ARTICLE INFO
Article ID: 14-13-01-0009
© 2024 The Authors
doi:10.4271/14-13-01-0009

Energy-Efficient Dispatching of Battery Electric Truck Fleets with Backhauls and Time Windows

Dongbo Peng,[1] Guoyuan Wu,[2] and Kanok Boriboonsomsin[2]

[1]University of California Riverside, Department of Electric and Computer Engineering, USA
[2]University of California Riverside, Center for Environmental Research & Technology, USA

Abstract

The adoption of battery electric trucks (BETs) as a replacement for diesel trucks has potential to significantly reduce greenhouse gas emissions from the freight transportation sector. However, BETs have shorter driving range and lower payload capacity, which need to be taken into account when dispatching them. This article addresses the energy-efficient dispatching of BET fleets, considering backhauls and time windows. To optimize vehicle utilization, customers are categorized into two groups: linehaul customers requiring deliveries, where the deliveries need to be made following the last-in-first-out principle, and backhaul customers requiring pickups. The objective is to determine a set of energy-efficient routes that integrate both linehaul and backhaul customers while considering factors such as limited driving range, payload capacity of BETs, and the possibility of en route recharging. We formulate the problem as a mixed-integer linear programming model and propose an algorithm that combines adaptive large neighborhood search and simulated annealing metaheuristics to solve it. The effectiveness of the proposed strategy is demonstrated through extensive experiments using a real-world case study from a logistics company in Southern California. The results indicate that the proposed strategy leads to a significant reduction in total energy consumption compared to the baseline strategy, ranging from 11% to 40%, while maintaining reasonable computational time. In addition, the proposed strategy provides solutions that are better than or comparable with those obtained by other metaheuristics. This research contributes to the development of sustainable transportation solutions in the freight sector by providing a novel approach for dispatching BET fleets. The findings emphasize the potential of deploying BETs to achieve energy savings and advance the goal of green logistics.

History

Received: 26 Jun 2023
Revised: 27 Sep 2023
Accepted: 04 Dec 2023
e-Available: 22 Dec 2023

Keywords

Sustainable freight transportation, Pick-up and delivery, En route charging, Meta-heuristics

Citation

Peng, D., Wu, G., and Boriboonsomsin, K., "Energy-Efficient Dispatching of Battery Electric Truck Fleets with Backhauls and Time Windows," *SAE Int. J. Elect. Veh.* 13(1):107–123, 2024, doi:10.4271/14-13-01-0009.

ISSN: 2691-3747
e-ISSN: 2691-3755

This article is part of a focus issue on Eco-Driving of Connected Electrified Vehicles in Intelligent Transportation Systems.

1. Introduction

In recent decades, there has been an increase in greenhouse gas (GHG) emissions due to human activities [1]. Among various sectors, transportation is the leading contributor to GHG emissions. In the US, the transportation sector accounted for 28% of total GHG emissions in 2021, primarily attributed to the combustion of fossil fuels in vehicular transportation and goods movements [2]. Specifically, heavy-duty vehicles contribute approximately 23% of GHG emissions within the transportation sector, placing them as the second-largest contributor, surpassed only by light-duty vehicles [3]. To address the environmental and climate challenges posed by the freight transportation sector, efforts have been directed toward promoting the accelerated adoption of clean heavy-duty vehicles, including zero-emission heavy-duty trucks, in the near future [4].

In recent years, the emergence of battery electric trucks (BETs) has shown promise in reducing GHG emissions in the transportation sector. However, the widespread adoption of BETs presents several challenges due to technological limitations of BETs as compared to conventional diesel trucks. These limitations include shorter driving range, longer refueling (recharging) time, and lower payload capacity. Furthermore, the availability of public charging stations for BETs is currently very limited.

In this article, we consider a real-world fleet dispatching of a logistics company in Southern California, who aims to transition to a 100% BET fleet. Given the technological limitations of BETs, as mentioned earlier, the company will need to adapt their fleet dispatching strategies, together with BET recharging strategies. Therefore, our study focuses on the BET dispatching problem with the objective of minimizing total energy consumption from dispatching and recharging operations. This problem can be seen as an extension of the green vehicle routing problem (GVRP) proposed by Erdoğan and Miller-Hooks [5].

To accurately estimate BET energy consumption, we propose a detailed energy consumption model for BETs in this study. This model takes into account various factors, such as varying cargo weight and driving speed. Additionally, we explore an en route partial recharging policy for the BET fleet, allowing BETs to recharge at any available charging stations along their routes. This approach is more practical in real-world scenarios compared to full recharging restrictions since a well-designed recharging schedule can reduce charging time and ensure adherence to customer time windows [6].

In addition to the requirements in the GVRP and the classic Electric Vehicle Routing Problem with Time Windows (EVRP-TW) [7], the BET dispatching problem under investigation incorporates practical constraints that are crucial for real-world applicability. Specifically, we consider precedence and time window constraints to enhance the efficiency of BET utilization. In this context, the truck driver follows a precedence strategy whereby deliveries are performed upon departure from the depot, followed by pickups during the backhaul trip. This strategy is known as Vehicle Routing Problem with Backhauls (VRP-B) [8]. The customers in the problem are divided into two distinct sets: linehaul customers who require deliveries, where the deliveries need to be made following the last-in-first-out principle, and backhaul customers who require pickups. There are several benefits for this approach from an economic and practical perspective. First, it reduces the occurrence of empty vehicle trips, thereby reducing energy consumption and mitigating the negative environmental impact associated with transportation [9]. Second, since trailers are often rear-loaded, the appropriate arrangement of cargo load or pallets can be established when departing from the depot to visit linehaul customers. This avoids the need for cargo rearrangement at each customer location [8].

In this article, we present a novel approach to address the energy-efficient BET dispatching problem with backhauls and time windows. We develop a mixed-integer linear programming (MILP) model specifically tailored for this problem, which we refer to as the Green Vehicle Routing Problem with Backhauls, Partial Recharging, and Time Windows (GVRP-B-PR-TW). To efficiently solve this model, we design and implement an adaptive large neighborhood search (ALNS)-based metaheuristic algorithm. Extensive experimentation is conducted using realistic test instances, allowing us to evaluate the effectiveness and performance of our proposed approach.

This research makes significant contributions in the following areas:

1. *Formulating an Energy-Efficient BET Dispatching Problem*: This study addresses the need for sustainable freight transportation by formulating an energy-efficient BET dispatching problem. The formulation extends the classic GVRP-TW by incorporating an energy consumption model, a partial recharging scheme, and a backhaul strategy. This comprehensive problem formulation enables the consideration of real-world operational constraints and optimization objectives.
2. *Development of a Computationally Efficient Approach*: To efficiently solve the proposed GVRP-B-PR-TW, an effective ALNS-based metaheuristic algorithm is employed. This algorithm utilizes a combination of exploration and exploitation strategies to find high-quality solutions for the dispatching problem. By employing the ALNS framework, the algorithm can efficiently explore the solution space and provide near-optimal or improved solutions.
3. *Validation with Real-World BET Fleet Dispatching Data*: The proposed dispatching strategy is rigorously evaluated using real-world BET fleet dispatching data, including orders, itineraries, and routes. The experimental results demonstrate the efficacy of the proposed approach in significantly reducing total energy consumption when compared to the baseline strategy implemented in the real world. In addition, the proposed ALNS framework provides solutions that are better than or comparable with those obtained by other metaheuristics algorithms for our case study. Lastly, the effect of battery capacities is further evaluated.

The remainder of this article is organized as follows. Section 2 provides a brief literature review on the GVRP and the VRP-B. In Section 3, we present the BET energy consumption model and introduce a MILP model specifically designed for the BET dispatching problem. The methodology of the ALNS metaheuristics algorithm to solve the proposed problem is detailed in Section 4. To assess the performance of the proposed solution, Section 5 presents an evaluation based on a real-world case study. Finally, in Section 6, we conclude the article and discuss future directions for research.

2. Related Literature

The past two decades have witnessed a growing research interest in solving the GVRP and its variants. In this section, we first briefly review GVRP-related models, which mainly focuses on alleviating the negative impacts on the environment when designing a routing strategy for electric vehicles (EVs) and considering the energy consumption models. Second, we discuss the related works focusing on the VRP-B, where the goal is to design the most-effective routes that satisfy the requirements of both linehaul and backhaul customers.

