-k8	9.26	9.19	9.20	**11.02**	9.52	**9.94**	9.69
	P-n21-k2	9.47	**11.19**	**11.69**	10.57	9.27	**11.49**	10.62
	P-n40-k5	8.89	10.54	10.26	**11.47**	**13.02**	**13.73**	11.32
	P-n50-k7	3.22	**13.41**	**10.61**	4.85	**11.42**	**12.42**	9.32

request graph removal (*RGR*) and *binary removal* (*BiR*) are also used in ALNS. Similar to Table 1, Table 2 shows the performance of the insertion operators. Row-wise results better than the average are indicated in bold. We can see that the *advanced regret*-3 *insertion* (*AR3I*) performs best among all insertion operators. This indicates that the more complicated insertion operator which considers not only improving the objective function value but also maintaining solution's feasibility is better than other simpler operators. According to the selection rule depicted above, four insertion operators are chosen, including *basic regret*-2 *insertion* (*BR2I*), *basic regret*-3 *insertion* (*BR3I*), *advanced regret*-2 *insertion* (*AR2I*) and *advanced regret*-3 *insertion* (*AR3I*). However, the *regret-k insertion* is more time consuming compared to the *greedy insertion*, especially *regret*-3 *insertion* [49], which may reduce the efficiency of ALNS. In addition, the *basic greedy insertion* (*BGI*) is able to get a solution close to the solution obtained by using the *basic regret*-3 *insertion* in a short computing time. Therefore, we trade off the solution quality against the computing time by replacing the *BR3I* with the *BGI*. The configuration of selected removal and insertion operators is used in the following computational experiments.

5.3. Computational results

Based on the parameter setting and algorithmic configuration, the proposed algorithms, namely TS-MCWS and SIGALNS, are tested on different instances. Firstly, this section makes a comparison between these heuristics and the exact algorithm in CPLEX Solver 12.3 for small-size instances. Then we evaluate the performance of TS-MCWS and SIGALNS for medium- and large-size instance.

Table 3 presents the results of computational experiments on the small-size instances, which are generated from "P-n16-k8" introduced by Augerat et al. [4] in Set P and "RY-att48" proposed by Rinaldi and Yarrow [48]. Instead of using all customers in the instance, each instance only contains the last N_c nodes. For example, "P-n6-k2" presents the last 6 customers in "P-n16-k8" with two vehicles, and "RY-n10-k3" uses the last 10 nodes in "RY-att48". Each heuristic is performed five times and the best solution is reported. In Table 3, columns 1–4 provide the problem instance, the number of customers, the number of vehicles, and the final located BSSs, respectively. Columns 5 and 6 report the computational results on the extended model by CPLEX: the best objective function value found within time limitation and the computing time in seconds, respectively. Columns 7–9 present the results by TS-MCWS: the best objective value, the average computing time in seconds and the gap in percentage compared to CPLEX, respectively. Columns 10–12 report the results by SIGALNS: the best objective value, the average computing time in seconds and the optimality gap compared to CPLEX, respectively. The gaps in columns 9 and 12 are defined as (corresponding best objective value − objective value obtained by CPLEX)/objective value obtained by CPLEX. For all instances, a time limit of 10,800 s (3 h) is imposed to CPLEX, which is also the stopping criteria used by Demir et al. [17]. "*" represents feasible solutions found by CPLEX within 3 h. "#" denotes that CPLEX failed to obtain a feasible solution in 3 h. Best values are shown in boldface. As shown in Table 3, in the first three cases where CPLEX obtains an optimal solution, TS-MCWS and SIGALNS are able to find it within 2 s. When CPLEX only obtains feasible solutions starred in Table 3 for the next two instances, the heuristics can find the solutions efficiently which are close to or even better than the solutions CPLEX obtains within the time limitation (3 h). When the number of customers reaches 20 in the last instance, it is impossible for CPLEX to solve the extended model in 3 h, however, the heuristics can obtain a satisfying solution instantly. On average, in comparison with CPLEX, the ability of our algorithms to solve BSS–EV–LRP within a short computing time is better, especially SIGALNS, which performs even better than TS-MCWS. Therefore, it is reasonable to apply these two heuristics to larger instances.