2.1. Green Vehicle Routing Problem

In [5], GVRP was introduced as a variant of VRPs, involving alternative and greener fuel vehicles (e.g., EVs, biodiesel, ethanol, etc.) that have limited travel range and need to be recharged en route. The goal of this problem is to decrease the total energy consumption in fleet operations. The authors proposed two heuristic approaches, involving a modified Clarke and Wright saving (MCMS) and a density-based clustering algorithm to solve this problem. In this study, the alternative fuel vehicles (AFVs) are incapacitated and the time windows and partial refueling are not considered. Extend to the GVRP [5], Schneider et al. [7] further investigated the Electric Vehicle Routing Problem with Time Windows (EVRP-TW), considering time windows, limited driving range, and freight capacities during route planning. They developed a hybrid metaheuristic framework that combines a variable neighborhood search (VNS) and a tabu search (TS) to solve the EVRP-TW, and demonstrated the performance of the proposed algorithm based on the GVRP instances [5] and the Solomon VRP-TW instances [10]. Furthermore, Keskin and Çatay [6] studied en route partial recharge strategies for the EV fleet, which relax the full recharge restriction in [7]. The authors proposed an ALNS algorithm to solve the problem and demonstrated the efficiency of the partial recharging option.

However, the energy consumption of EVs is nonlinearly in realistic scenario [11]. Many studies have been proposed to investigate the EVRP (e.g., [5, 6, 7]) considering a more realistic energy consumption model of EV, where the energy consumption is nonlinearly proportional to travel distance. For instance, Goeke and Schneider [12] utilized a realistic energy consumption model considering speed, terrain gradient, and cargo payload in the routing strategy for both ECVs and ICCVs and evaluated the cost of battery replacement as one of the objective functions. In this study, the charging time varies depending on the battery state-of-charge (SOC) when the EV arrives at the CSs. Zhang et al. [13] introduced the electric vehicle routing problem with recharging stations for minimizing energy consumption, where the energy consumption model of ECV is similar to [11, 12]. The authors developed an ant colony (AC)-based metaheuristic algorithm and to address the proposed problem. In [14], Macrina et al. studied a GVRP with a mixed fleet composed with electrical and conventional vehicles. The authors proposed more realistic energy consumption models for both EVs and internal combustion engine vehicles (ICEVs) and investigated the effects of acceleration and deceleration on energy consumption. Recently, Yu et al. [15] proposed an ALNS framework embedded with a dynamic programming procedure to address the green mixed fleet VRP and evaluate its potential for carbon emission reduction.

Few studies have considered a practical constraint, the pick-up and delivery sequence, in GVRPs. In [16], Granada-Echeverri et al. introduced an EVRP with backhauls (EVRP-B) in logistics distribution, and an iterated local search heuristic algorithm was developed to solve the EVRP-B. Yang et al. [17] proposed an EVRP-TW with mixed backhauls and recharging strategies. The authors constructed a multidimensional network to represent the transportation process, and an augmented Lagrangian relaxation model is provided to solve the vehicle routing and recharging strategies. Moreover, Xiao et al. [18] proposed a diversity-enhanced memetic algorithm (DEMA) to solve the EVRP-TW with mixed backhauls.

Nevertheless, from the economic and sustainable perspectives, cargo weight and visit sequence significantly impact energy consumption and transportation efficiency. In this study, we introduce the GVRP-B-PR-TW, which accounts for energy consumption characteristics of BETs, backhaul strategy, partial recharging policy, and time windows. In addition, the impact of cargo weight on electricity consumption is also considered. We summarize the main differences between our work and the related literature in Table 1.

2.2. Vehicle Routing Problem with Backhauls

The second strand of relevant literature focuses on solving the VRP-B, where the goal is to find a cost-effective routing plan for the linehaul and backhaul customers. The VPR-B was first introduced by Deif and Bodin [19] in the literature, an extension of the classic VRP. In this problem, the linehaul customers who request deliveries are first visited, and followed by the backhaul customers who request pickups. In [8], Toth and Vigo introduced a mixed-integer programming formulation for a general VRP-B and developed an exact branch-and-bound algorithm to address the proposed problem.

Similarly, Mingozzi et al. [20] developed an exact method based on the set partitioning model to solve the vehicle routing

TABLE 1 Characteristic of GVRPs addressed in this article and related literature.

			Customer			Fleet composition		Recharging strategy		
Reference	Solution approach	Energy cost function	Linehaul	Backhaul	Mixed backhaul	Homogeneous	Heterogeneous	Full	Partial	Time windows
Erdoğan and Miller-Hooks (2012)	MCWS	Linear	•			•		•		•
Schneider et al. (2014)	VNS+TS	Linear	•			•		•		•
Goeke and Schneider (2015)	ALNS+LS	Nonlinear	•				•	•		•
Keskin and Çatay (2016)	ALNS+SA	Linear	•			•		•		•
Zhang et al. (2018)	AC+ALNS	Nonlinear	•			•		•		•
Macrina et al. (2019)	ILS	Linear	•				•	•		•
Granada-Echeverri et al. (2020)	ILS	Linear		•		•		•		
Yang et al. (2021)	ADMM	Linear			•	•			•	
Yu et al. (2021)	ALNS+DP	Nonlinear	•				•		•	•
Xiao et al. (2023)	DEMA	Linear			•	•		•		•
This article	*ALNS+SA*	*Nonlinear*		•		•			•	•

Note: AC, Ant Colony; ADMM, Alternating Direction Method of Multipliers; ALNS, Adaptive Large Neighborhood Search; DEMA, Diversity-enhanced Memetic Algorithm; DP, Dynamic Programming; ILS, Iterated Local Search; LS, Local Search; MCWA, Modified Clarke and Wright Savings; SA, Simulated Annealing; TS, Tabu Search; VNS, Variable Neighborhood Search.

problem with backhauls. This approach computed a valid lower bound for the optimal solution by solving the dual of the LP-relaxation of its integer program. Compared with the classic VRP-B, Ropke and Pisinger [21] introduced a rich pick-up and delivery problem with time windows and proposed a unified heuristic solver to solve the VRP-B and its variants. They proposed an improved version of the large neighborhood search (LNS) algorithm to solve different types of VRP-B.

The VRP-B has many realistic applications in transportation and logistics. For example, Salhi et al. [22] presented the fleet size and mixed vehicle routing problem with backhauls, where the goal is to minimize the total travel cost, including the fixed cost of different types of vehicles and related traveling cost. In [23], Chávez et al. presented a multi-depot vehicle routing problem with backhauls, where the vehicle fleet is collecting after the delivering process. The authors proposed a multi-optimization approach based on an AC heuristic algorithm to solve the proposed problem with respect to three objectives of travel distance, travel time, and total energy consumption. Recently, Lin et al. [24] investigated several real-world operational constraints for the VRP-B, including last-in, first-out precedence, order payload, vehicle types, and operation times. Inspired by a practical application with a major grocery chain, the authors formulated the proposed problem as a mixed-integer programming model. A greedy randomized adaptive search procedure (GRASP)-based algorithm was developed to solve the proposed problem. In [25], Yu et al. introduced VRP with simultaneous pick-up and delivery and occasional drivers. A simulated annealing (SA)-based heuristic algorithm that incorporates a set of neighborhood operators is proposed to solve the model.

A general overview of the literature for the VRP-B was provided by [26] with a focus on the computational performance of exact algorithms and heuristic algorithms. In [27], the authors provided a comprehensive review for the existing literature on VRP-Bs, including the mathematical formulation, solution methodology, and industrial applications. Recently, from a sustainability perspective, Santos et al. [28] conducted a survey on the VRP-B, focusing on the effectiveness of the economic and sustainable concerns.

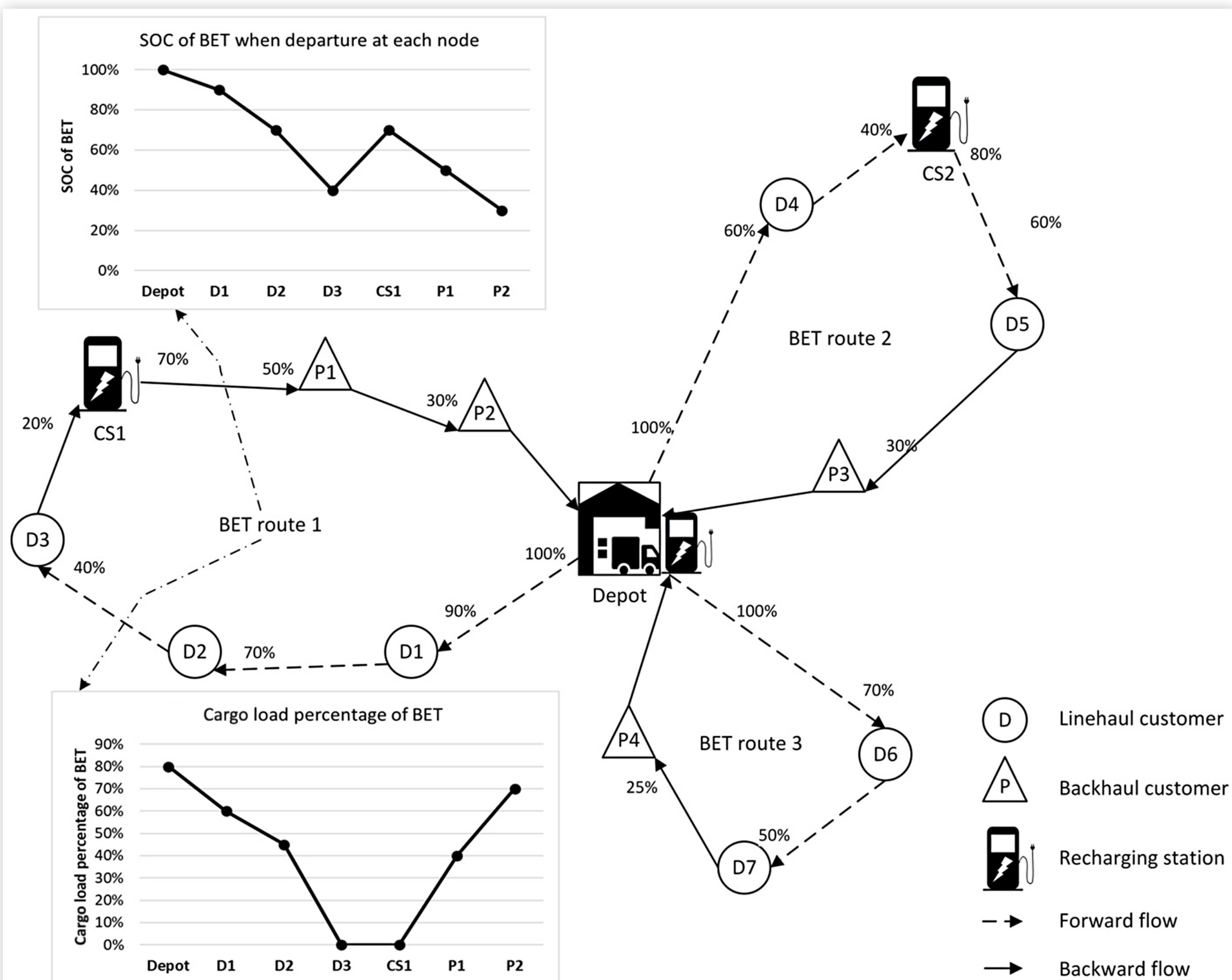
3. Problem Description and Formulation

The proposed EVRP-B-PR-TW concerns a set of clustered customers with known delivery types, demand, address, appointment time windows, and service times. The dispatching center receives the customers' information, then makes a precedence dispatching strategy and recharging schedule for a homogeneous fleet of BETs with limited cargo payload and battery capacity. To serve the integrated inbound–outbound logistics, the BET is fully recharged before departure at the depot, and is partially/fully loaded with the requested delivery orders for the linehaul customers. When arranging those delivery orders into the trailer, the ones that will be delivered later are loaded first and the ones that will be delivered sooner are loaded at the end. This last-in-first-out arrangement can reduce the cargo handling time when making delivery to the linehaul customers on the outbound trip. On the return trip to the depot, the BET visits the backhaul customers and loads the pick-up orders (within its capacity). As a result, a possible reduction in the number of BETs and the total travel distance for the BET fleet leads to less energy consumption.

Figure 1 graphically describes our dispatching problem involving seven linehaul customers (D1–D7) who request deliveries, four backhaul customers (P1–P4) who request pickups, two charging stations (CS1 and CS2), and a depot where the BET fleet is fully charged when starting the daily operation. The percentage value in each arc/flow shows the battery SOC when the BET arrives at each vertex or departs from the CS. It should be noted that the BET can visit the CS at most once and the maximum recharging time is one hour.

The BET dispatching problem can be defined as a complete directed graph $\mathcal{G}=\left(\mathcal{N}'_{O,D}\cup\mathcal{R},\mathcal{A}\right)$, where $\mathcal{N}'_{O,D}$ denotes the vertices including all customers' nodes $\mathcal{N}$ and depot (O, D), and $\mathcal{R}$ represents recharging stations. A set of customers $\mathcal{N}$ can be partitioned into two groups $\{L, B\}$, where the set $L = (1, 2, \ldots, n)$ represents the linehaul customers, and the set $B = (n + 1, n + 2, \ldots, n + m)$ denotes the backhaul customers. Each customer $i \in \mathcal{N}$ has an assigned delivery type with demand q_i (positive if pick-up, negative if delivery), service time s_i, and time window $[e_i, l_i]$, where e_i and l_i denotes the earliest and latest service starting times, respectively. All BETs should departure from the depot O and returns at D, with a maximum load capacity C and a battery capacity Q.

FIGURE 1 Illustration of a feasible solution to the BET dispatching problem with backhauling.



Extend the integer linear programming formulation and notation of [8], we define the set of arc $\mathcal{A} = A_1 \cup A_2 \cup A_3$. Specifically, let $A_1 = \{(i,j) \in A : i \in L \cup O, j \in L \cup \mathcal{R}\}$ denote all forward flows (i.e., from the depot to the linehaul vertices), $A_2 = \{(i,j) \in A : i \in B \cup \mathcal{R}, j \in B \cup D\}$ represent the backward flows include all backhauling vertices, and the interface arc $A_3 = \{(i,j) \in A : i \in L \cup \mathcal{R}, j \in B \cup D\}$. Each arc (i,j) is associated with a travel distance d_{ij} and travel time t_{ij}. We define $\Delta_i^+ = \{j : (i,j) \in \mathcal{A}, i \in \bar{V}\}$, which denotes the forward of i, and $\Delta_i^- = \{j : (j,i) \in \mathcal{A}, i \in \bar{V}\}$, which denotes the backward of i.

3.1. The Energy Consumption Model for the BETs

The battery energy consumption in our model is calculated as follows. First, we determine the mechanical power P_M using the model presented in [29]. In mechanical power, it determines the energy consumption based on factors such as travel distance, vehicle weight, speed, acceleration, and the like. Second, the mechanical power P_M is translated into the electric power P_E that the electric motor needs to provide the required amount of mechanical power. Third, the electric energy needed by the electric motor is converted to the amount of power that has been taken from the battery P_B based on the battery discharge efficiency [13].

The mechanical power P_M of BET is needed to overcome rolling, drag and wind resistance, and gravitational force as well as to enable the acceleration (a). With the rolling resistance factor c_r, the total vehicle mass M and the gravitational constant g, and the gradient angle θ, the rolling resistance F_r can be determined as

$$F_r = c_r \cdot M \cdot g \cdot \cos(\theta) \qquad \text{Eq. (1)}$$

For the aerodynamic resistance F_a, we can combine the speed v, the aerodynamic drag coefficient c_d, ρ_a the air density, and the frontal area A. Then, the aerodynamic resistance can be calculated by

$$F_a = \frac{1}{2} \cdot \rho_a \cdot A \cdot c_d \cdot v^2 \qquad \text{Eq. (2)}$$

Therefore, the total mechanical power P_M is:

$$P_M = \left(M \cdot a + \frac{1}{2} \cdot c_d \cdot \rho \cdot A \cdot v^2 + M \cdot g \cdot \sin(\theta) + c_r \cdot M \cdot g \cdot \cos(\theta) \right) \cdot v \qquad \text{Eq. (3)}$$

To calculate the mechanical power requirement P_E of the BET on the linked level arc (i, j), we use the model presented in [11, 13], which is a linear function of vehicle weight and a quadratic form of vehicle speed. To simplify the problem, we assume the total weight $M = w + C_{ij}$ where w and C_{ij} represent the curb weight and load carried by the BET, respectively, the distance for the arc (i, j) is represented as d_{ij}. Then, the mechanical energy required by the BET is shown as follows:

$$P_{E_{ij}} \approx P_M \left(d_{ij} / v_{ij} \right) = \frac{P_M d_{ij}}{v_{ij}} = \alpha_{ij} \left(w + C_{ij} \right) d_{ij} + \beta v_{ij}^2 d_{ij} \qquad \text{Eq. (4)}$$

where $\alpha_{ij} = a + g \sin \theta_{ij} + gC_r \cos \theta_{ij}$ is an arc-specific constance and $\beta = 0.5 C_d A \rho$ is a vehicle-specific constant. In our problem, we assume the vehicle speed is constant, and the result is represented by kilowatt hour (kWh).