The comparison results of the medium and large size instances between these two heuristics are shown in Table 4. For clarity, columns 1–3 contain, respectively, the problem instance, the number of customers in the original instances and the number of vehicles in the optimal solution. Notation $|K|+$ means that one more vehicle is required in TS-MCWS than SIGALNS. For TS-MCWS, the number of final located BSSs, the best objective value, the average objective function value, the average computing time in seconds are presented in columns 4–7. For SIGALNS, columns 8–10 are the number of final located BSSs, the best objective value and the average objective function value, respectively. The average computing time in seconds is reported in column 11. The gaps in column 12 are computed as $(Best_1 - Best_2)/Best_2$. Both the TS-MCWS and the SIGALNS have run for five times to each instances with 200 or fewer customers and three times to each instance with more than 200 customers. Best values are shown in boldface.

The results in Table 4 show that the SIGALNS outperforms TS-MCWS in most instances. For the first set of instances from "P-n16-k8" to "P-n70-k10", SIGALNS performs better in terms of the best objective function value and the computing time in seconds. The biggest gap reaches 11.79% at instance "P-n23-k8". In the second set of instances, the results show that SIGALNS is much more efficient when the number of customers increases. SIGALNS is able to obtain a better solution than TS-MCWS within a rather short time except in two cases, namely "tai100c" and "tai100d". The exception can be explained by the special customer spatial distribution which makes a great influence on the average solution quality in ALNS, as depicted by Ropke and Pisinger [49]. In the larger instances of the third data set, the solutions of TS-MCWS are better. However, the computing time of TS-MCWS is at least four times longer than SIGALNS's. On average, the SIGALNS produces a better solution than the TS-MCWS by 3.16% in less computing time. We also observe that in more than half the instances, the location strategy in the SIGALNS's solution employs at least one fewer BSSs than TS-MCWS's. Also, one can see that fewer routes are used in SIGALNS's solutions in several cases, such

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

Table 3
Results of Comparison Between CPLEX and two heuristics for the generated instances

INPUTS				CPLEX		TS-MCWS			SIGALNS		
Instance	$\|I\|$	$\|K\|$	$\|J_L\|$	Best	Time (s)	Best	Time (s)	Gap (%)	Best	Time (s)	Gap (%)
P-n6-k2	6	2	2	**426.86**	10.51	**426.86**	1.23	0.00	**426.86**	1.73	0.00
P-n7-k3	7	2	2	**428.60**	489.69	**428.60**	1.12	0.00	**428.60**	1.93	0.00
P-n8-k3	8	3	2	**597.16**	1853.04	**597.16**	1.19	0.00	**597.16**	2.00	0.00
RY-n12-k2	12	2	1	53,117.67*	10,800.00	53,403.00	1.86	0.54	**52,792.61**	1.91	−0.61
RY-n15-k3	15	2	1	53,901.78*	10,800.00	53,801.76	2.25	−0.19	**52,985.62**	1.86	−1.70
RY-n20-k4	20	3	1	#	10,800.00	65,052.45	1.89	#	**63,919.20**	2.22	#
Avg.	11.33	2.33	1.5	21,694.41	4790.65	21,731.47	1.53	0.07	**21,446.17**	1.89	−0.46

Table 4
Results of comparison between SIGALNS and TS-MCWS for instances proposed by Augerat et al. [4], Taillard [53] and Golden et al. [22]

INPUTS			TS-MCWS				SIGALNS				Gap (%)
Instance	$\|I\|$	$\|K\|$	$\|J_L\|$	$Best_1$	$Average_1$	*Time* (s)	$\|J_L\|$	$Best_2$	$Average_2$	*Time* (s)	
P-n16-k8	16	8	2	1325.81	1325.81	1.75	1	**1281.95**	1299.47	2.58	3.42
P-n19-k2	19	2	1	496.25	500.57	2.47	1	**471.39**	483.95	2.78	5.27
P-n21-k2	21	2	1	504.40	504.40	3.58	1	**478.64**	478.95	3.13	5.38
P-n23-k8	23	8+	2	1520.85	1526.34	2.47	1	**1360.51**	1436.20	3.14	11.79
P-n40-k5	40	5	2	942.89	945.95	13.93	1	**893.23**	908.05	6.18	5.56
P-n45-k5	45	5	2	953.91	964.04	19.76	2	**939.63**	950.39	7.69	1.52
P-n50-k7	50	7	2	1252.76	1261.65	25.93	2	**1196.48**	1221.07	8.52	4.70
P-n55-k8	55	7	4	1319.52	1321.23	207.66	2	**1247.10**	1268.30	20.13	5.81
P-n60-k10	60	10	4	1744.64	1752.88	113.12	3	**1684.24**	1687.65	24.50	3.59
P-n70-k10	70	10	5	1847.03	1847.77	276.43	3	**1738.98**	1764.56	35.93	6.21
tai75a	75	10	5	1945.58	1969.35	119.22	4	**1924.32**	1967.90	53.69	1.10
tai75b	75	10	4	1630.38	1635.20	109.06	3	**1607.22**	1658.93	76.73	1.44
tai75c	75	9	5	1638.84	1639.19	239.53	4	**1602.15**	1654.52	75.11	2.29
tai75d	75	9+	4	1688.79	1691.80	395.81	3	**1643.63**	1647.98	51.06	2.75
tai100a	100	12	6	2679.26	2707.76	936.17	4	**2467.90**	2509.95	118.80	8.56
tai100b	100	12+	7	2489.68	2490.66	209.33	5	**2393.34**	2431.90	134.42	4.03
tai100c	100	11	4	**1683.69**	1690.74	498.70	4	1783.45	1834.85	123.29	−5.59
tai100d	100	12	4	**1918.81**	1931.31	344.75	4	1926.96	1986.75	188.95	−0.42
tai150a	150	15	5	3859.71	3888.55	1087.29	4	**3620.34**	3722.45	329.67	6.61
tai150b	150	14	6	3470.91	3487.83	960.44	7	**3354.00**	3375.64	367.24	3.49
tai150c	150	15	5	2881.95	2883.92	1216.05	5	**2879.32**	2958.83	337.44	0.09
tai150d	150	15	6	3350.82	3357.78	2634.30	5	**3121.36**	3175.92	492.06	7.35
$GWKC_{09}$	255	14	5	**787.52**	795.54	6364.14	3	790.99	814.64	1798.23	−0.44
$GWKC_{16}$	480	38	14	**2182.47**	2185.35	62488.64	12	2359.08	2397.72	10695.97	−7.49
Avg.	101.42	10.83	4.38	1838.43	1850.51	3261.27	3.50	**1781.93**	1818.19	623.22	3.16

as “P-n23-k8”, “tai75d” and “tai100b”. To sum up, in the small-size and the medium-size cases, SIGALNS can find the better solutions within a much shorter computing time. In the large-size cases, the solutions of SIGALNS are close to those of TS-MCWS but TS-MCWS is rather time-consuming.

6. Economic analysis

The previous section evaluates the proposed algorithm performance. In this section, we analyze the economic influence of several key factors, including comparisons between the basic and extended scenarios (Section 6.1), the systematic analysis on the sensitivity of battery driving range, and the vehicle emissions reduction when electric vehicles are widely used in Section 6.2.

6.1. Comparison between basic and extended scenarios

We establish a basic integer programming model in Section 3.3 for BSS–EV–LRP based on the classical LRP model where each route can visit a node at most once. However, in the BSS–EV–LRP, it is more practical and economical to allow EVs to swap batteries at the same station more than once as illustrated in Section 3.3. This section further evaluates the benefits under the extended scenario by comparing two scenarios in instance “P-21-k2” proposed by Augerat et al. [4] in Set P. We ran SIGALNS five times under the basic and extended scenarios. The best solution and average shipping costs are reported. Fig. 4 shows their optimal solutions. It clearly shows that under the basic scenario three BSSs are constructed and each route visits a station or customer at most once. In the extended setting, there are only two located stations. The first route visits a station three times and the second route passes a station twice. Moreover, the average shipping cost under the extended scenario is 3.09% more than that under the basic scenario but the objective function value in the extended model is 3.16% less than the other. This numerical test indicates that it is beneficial for reducing the overall cost, especially station construction costs, if EVs are allowed to swap their batteries at a BSS more than once.