Hence, to compute the battery power demand on a graph, the motor efficiency (eff_m) and battery discharging efficiency (eff_d) of a BET are taken into consideration in the model. The electric energy consumption E_{ij} for traveling this arc can be calculated by:

$$E_{ij} = \frac{P_{Eij}}{eff_d \cdot eff_m} = \frac{\left[\alpha_{ij} \left(w + C_{ij} \right) d_{ij} + \beta v_{ij}^2 d_{ij} \right]}{eff_d \cdot eff_m} \qquad \text{Eq. (5)}$$

3.2. Mathematical Formulation

The BET dispatching problem extends the classic EVRP-TW, and the goal is to minimize the total energy consumption to serve a set of customers, considering precedence constraint (last in, first out), cargo payload capacity, battery capacity, and partial en route recharging policy. The variables and parameters used in this study are summarized in Table 2.

Thus, the BET dispatching problem can be formulated as a mixed-integer program as follows:

$$\min \sum_{i \in \mathcal{N}_O' \cup \mathcal{R},\, j \in \mathcal{N}_D' \cup \mathcal{R},\ i \neq j} E_{ij} x_{ij} \qquad \text{Eq. (6)}$$

Subject to:

Demand and flow balance constraints

$$\sum_{i \in \Delta_j^-} x_{ij} = 1, \ \ j \in \mathcal{N} \cup \mathcal{R} \qquad \text{Eq. (7)}$$

$$\sum_{j \in \Delta_i^+} x_{ij} = 1, \ \ i \in \mathcal{N} \cup \mathcal{R} \qquad \text{Eq. (8)}$$

$$\sum_{j \in \mathcal{N}'_D \cup \mathcal{R}, i \neq j} x_{ij} - x_{ji} = 0, \forall i \in \mathcal{N}'_O \cup \mathcal{R} \qquad \text{Eq. (9)}$$

$$\sum_{i \in \Delta_O^-} x_{ij} = K \qquad \text{Eq. (10)}$$

$$\sum_{i \in \Delta_D^-} x_{ij} = K \qquad \text{Eq. (11)}$$

Vehicle constraints:

$$y_O = Q, \quad \forall j \in \mathcal{N} \cup \mathcal{R} \qquad \text{Eq. (12)}$$

TABLE 2 Variable definitions.

Variable	Description
m_B	Set of BETs available at the depot
$\mathcal{N}$	Sets of customer vertices
L	Sets of linehaul customer vertices
B	Sets of backhaul customer vertices
K	A total number of BETs in operation
$\mathcal{R}$	Recharging station(s)
r	Recharging rate
d_{ij}	Distance between vertices i and j
t_{ij}	Travel time between vertices i and j
E_{ij}	Energy consumption between vertices i and j
T_O	Earliest departure time
T_D	Latest return time
C	Cargo payload capacity
Q	BET maximum battery capacity
q_i	Demand at vertex (positive if pick-up, negative if drop-off)
e_i	Earliest start of service time at vertex i
l_i	Latest start of service time at vertex i
s_i	Service time at vertex i
τ_i	Decision variable specifying the time of arrival at vertex i
k_i	Decision variable specifying the visit to recharging station vertex i. 0 if customer, 1 if charging station
u_i	Decision variable specifying the remain cargo on arrival at vertex i
y_i	Current SOC for BET v_B when arrive at vertex i
Y_i	Finish charging SOC for BET v_B at vertex i
x_{ij}	Binary decision variable. 0 if the route from i to j is not visited by BET v_B, 1 otherwise

$$\sum_{j\in\mathcal{N}\cup\mathcal{R}} x_{Oj} \le m_B \quad \text{Eq. (13)}$$

Recharging visit constraints:

$$\sum_{j\in(D\cup\mathcal{N}\cup\mathcal{R})} x_{ij} \le 1, \quad \forall i\in\mathcal{R} \quad \text{Eq. (14)}$$

Recharging time with time window:

$$T_O \le \left(t_{ij} + (1-k_i)s_i + k_i\cdot\frac{Y_i - y_i}{r}\right)x_{ij} \le T_D, \quad \forall i\in O\cup\mathcal{N}\cup\mathcal{R}, \quad j\in(D\cup\mathcal{N}\cup\mathcal{R}), \quad i\ne j \quad \text{Eq. (15)}$$

$$0 < Y_i \le Min\{60\cdot r, 80\%\cdot Q\}, \quad \forall i\in\mathcal{R} \quad \text{Eq. (16)}$$

Time window constraints:

$$\tau_i + (s_i + t_{ij})x_{ij} - l_0(1-x_{ij}) \le \tau_j \quad \forall i\in O\cup\mathcal{N}\cup\mathcal{R}, \quad \forall j\in j\in(D\cup\mathcal{N}\cup\mathcal{R}), i\ne j \quad \text{Eq. (17)}$$

$$e_i \le \tau_i \le l_i, \quad \forall i\in\mathcal{N}'_{O,D} \quad \text{Eq. (18)}$$

Demand constraints:

$$0 \le u_o \le C \quad \text{Eq. (19)}$$

$$0 \le u_j \le u_i - q_i x_{ij} + C(1-x_{ij}) \quad \forall i\in O\cup\mathcal{N}\cup\mathcal{R}, \quad \forall j\in D\cup\mathcal{N}\cup\mathcal{R}, \quad i\ne j \quad \text{Eq. (20)}$$

Battery recharging constraints:

$$0 \le \left((1-k_i)\cdot y_i + k_i\cdot Y_i - h\cdot E_{ij}\right) x_{ij} \le Q, \forall i\in\mathcal{N}'_O\cup\mathcal{R}, j\in\mathcal{N}'_D\cup\mathcal{R}, \ i\ne j \quad \text{Eq. (21)}$$

Binary decision variable:

$$x_{ij}\in\{0,1\}, \quad \forall i,j\in\mathcal{N}'_{O,D}, \quad i\ne j \quad \text{Eq. (22)}$$

The objective function of minimizing the total energy consumption is defined in (6). Constraints (7), (10), and (8), (11) impose the indegree and outdegree constraints for the customers nodes and the charging stations. Constraint (9) define the flow conservation constraints. Constraint (12) ensures the BET is fully charged when departure at the depot. Constraint (13) ensures that the operating BETs do not exceed the maximum number of BETs available at the depot. Constraints (14)–(16) define the en route recharging policy, each BET is allowed to recharge at most once, considering one-hour maximum recharging time as full charge may slowly. Constraints (17) and (18) define the arrival time at each vertex should satisfy the time windows. Constraints (19) and (20) represent the capacity of each BET does not exceed the maximum cargo payload when visiting each vertex, for both inbound and outbound trips. Constraint (21) restricts the battery SOC is non-negative when dispatching. Finally, condition (22) defines the binary decision variables.

4. Methodology

In this section, we have developed an ALNS algorithm to solve the proposed BET dispatching problem. The ALNS first introduced by [30], which extended the LNS [31], has been demonstrated as a successful approach capable of solving the standard vehicle routing problem with pick-up and delivery [21], the electric vehicle routing problem with backhaul and time windows [6], pollution routing problem [29], mixed fleet vehicle routing problem [12], and the like.

The entire framework of the ALNS algorithm is described in Algorithm 1. The algorithm is initialized with an energy-feasible solution generated by a constructive heuristic,

ALGORITHM 1 Overview of the ALNS framework.