6.2. Cost component and environmental analysis

This section investigates the potential impact of battery technology improvement on the total operational costs and vehicle

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

emissions by increasing the driving range of EVs. Table 5 presents the computational results on the medium-size instances with 75 customers in different customer spatial distribution. With respect to different battery driving ranges, SIGALNS is performed five times for each instance with the default parameter configuration (only battery driving range varies) and the best results are reported. Columns 1 and 2 provide the name of instance and the battery driving range (150, 200 and 300 km). Columns 3–6 show the number of vehicles, the number of located BSSs, the length of the longest tour and the total times of swapping batteries in the best solution, respectively. Cost components are presented in columns 7–9, including the shipping cost (*Ship.*), the construction cost (*Cons.*) and the overall cost (*Obj.*). The gaps shown in column 10 are defined as $(Ship. - BKS_{CVRP})/BKS_{CVRP}$, where BKS_{CVRP} is the best-known solution in capacitated vehicle routing problem (CVRP) for corresponding instance. Columns 11–13 report the emission reduction efficiency (ERE) of nitrogen oxides (NO_x), particulate matters (PM), and carbon dioxide(CO_2), respectively, when EVs are put to use. The emission reduction efficiency (ERE), which measures the marginal cost of reducing the emission, is defined as $(Obj. - BKS_{CVRP})/(Er_x \times BKS_{CVRP})$, where $(Obj. - BKS_{CVRP})$ is the increased cost after adopting EVs and $(Er_x \times BKS_{CVRP})$ is the total emission of diesel vehicles in CVRP. A battery swapping station will cost $ 500,000 to build [10]. After amortizing over 25 years [18], the unit construction cost of a BSS is about $50 per day. We only use the diesel vehicles in the CVRP as a reference, because diesel engines are widely used in modern logistics. For diesel vehicles, the emission rate (*Er*) in unit of grams per kilometers of NO_x, PM and CO_2 are 17.8334, 0.1988 and 1739.8393 (g/km), respectively [14].

First of all, we target the analysis of the impact on the location strategy. Column 4 (J_L) indicates that as the battery driving range increases from 150 km to 300 km, the number of BSSs decreases to 0. Moreover, the larger battery driving range is also helpful for reducing the times of swapping batteries (see column 6). Next, we discuss the change in the vehicle routing plan. Columns 3 and 5 indicate that the number of EVs ($|K|$) and the length of the longest tour (T) are insensitive to battery driving range. Moreover, there are no significant differences in shipping cost. The reason may be that both are dictated mainly by the capacity of the vehicles in terms of goods. With the ability to revisit stations, every vehicle can fill to the max and never run out of fuel. The longest tour is thus limited by the goods-carrying capacity of the vehicle. It is worth mentioning that if the battery driving range is large enough such as 300 km, no BSSs is required and the BSS–EV–LRP becomes the classic CVRP because of the relaxation of battery driving range limitation. Under such circumstance, the best solutions of SIGALNS are equal or close to the best-known solutions in CVRP, which shows the efficiency of our algorithm to solve BSS–EV–LRP. Finally, we analyze the emission reduction efficiency. As indicated in columns 11–13, the battery driving range has a notable impact on the emission reduction efficiency and the ERE when $Q=150$ is almost twice as much as that when $Q=200$. Therefore, to make improvement on EV battery technology is significantly beneficial for lowering the cost of emission reduction.