Input: An initial feasible solution S^{inital} generated by initialization phase;
Output: a set of near-optimal solution S^b
1: $S^c \leftarrow$ $generate\_inital\_solution(\,)$
2: $S^b = S^c$; $\omega^- = (1, \ldots, 1)$; $\omega^+ = (1, \ldots, 1)$
3: **while** specified maximum runtime T is not reached **do**
4:     {select a destroy operator $\zeta^- \in \Gamma^-$ by $P(\omega_i^-)$}
5:     destroy current solution S^c with destroy operator ζ^-
6:     $S^{c\prime} \leftarrow$ DestroyedOperator(S^c)
7:     {select a repair operator $\zeta^+ \in \Gamma^+$ by $P(\omega_i^+)$}}
8:     $S^{c\prime} \leftarrow$ RepairOperator($S^{c\prime}$)
9:         **if** accept_SA($S^{c\prime}$, S^b) **then**
10:             $S^c \leftarrow S^{c\prime}$
11   **if** $S^{c\prime}$ is better than S^b **then**
12                 $S^b \leftarrow S^{c\prime}$
13   **end if**
14:         **end if**
15:     **Update:** the weight of destroy operators ω^- and repair operators ω^+
16: **end while**
17: **return** S^b

described in Section 4.1. In line 2, at the beginning of the main loop of ALNS improvement process, we regard an initial feasible solution S^c as the current best solution S^b. Additionally, we initialize the weight vectors (ω^- and ω^+) for the destroy and repair operators (detailed in Section 4.3), denoted by Γ^- and Γ^+, respectively. The main loop of the ALNS improvement (lines 3–16) then starts and searches for a near-optimal solution S^b until the termination criteria is met. To iteratively improve the best solution, a set of destroy operators Γ^- and repair operators Γ^+ are used to modify the initial current solution S^c and obtain a new solution $S^{c'}$. Section 4.2 describes the solution improvement process in detail, including the framework of ALNS, acceptance criteria, and termination criteria.

4.1. Generation of Initial Solution

A greedy constructive heuristic is used to generate an initial feasible solution to the ALNS, similar to the method implemented by [32]. Algorithm 2 shows the pseudocode for generating an initial feasible solution.

Initially, a candidate customer is randomly chosen from a set of unvisited customers and inserted into the current route. Next, during each iteration, an appropriate customer is inserted into the current BET route greedily, leading to the minimum increase in the total energy consumption. When the current route becomes energy infeasible, we attempt to insert a potential recharging schedule from a set of available

ALGORITHM 2 Construction of initial feasible solution.

Input: A set of customers $\mathcal{N} = \{L, B\}$, recharging stations $\mathcal{R}$;
Output: an energy-feasible solution S^{inital}
$\mathcal{N}^{Unvisited} \leftarrow \mathcal{N}$
Current route for BET $K_i \in m_B, i \in \{1, 2, \ldots, m_B\}$
while unvisited customer $\mathcal{N}^{Unvisited} \neq \emptyset$ **do**
    if a new route K_i starts **then**
        Random sample a candidate customer p from $\mathcal{N}^{Unvisited}$ and insert to K_i
        Update: $\mathcal{N}^{Unvisited} \leftarrow \mathcal{N}^{Unvisited} \setminus p$
    $c, i \leftarrow$ Find a candidate customer c and an insertion position i that generates the lowest cost $f(S^{inital})$
    Update: $\mathcal{N}^{Unvisited} \leftarrow \mathcal{N}^{Unvisited} \setminus c$
    if c cannot be inserted in the current route K_i since energy infeasible **then**
        Find an insertion position and a recharging station from $\mathcal{R}$ that generates the lowest cost $f(S^{inital})$
    else
        Update: start new route for BET $K_i, i = i + 1$
end while
return S^{inital}

charging stations $\mathcal{R}$. Therefore, the remaining unvisited customers may be allowed to insert into the current route.

The current BET route terminates when there are no vertices that can be visited by the current BET due to the battery capacity, time windows, or cargo capacity violation. At this point, a new BET route starts following the same processes as described above, until all customers have been visited.

4.2. ALNS Improvement

We next detail the ALNS procedure for solving the BET dispatching problem, which includes a set of removal operators $\Gamma^- = \left\{\zeta_1^-, \zeta_2^-, \ldots, \zeta_{ND}^-\right\}$ to destroy vertices (i.e., a few customers and CSs), and a set of repair operators $\Gamma^+ = \left\{\zeta_1^+, \zeta_2^+, \ldots, \zeta_{NR}^+\right\}$ (detailed in Section 4.3) to reinsert unvisited customers or CSs, where ND and NR represent the number of destroy and repair operators, respectively. As the feasible initial solution can be obtained in Section 4.1, we define it as the current feasible solution S^c. Then, the ALNS procedure iteratively improves S^c until the termination criteria meet.

At each iteration, a removal operator $\zeta^- \in \Gamma^-$ and a reinsertion operator $\zeta^+ \in \Gamma^+$ are applied to destroy and repair the current solution S^c, respectively. Those operators are selected dynamically and adaptively based on the roulette wheel principle. To choose an operator in each iteration, we define two weight vectors, $\omega^- = \left[\omega_1^-, \omega_2^-, \ldots, \omega_{ND}^-\right]$ and $\omega^+ = \left[\omega_1^+, \omega_2^+, \ldots, \omega_{NR}^+\right]$, to store the weight of a set of destroy and repair operators, consecutively. Therefore, the probability of choosing an operator ζ can be calculated by $P^t(\omega) = \omega_i / \left(\sum_{j=1}^{|\Gamma|} \omega_j\right)$. After the current solution $S^{c'}$ is repaired, then, we obtain a new solution $S^{c'}$.

In order to overcome the local optimal results, we use a SA approach to accept or reject the new solution $S^{c'}$ generated by the ALNS algorithm. There are three circumstances in an iteration. If a new solution $S^{c'}$ has been found to be better than or equal to the global best solution S^b, we accept the new solution $S^{c'}$ as a new global best solution S^b. If the new solution is worse than the global best solution, a SA heuristic algorithm will accept the worse solution with the probability $e^{-(f(S^{c'}) - f(S^b))/T}$, where $f(X)$ is the total energy consumption of solution X, and T is the current temperature of a SA heuristic. We predefine an initial temperature T_{init}, which can be decreased at every iteration by $T = \delta T_{init}$, where the deteriorate rate $\delta \in (0, 1)$. Furthermore, in our ALNS framework, the algorithm terminates when a specified maximum runtime T is reached. In this study, we assume a maximum runtime of 1800 seconds to ensure both solution quality and time efficiency.

Moreover, in the "adaptive" mechanism, the weight vectors ρ^- and ρ^+ will be updated dynamically based on the quality of new solution $S^{c'}$. A score variable $\psi = [\varphi_1, \varphi_2, \varphi_3, \varphi_4]$ is employed to assess the performance of the ALNS improvement. For example, a score φ_1 denotes that a new solution S^b has been found. Similarly, a score φ_2 is obtained when an improved solution is found. φ_3 indicates when a new solution has been accepted by a SA heuristic, while the score φ_4 is used when the solution is rejected. At the end of each iteration, the weight vector updates by $\omega_i = \lambda\omega_i + (1 - \lambda)\psi$, where $\lambda \in (0, 1)$ is a smooth variable to control the sensitivity of the weight vector.

4.3. Destroy and Repair Operators

The number of customers/vertices n to remove is predefined by the destroy rate ϵ, where $n = \epsilon \cdot \mathcal{N}$. Then, the ALNS framework employs four removal operators to find a set of removal vertices based on the input n and store them in the removal pool $L^{removal}$. The removal heuristics are detailed as follows:

- **Random removal** randomly removes some vertices from the BET routes. The procedure terminates when n customers/vertices have been removed.
- **Random path removal** destroys an entire consecutive sub-path with n vertices.
- **Worst removal** iteratively removes n unfavorable vertices based on their cost. This operator sorts the insertion cost of all customers in descending order by calculating $c_i = f(s) - f(s_{-i})$, where s_{-i} is the route without customer i and s is the route with customer i. During each iteration, the worst vertex contributes the largest insertion cost and will be removed to the unvisited list.
- **Shaw removal** removes a set of n customers according to their similarity, which can be calculated by the relatedness function

$$\Lambda(i,j) = \phi_1 \frac{d_{ij}}{\max\limits_{i,j\in\mathcal{N}}\left(d_{ij}\right)} + \phi_2\left|e_i - e_j\right| + \phi_3 \frac{\left|q_i - q_j\right|}{\max\limits_{i\in\mathcal{N}}\left(q_i\right) - \min\limits_{i\in\mathcal{N}}\left(q_i\right)},$$

  where the weight vector $\phi = (\phi_1, \phi_2, \phi_3)$ is used to normalize the relatedness function, d_{ij} represents the distance between customers i and j, $|e_i - e_j|$ is the absolute difference between their arrival time, and $|q_i - q_j|$ is the absolute difference of their demand. At the beginning of using the Shaw removal algorithm, a customer $i \in \mathcal{N}$ is randomly selected as a candidate customer to be removed, and we calculate the most related customer $j \in \mathcal{N} \setminus i$. The customer with the highest similarity to i is the one with the smallest value of $\Lambda(i, j)$. Next, we calculate the most similarly customer and remove it by evaluate relatedness with j. Finally, this operator terminates until n vertices have been removed.

After n vertices have been removed from a solution, the repair heuristics are employed to rebuild a new solution by inserting the removed n vertices into the incomplete solution. During the reconstruction phase, we use three insertion operators to find new routes or recharging schemes:

- **Greedy insertion** iteratively conducts a series of insertions by selecting the best option. At each iteration, the operator selects one unassigned customer from the removal pool $L^{removal}$. Then, it assesses the cost function to determine whether the current insertion yields the minimum cost. This insertion process continues until all unvisited customers have been chosen.

- **Regret insertion** selects the customer with the highest difference between the cost of the first and kth best insertion and inserts it into its optimal position. The regret-k value is calculated by $reg_{i,k} = \Delta f(i, pos_{i,1}) - \Delta f(i, pos_{i,k})$, where $\Delta f(i, pos_{i,1})$ represents the cost improvement generated by the best insertion and $\Delta f(i, pos_{i,k})$ denotes the cost improvement generated by the kth best insertion. At each iteration, the operator finds the kth best insertion for customer i, which generates the highest regret-k value. This approach avoids the myopic behavior of the greedy insertion algorithm by not necessarily selecting the task with the lowest cost. In this study, we have implemented the regret-2 insertion method.
- **Greedy insertion with charging stations** was introduced in [14], a variation of the greedy insertion operator was introduced to handle energy constraints in BET routes that include CSs. This operator extends the general greedy insertion approach, which assumes that BETs do not visit en route CSs. Initially, this operator inserts unvisited customers until the battery SOC violation. Then, it computes a near-optimal charging scheme to minimize the deviation from the original BET route, allowing additional unvisited customers to be inserted. However, if a feasible charging scheme cannot be found in the current solution, the operator will terminate the insertion process after adding the customers.

5. Case Study: A Real-World BET Fleet Dispatching Problem in Southern California

This section presents the results of our numerical experiments using a real-world case study from a logistics company in Southern California. The goal is to find an energy-efficient dispatching strategy and recharging scheme for the BET fleet. To evaluate the performance of our proposed strategy, we compare our results with the historical dispatching data. The mathematical model in Section 3 is implemented in Python 3.9, and all experiments are conducted on a server with 32 GB RAM.

5.1. Data Description

Four instances ranging from 47 to 90 customers were used to evaluate the proposed strategy. These instances were generated from a real-world dataset, representing typical one-day movements of a heavy-duty diesel truck fleet that operated in the Riverside County and San Bernardino County of California. Specifically, from the truck fleet perspective, each truck has a historical dispatching data file that contains the tractor ID, delivery and pick-up time stamps (from departure to termination), service time windows, cargo weight information, service addresses, and global positioning systems (GPS) logs. From the customer perspective, the historical data contains the customer ID, service types (delivery or pick-up), address, longitude, latitude, demands, service time, and time windows.

To assess the proposed dispatching problem, we create four test instances using data from 266 customer orders, which were fulfilled by 23 trucks. These instances include geographic coordinates of customer locations as well as information on the delivery types, required demands, time windows, and service times. We randomly designate five customer locations where a recharging station is equipped in their parking lot. It is worth noting that the BET has the flexibility to visit any of the charging stations during operations if required. Table 3 provides a summary of the characteristics of the four instances.

To generate accurate distance and travel duration matrices for the truck routes between customer locations, we utilized the Direction Service Application Programming Interface (DSAPI) provided by OpenRouteService [33]. The DSAPI takes into consideration various real-world factors such as the actual road network, speed limits, and restricted roads applicable to heavy-duty trucks. By utilizing the DSAPI, we obtain distance values that are more realistic and relevant compared to using simple Euclidean distances. However, for the purpose of simplifying the dispatching problem in our study, we do not incorporate real-time traffic conditions. While traffic conditions play a crucial role in route optimization, for the scope of this research, we focus on other significant aspects and do not consider the dynamic traffic conditions.

5.2. Problem Variables and Parameter Tuning

In the numerical study, we use the problem parameter settings presented in Table 4 based on a real-world scenario. The total operation time is limited to 8 hours, including driving, idling when recharging, and service time. In our study, we assume a set of homogenous BETs in the fleet, with either short-range battery capacity (300 kWh) or long-range battery capacity (452 kWh), to evaluate the effect of driving range.

TABLE 3 Summary of dataset characteristics.

Instance	# of Customers	# of Linehauls	# of Backhauls	CSs
BETVRPB1	47	33	14	5
BETVRPB2	58	26	32	5
BETVRPB3	71	39	32	5
BETVRPB4	90	54	36	5

TABLE 4 Summary of problem parameters.

Notation	Description	Value
A	Frontal surface area of a BET [m^2]	5
C	Maximum BET cargo capacity [34] [lb]	37,000
Q	Maximum BET battery capacity [kWh]	{300, 452}
eff_m	Motor efficiency [13]	0.80
eff_d	Discharging efficiency [13]	0.90
c_r	Unitless rolling resistance	0.01
c_d	Coefficient of rolling drag	0.7
w	Vehicle curb weight [lb]	8,000
g	Gravitational constant [m/s^2]	9.81
ρ_a	Air density [km/m^3]	1.2041
θ	Road angle	0°
a	Acceleration [m/s^2]	0
ν	Vehicle speed [mph]	68
s	Loading/unloading time [hour]	(0, 2]
$[T_O, T_D]$	Working hour	[8 am, 4 pm]
r	Recharging rate [kWh/min]	3.96

A parameter tuning process has been conducted using instance BETVRPB2 with a short-range BET fleet. First, we follow a fair parameter tuning strategy by an ad hoc trial-and-error phase conducted in [30]. We predefine a set of initial parameters while developing the ALNS framework. The list of initial parameters and the considered parameter settings are shown in Table 5. This set of parameters is improved by allowing one parameter to take different values while the rest are fixed. Each parameter setting is restarted six times, and the parameter showing the lowest average cost (in terms of average deviation from the best observation) is chosen. This process is repeated until all parameters have been tuned.

Moreover, we calibrate the parameters of the SA heuristic, as they play a critical role in quantifying the performance improvement of a new solution. In this article, this calibration mainly focuses on the initial temperature and deterioration rate by implementing a 2^k factory design. Those parameters control the level of diversification when converging to the solution. Figure 2 shows the calibration results based on four parameter combinations (i.e., from p1 to p4). The final value of parameter setting shown in Table 5 is a combination leading to the lowest objective value.

TABLE 5 Summary of parameters in the experiment.

Variable	Candidate value	Final value
Score vector $\psi = [\omega_1, \omega_2, \omega_3, \omega_4]$	[15, 9, 4, 3], **[18, 10, 5, 2]**	[15, 9, 4, 3]
Decay parameter λ	**0.8**, 0.85	0.8
Destroy percentage ϵ	**35%**, 38%	38%,
Number of removal vertices	$\lfloor 0.35\mathcal{N} \rfloor$, $\lfloor 0.38\mathcal{N} \rfloor$	$\lfloor 0.38\mathcal{N} \rfloor$
Shaw removal weight vector $\phi = (\phi_1, \phi_2, \phi_3)$	**[0.5, 0.25, 0.25]**, [0.5, 0.30, 0.30]	[0.5, 0.25, 0.25]
SA initial temperature T_{init}	10, **20**	10
SA end temperature	**0.5**, 0.8	0.5
SA deterioration rate δ	0.99800, **0.99991**	0.99800

Bold values indicate the initial values while developing the ALNS framework.

5.3. Performance in Real-World Instances

In order to assess the performance of our BET dispatching strategy, we apply the ALNS algorithm to solve the generated real-world instances described in Section 5.1. We compare the results with a baseline dispatching strategy from the logistics company. The baseline strategy is provided by a routing solver in the company, which has been implemented in real-world freight operations. To make a fair comparison between the baseline strategy and the proposed dispatching strategy, we presume all historical movements were served by a BET fleet and estimate the total energy consumption by the objective function (6) for the historical iterations using the same distance matrices. Table 6 summarizes the historical iterations as the baseline in our case study.

Using the problem parameter settings shown in Table 4, we conducted 10 runs and recorded the best solution for each run. To assess the effect of battery capacity on total energy consumption, we conduct two case studies, each representing a different level of battery capacity. The first case study involves short-range BETs fitted with a 300 kWh battery, while the second case study involves long-range BETs equipped with a larger 452 kWh battery.