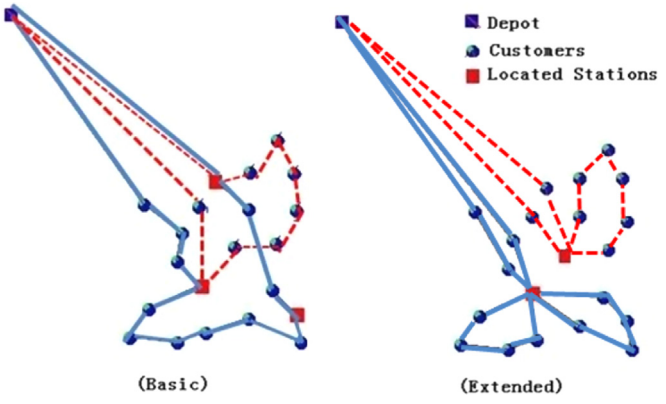

Fig. 4. Comparison between the basic and extended model by Augerat et al. [4], Set P, *'P-21-k2'* Instance.

7. Conclusion

Facility location and vehicle routing are two of the most crucial decisions in reducing logistics cost for many companies. Many studies have shown that tackling these two interrelated components separately often leads to suboptimal solutions. For a logistics corporation equipped with a fleet of electric vehicles, the location strategy for battery swap stations directly determines the battery swapping plan of each route, which affects the vehicle routes significantly. Meanwhile, the station location strongly depends on the vehicle routing plan. Therefore, this paper studied the electric vehicles battery swap stations location and routing problem (BSS–EV–LRP), which determines the station location and vehicle routing plan simultaneously under battery driving range limitation. First, a basic IP model based on classic LRP was established, followed by an extended model which allows an EV to visit a station and swap batteries more than once. We developed

Table 5
Sensitivity analysis on driving range and environmental analysis for instances proposed by Taillard [53]

INPUTS				ECONOMIC						ERE		
Instance	Q (km)	$\|K\|$	$\|J_L\|$	T (km)	S	Ship.	Cons.	Obj.	Gap (%)	NO_x/kg	PM/g	CO_2/kg
tai75a	150	10	3	229.76	7	1650.76	150	1800.76	2.00	6.320	0.567	0.065
	200	10	2	219.88	2	1618.36	100	1718.36	0.00	3.465	0.311	0.036
	300	10	0	219.88	0	1618.36	0	1618.36	0.00	0.000	0.000	0.000
tai75b	150	10	3	257.31	6	1439.37	150	1589.37	7.04	10.206	0.915	0.105
	200	10	1	269.27	2	1367.25	50	1417.25	1.68	3.028	0.272	0.031
	300	10	0	269.13	0	1366.40	0	1366.40	1.62	0.907	0.081	0.009
tai75c	150	9	3	285.44	4	1341.52	150	1491.52	3.91	8.709	0.781	0.089
	200	9	1	294.44	1	1342.78	50	1392.78	4.01	4.420	0.397	0.045
	300	9	0	285.44	0	1336.35	0	1336.35	3.51	1.969	0.177	0.020
tai75d	150	9	2	270.48	4	1442.92	100	1542.92	5.68	7.289	0.654	0.075
	200	9	1	265.66	1	1399.30	50	1449.30	2.48	3.445	0.309	0.035
	300	9	0	289.03	0	1365.42	0	1365.42	0.00	0.000	0.000	0.000

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

two heuristics for solving the problem. One heuristic TS-MCWS combines tabu search and modified Clarke–Wright saving method. Another approach SIGALSN is a four-phase algorithm, including modified sweep heuristic for initialization, iterative greedy method for location subproblem, adaptive large neighborhood search (ALNS) for routing subproblem and final improvement procedure. These heuristics were tested on three set of instances in small-, medium-, and large-size with up to 480 customers.

In the computational experiments, we compared the performance of CPLEX and the other two heuristic methods. For the small-size instances, CPLEX and two heuristics were able to find optimal solutions. When the size of instance increase, the heuristic methods were able to instantly obtain the solutions which were better than the solutions that CPLEX obtained within 3 h. We then evaluated the performance of TS-MCWS and SIGALNS. The results show that SIGALNS outperformed TS-MCWS on most of instances. In the small-size and the medium-size cases, SIGALNS was able to find better solutions within a much shorter computing time. In the large-size cases, the solutions of SIGALNS were close to TS-MCWS's but TS-MCWS was rather time-consuming. On average, SIGALNS produced a better solution than the TS-MCWS by 3.16% in less time. In addition, fewer BSSs and vehicles were required in SIGALNS's solution. Thereafter, comparison between the basic and extended scenarios indicated that it was helpful for reducing overall cost, especially BSS construction cost, to allow EVs to swap their battery at a BSS more than once. At last, in the cost component and environmental analysis, though the increase in battery driving range made no significant influence on the number of vehicles needed and shipping cost, larger driving range could result in fewer BSSs required and lower construction cost. Furthermore, the environmental analysis concluded that the improvement of battery driving range and the infrastructure construction of BSSs are very beneficial for lowering the cost of emission reduction.

This work suggests several improvements and further research directions. It is meaningful to extend the BSS–EV–LRP by considering the battery inventory cost and customer service time window. The problem with multiple depots or battery swap stations capacity limitation is also worth studying.

Acknowledgments

The author thanks the editor-in-chief, the guest editor, and the anonymous reviewers for offering extraordinarily constructive feedback in the review process. The authors gratefully acknowledge the financial support of the National Nature Science Foundation of China under Grant (71320107001) and the Fundamental Research Funds for the Central Universities (HUST: 2013QN101).