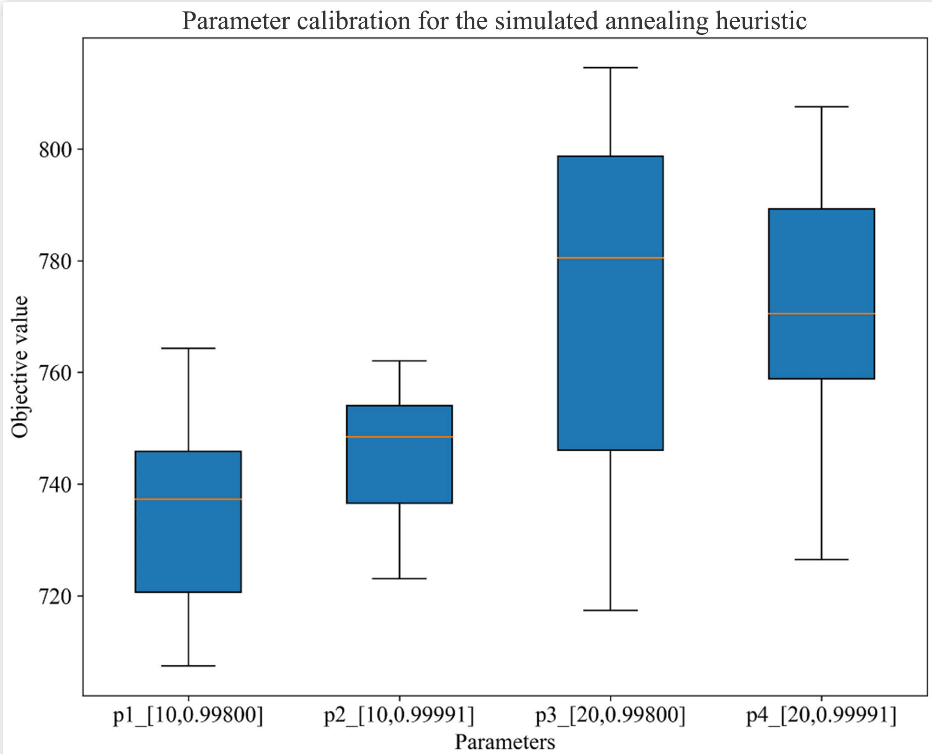
The results show that our dispatching strategy is able to solve the BET dispatching problem for all generated instances efficiently. We compare our strategy with the baseline strategy using the relative percentage deviation ($RPD_{a,b,c}$) with respect to (a) total energy consumption, (b) total vehicle miles traveled, and (c) total travel time. The formula used to calculate the $RPD_{a,b,c}$ is shown as follows:

$$RPD_{a,b,c} = \frac{C_{a,b,c}(hist) - C_{a,b,c}(opt)}{C_{a,b,c}(hist)} \times 100\%,$$

where $C_{a,b,c}(hist)$ denotes the historical cost and $C_{a,b,c}(opt)$ denotes the solutions obtained from the dispatching strategy.

As demonstrated in Tables 7 and 8, the proposed strategy can reduce total energy consumption compared with the baseline strategy. The reduction in total energy consumption ranges from 11% to 40% across the different instances. The columns "Total_dist" and "Total_time" provide information on the total vehicle miles traveled and total travel time, respectively, achieved through the implementation of energy-efficient routes. It is noteworthy that when optimizing for energy minimization, the total energy consumption can be reduced by 27% and 28%, respectively, with short-range battery and long-range battery. This reduction

FIGURE 2 The parameter calibration for the SA heuristic.



	Initial temperature T_{init}	**Deterioration rate δ**
p1	10	0.99800
p2	10	0.99991
p3	20	0.99800
p4	20	0.99991

in total energy consumption does not necessarily align proportionally with the reduction in total vehicle miles traveled. The discrepancy may be attributed to the distribution of cargo payload, which can impact the overall energy consumption of the BET fleet.

Figure 3 shows the results under different battery capacity BET fleets. In most instances, the long-range BET fleet exhibited greater energy savings compared to the short-range BET fleet, as there were potentially fewer detour trips required to visit charging stations. Notably, the cargo weight influences the energy consumption of BET, resulting in a non-proportional relationship between energy consumption and travel distance. For instance, in the BETVRPB2 scenario involving 58 customers, deploying a short-range BET fleet can lead to a significant reduction of 35% in total energy consumption. However, the corresponding total travel distance is reduced by a relatively smaller percentage of 23%. This observation highlights the impact of cargo weight on energy consumption and emphasizes the need to consider other factors beyond travel distance when optimizing energy efficiency.

TABLE 6 Summary of real-world historical movements

Instances	# of BETs	Total energy	Total_dist	Total_time
BETVRPB1	5	915	512	13.1
BETVRPB2	5	1094	490	13.7
BETVRPB3	5	1406	726	18.5
BETVRPB4	8	1062	657	22.7
Total	23	4477	2385	68.0

5.4. Analysis of the Solution Quality in ALNS

This section analyzes the effectiveness of considering the solution quality during the search. We assess the solution quality of the proposed ALNS framework (ALNS-SA) by comparing the solution with other general metaheuristics algorithms that are detailed as follows.

TABLE 7 Results for the BET dispatching problem with short-range battery.

Instance	# of BETs	Total energy	RPD_a	Total_dist	RPD_b	Total_time	RPD_c
BETVRPB1	5	751	18%	409	20%	11.2	15%
BETVRPB2	5	707	35%	379	23%	11.2	18%
BETVRPB3	5	845	40%	462	36%	12.6	32%
BETVRPB4	7	950	11%	584	11%	21.2	7%
Total	22	3253	27%	1834	23%	56.2	17%

TABLE 8 Results for the BET dispatching problem with long-range battery.

Instance	# of BETs	Total energy	RPD_a	Total_dist	RPD_b	Total_time	RPD_c
BETVRPB1	5	723	21%	396	23%	11.1	15%
BETVRPB2	5	701	36%	375	23%	11.2	18%
BETVRPB3	5	839	40%	457	37%	12.2	34%
BETVRPB4	7	942	11%	582	11%	21.3	6%
Total	22	3205	28%	1810	24%	55.8	18%

- ALNS without SA (ALNS-noSA): The ALNS framework is modified by adjusting the acceptance criteria, i.e., a SA heuristic. Thus, a worse solution is always declined. The parameter setting of the ALNS framework is described in Table 5.
- LNS with SA (LNS-SA): We implement the LNS framework as described in [30]. Following the simple LNS heuristics, we utilize the Shaw removal operator and the greedy insertion heuristic with charging station operator as mentioned in Section 4.3. The parameter settings of the Shaw removal operator and SA heuristic are identical to those used in ALNS-SA.

For each problem instance, we perform six replications on ALNS-SA, ALNS-noSA, and LNS-SA with the stopping criteria of maximum time limit (i.e., 1800 seconds) presented in Section 4.2. The best objective results and the average objective values are described in Table 9. Overall, the proposed ALNS-SA outperforms other general metaheuristics algorithms in terms of total energy consumption in five out of eight instances. Additionally, these results demonstrate that the SA procedure can improve the solution quality of our BET dispatching problems.

5.5. Effect of Battery Capacities on Total Energy Consumption

To investigate the impact of battery capacities on the solution of the BET dispatching problem, we conducted an experiment in the problem instance BETVPRB1, where we varied the BET battery capacities from 300 kWh to 500 kWh in 50 kWh increments. The results are presented in Figure 4.

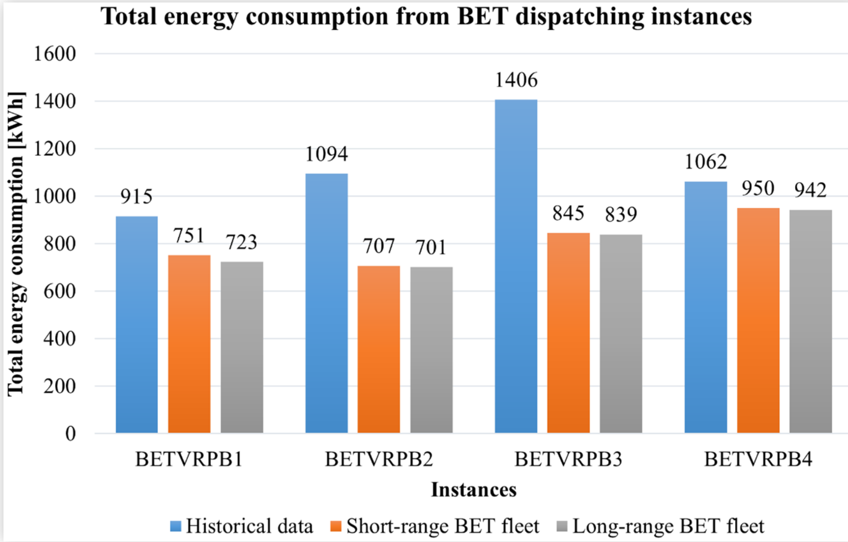
From the results, we observed that increasing the battery capacity from 300 kWh to 400 kWh in the BET fleet led to a reduction of 28 kWh in total energy consumption and a decrease of 8 miles in total vehicle distance traveled. Moreover, based on the heuristic solution, the total distance that the BET fleet travels to serve the customers can decrease to 396 miles. However, no further improvements were observed beyond a 400 kWh battery capacity in this particular scenario. This suggests that all customers can be effectively served by the BET fleet equipped with a 400 kWh battery capacity. Figure 5(a) and 5(b) show the visualization of the BET trips with 300 kWh and 400 kWh battery capacity, respectively. Each colored curve represents a different BET route. The BET fleet may detour to fulfill driving range limitations or visit charging stations. Therefore, the total energy consumption may reduce as the battery capacity increases.