References

[1] Adulyasak Y, Cordeau JF, Jans R. Optimization-based adaptive large neighborhood search for the production routing problem. Transp Sci 2014;48(1):20–45.
[2] Albareda-Sambola M, Daz JA, Fernndez E. A compact model and tight bounds for a combined location-routing problem. Comput Oper Res 2005;32(3):407–28.
[3] Albareda-Sambola M, Fernndez E, Laporte G. Heuristic and lower bound for a stochastic location-routing problem. Eur J Oper Res 2007;179(3):940–55.
[4] Augerat P, Belenguer J, Benavent E, Corberan A, Naddef D, Rinaldi G. Computational results with a branch and cut code for the capacitated vehicle routing problem. Technical Report 949-M, Universite Joseph Fourier, Grenoble; 1995.
[5] AutoblogGreen. DHL orders at least 50 Renault EVs by 2015; 2012. Available at ⟨http://green.autoblog.com/2012/10/15/dhl-orders-at-least-50-renault-evs-by-2015/⟩.
[6] AutoblogGreen. UPS puts 100 electric trucks into service in central California; 2013. Available at ⟨http://green.autoblog.com/2013/02/08/ups-puts-100-electric-trucks-into-service-in-central-california/⟩.
[7] Baldacci R, Mingozzi A, Calvo RW. An exact method for the capacitated location-routing problem. Oper Res 2011;59(5):1284–96.
[8] Belenguer J-M, Benavent E, Prins C, Prodhon C, Calvo RW. A branch-and-cut method for the capacitated location-routing problem. Comput Oper Res 2011;38(6):931–41.
[9] Berman O, Larson RC, Fouska N. Optimal location of discretionary service facilities. Transp Sci 1992;26(3):201–11.
[10] Business Insider. The Cost Of A Better Place Battery Swapping Station: $ 500,000; 2009. Available at ⟨http://www.businessinsider.com/the-cost-of-a-better-place-battery-swapping-station-500000-2009-4⟩.
[11] Chan Y, Carter WB, Burnes MD. A multiple-depot, multiple-vehicle, location-routing problem with stochastically processed demands. Comput Oper Res 2001;28(8):803–26.
[12] Chang C. Business models and public policies for EV charging stations in China; 2010. Available at ⟨http://www.changce.org/attachments/559_evstation_businessmodel_changce_20101018.pdf⟩.
[13] Clarke G, Wright JW. Scheduling of vehicles from a central depot to a number of delivery points. Oper Res 1964;12(4):568–81.
[14] Cohen JT, Hammitt JK, Levy JI. Fuels for urban transit buses: a cost-effectiveness analysis. Environ Sci Technol 2003;37:1477–84.
[15] Conrad RG, Figliozzi MA. The recharging vehicle routing problem. In: Proceedings of the industrial engineering research conference. Reno, NV; 2011.
[16] Contardo C, Hemmelmayr V, Crainic TG. Lower and upper bounds for the two-echelon capacitated location-routing problem. Comput Oper Res 2012;39(12):3185–99.
[17] Demir E, Bektas T, Laporte G. An adaptive large neighborhood search heuristic for the pollution-routing problem. Eur J Oper Res 2012;223:346–59.
[18] Department of the Treasury. How to depreciate property; 2013. Available at ⟨http://www.irs.gov/pub/irs-pdf/p946.pdf⟩.
[19] Electric Cars Report. Deutsche post DHL introduces electric vehicles to Bonn; 2013. Available at ⟨http://electriccarsreport.com/2013/05/deutsche-post-dhl-introduces-electric-vehicles-to-bonn/⟩.
[20] Erdoan S, Miller-Hooks E. A green vehicle routing problem. Transp Res Part E: Logist Transp Rev 2012;48(1):100–14.
[21] Gillett BE, Miller LR. A heuristic algorithm for the vehicle-dispatch problem. Oper Res 1974;22(2):340–9.
[22] Golden BL, Wasil EA, Kelly JP, Chao I-M. Metaheuristics in vehicle routing. In: Crainic G, Laporte G, editors. Fleet management and logistics. Boston: Kluwer; 1998. p. 33–56.
[23] Hemmelmayr VC, Cordeau J-F, Crainic TG. An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics. Comput Oper Res 2012;39:3215–28.
[24] Hodgson MJ. A flow capturing location–allocation model. Geograph Anal 1990;22(3):270–9.
[25] Kang JE, Recker W. Strategic hydrogen fuel cell charging station location analysis with scheduling and routing considerations of individual vehicles. Working Paper. University of California, Irvine, USA; 2012.
[26] Kim L. Namsan electric bus cruises into international lime-light; 2011. Available at ⟨http://www.cnngo.com/seoul/life/namsan-electric-bus-cruises-limelight-262862⟩.
[27] Kim JG, Kuby M. The deviation-flow refueling location model for optimizing a network of refueling stations. Int J Hydrog Energy 2012;37(6):5406–20.
[28] Kuby M, Lim S. The flow-refueling location problem for alternative-fuel vehicles. Socio-Econ Plan Sci 2005;39(2):125–45.
[29] Kuby M, Lim S. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. Netw Spat Econ 2007;7(2):129–52.
[30] Laporte G, Nobert Y. An exact algorithm for minimizing routing and operating costs in depot location. Eur J Oper Res 1981;6(2):224–6.
[31] Laporte G, Nobert Y, Arpin D. An exact algorithm for solving a capacitated location-routing problem. Ann Oper Res 1986;6(9):291–310.
[32] Laporte G, Nobert Y, Taillefer S. Solving a family of multi-depot vehicle routing and location-routing problems. Transp Sci 1988;22(3):161–72.
[33] Laporte G, Musmanno R, Vocaturo F. An adaptive large neighbourhood search heuristic for the capacitated arc-routing problem with stochastic demands. Transp Sci 2010;44:125–35.
[34] Li JQ. Transit bus scheduling with limited energy. Transp Sci 2013. Articles in Advance. http://dx.doi.org/10.1287/trsc.2013.0468.
[35] Lin S, Lee Z, Ying K, Lee C. Applying hybrid meta-heuristics for capacitated vehicle routing problem. Expert Syst Appl 2009;36(2):1505–12.
[36] Lysgaard J, Letchford AN, Eglese RW. A new branch-and-cut algorithm for the capacitated vehicle routing problem. Math Progr 2004;100(2):423–45.
[37] Mak HY, Rong Y, Shen ZJM. Infrastructure planning for electric vehicles with battery swapping. Manag Sci 2013;59:1557–75.
[38] Min H, Jayaraman V, Srivastava R. Combined location-routing problems: a synthesis and future research directions. Eur J Oper Res 1998;108:1–15.
[39] Mirchandani P, Madsen OBG, Adler J. Scheduling and location issues in transforming service fleet vehicles to electric vehicles. In: 12th international conference on advanced systems for public transport, Santiago, Chile; 2012.
[40] Muller LF, Spoorendonk S, Pisinger D. A hybrid adaptive large neighborhood search heuristic for lot-sizing with setup times. Eur J Oper Res 2012;218(3):614–23.
[41] Nagy G, Salhi S. Location-routing: issues, models and methods. Eur J Oper Res 2007;177(2):649–72.
[43] Perl J. Unified warehouse location-routing analysis. Technical Report, Northwestern University, Illinois; 1983.
[44] Pisinger D, Ropke S. A general heuristic for vehicle routing problems. Comput Oper Res 2007;34(8):2403–35.

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003

[45] Post and Parcel. DHL launches major test for electric delivery vehicles in Manhattan; 2011. Available at ⟨http://postandparcel.info/37383/news/dhl-launches-major-test-for-electric-delivery-vehicles-in-manhattan/⟩.
[46] Prins C, Prodhon C, Ruiz A, Soriano P, Calvo RW. Solving the capacitated location-routing problem by a cooperative Lagrangian relaxation-granular tabu search heuristic. Transp Sci 2007;41(4):470–83.
[47] Ribeiro GM, Laporte G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. Comput Oper Res 2012;39 (3):728–35.
[48] Rinaldi G, Yarrow LA. Optimizing a 48-city traveling salesman problem: a case study in combinatorial problem solving. doctoral dissertation. New York University, Graduate School of Business Administration; 1985.
[49] Ropke S, Pisinger D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. Transp Sci 2006;40 (4):455–72.
[50] Schneider M, Stenger A, Goeke D. The electric vehicle routing problem with time windows and recharging stations. Technical Report, University of Kaiserslautern, Kaiserslautern; 2012.
[51] Shaw P. A new local search algorithm providing high quality solutions to vehicle routing problems. Technical Report. University of Strathclyde, Glasgow; 1997.
[52] Solomon MM. Algorithms for the vehicle routing and scheduling problems with time window constraints. Oper Res 1987;35(2):254–65.
[53] Taillard E. Benchmarks for basic scheduling problems. Eur J Oper Res 1993;64 (2):278–85.
[54] Tuzun D, Burke KI. A two-phase tabu search approach to the location routing problem. Eur J Oper Res 1999;116:87–99.
[55] United Nations Environment Programme. UNEP environmental assessment Expo 2010 Shanghai, China; 2009. Available at ⟨http://www.unep.org/pdf/SHANGHAI_REPORT_FullReport.pdf⟩.
[56] Upchurch C, Kuby M, Lim S. A model for location of capacitated alternative fuel stations. Geograph Anal 2009;41(1):85–106.
[57] Wang YW. An optimal location choice model for recreation-oriented scooter recharge stations. Transp Res Part D: Transp Environ 2007;12(3):231–7.
[58] Wang YW. Locating battery exchange stations to serve tourism transport: a note. Transp Res Part D: Transp Environ 2008;13(3):193–7.
[59] Wu TH, Low CW, Bai J. Heuristic solutions to multi-depot location-routing problems. Comput Oper Res 2002;29(10):1393–415.
[60] Zarandi MHM, Hemmati A, Davari S. The multi-depot capacitated location-routing problem with fuzzy travel times. Expert Syst Appl 2011;38(8):10075–84.
[61] Zarandi MHM, Hemmati A, Davari S, Turksen IB. Capacitated location-routing problem with time windows under uncertainty. Knowl-based Syst 2013;37 (4):480–9.

Please cite this article as: Yang J, Sun H. Battery swap station location-routing problem with capacitated electric vehicles. Computers and Operations Research (2014), http://dx.doi.org/10.1016/j.cor.2014.07.003