6. Conclusion and Discussion

This article presents an investigation into an energy-efficient BET dispatching problem with backhauls and time windows. Building upon the classic GVRP, our study focuses on a homogeneous BET fleet with limited cargo payload and battery capacities, as well as precedence constraints for a customer set comprising linehaul customers requiring deliveries and backhaul customers requiring pickups within specific time windows. Moreover, we incorporate an en route partial recharging policy for the BET fleet, allowing partial recharging at any available charging station based on the battery SOC upon arrival.

We have formulated a MILP model to devise a dispatching strategy for a BET fleet that satisfies order types and time windows of all customers, while minimizing the total energy consumption of the BET fleet, taking into account a realistic energy consumption model specific to BETs. Notably, we have highlighted the limitations of minimizing the total travel distance alone, as it may underestimate the total energy consumption due to the influence of cargo load on BETs' energy usage.

FIGURE 3 Total energy consumption [in kWh] vs. battery capacity.



To solve the proposed problem, we have developed a metaheuristic algorithm based on the ALNS framework. In order to evaluate the performance of our dispatching strategy, we have applied the model to real-world operation data obtained from a logistics company in Southern California. The extensive experimental results demonstrate the effectiveness and efficiency of our strategy, with computational time comparable with that of the baseline strategy. Moreover, we have assessed the performance of our ALNS framework by comparing it with other metaheuristics, including standard ALNS and LNS. The results show that the proposed dispatching strategy outperforms the others in five out of eight instances.

It is important to discuss limitations of the proposed dispatching strategy. First, the fine-tuning of parameter settings plays a vital role in achieving the optimal results, although it is time-consuming. In addition, due to the problem being NP-hard, the proposed dispatching strategy becomes computationally intensive when attempting to find favorable solutions to problem instances with hundreds or more customers. Lastly, this study does not consider dynamic or uncertain factors, such as traffic condition, vehicle energy consumption, and charging behavior, which would pose more challenges when formulating an effective dispatching strategy.

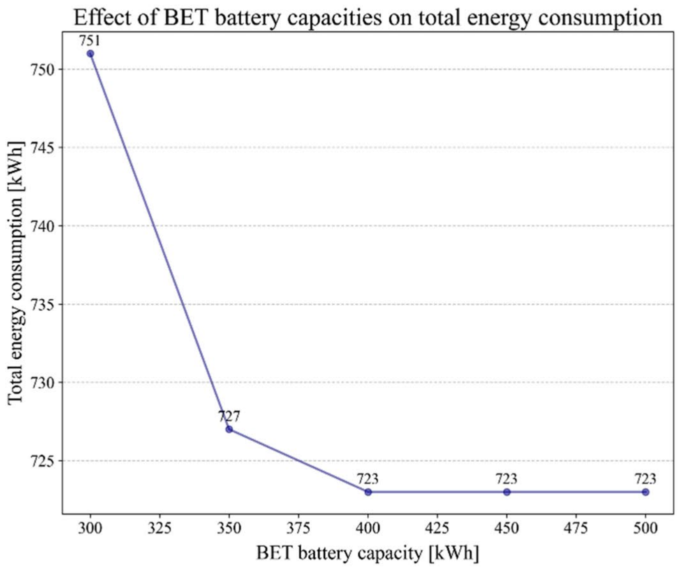
In terms of future work, there are several directions worth exploring. First, we can consider formulating additional variants of the BET dispatching problem based on real-world scenarios. For instance, incorporating variations and uncertainties into the model to account for factors such as varying service times at customer locations or dynamic traffic conditions would be a valuable extension. Second, it would be beneficial to incorporate a nonlinear charging function (e.g., [35]) into the existing model to better reflect more realistic charging rate dynamics.

TABLE 9 The results of the BET dispatching problem of ALNS-SA, ALNS-noSA, and LNS-SA.

	ALNS-SA		ALNS-noSA		LNS-SA	
Instance	Best Obj	Ave Obj	Best Obj	Ave Obj	Best Obj	Ave Obj
Scenario-I						
BETVRPB1	751	764	**747**	789	763	775
BETVRPB2	**707**	735	722	768	717	725
BETVRPB3	**845**	882	902	934	877	898
BETVRPB4	950	960	**945**	975	955	982
Total	**3253**	3341	3316	3466	3312	3380
Scenario-II						
BETVRPB1	**723**	755	755	784	738	754
BETVRPB2	**701**	761	710	738	709	737
BETVRPB3	839	860	**834**	854	846	865
BETVRPB4	**942**	968	**942**	987	954	987
Total	**3205**	3344	3241	3363	3247	3343

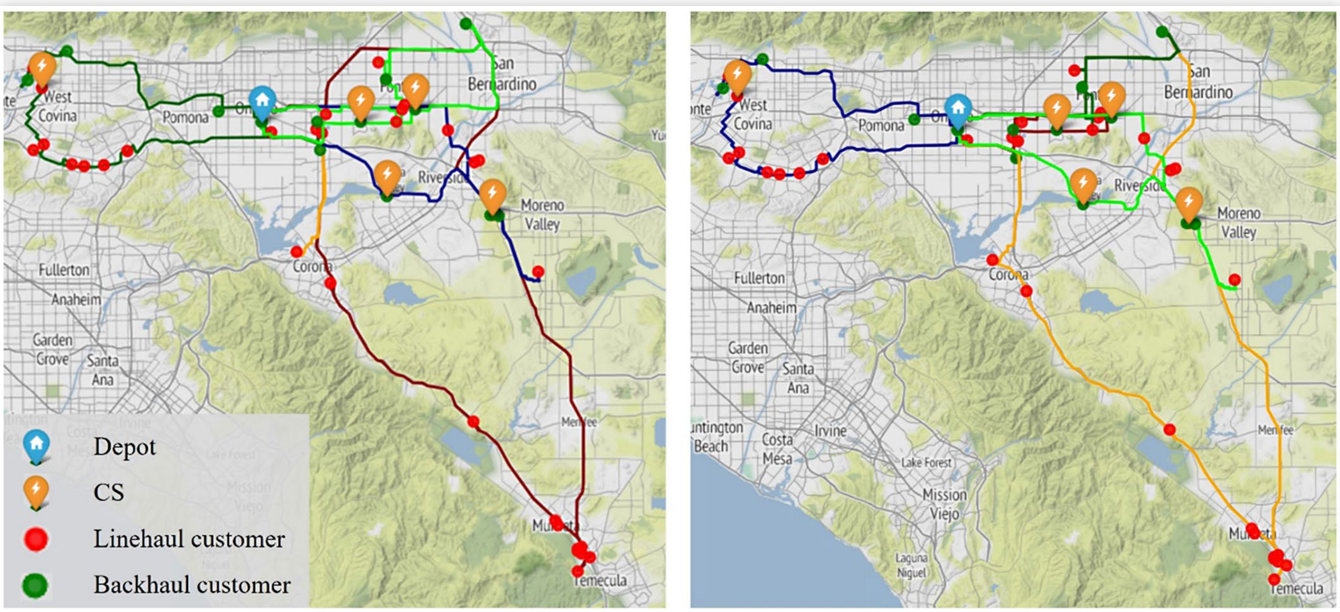
Bold values represent the best obtained objective value between ALNS-SA, ALNS-noSA, and LNS-SA.

FIGURE 4 Effect of BET battery capacities on total energy consumption.



Battery capacity (kWh)	Total energy consumption (kWh)	Total travel distance (miles)	Total travel time (hours)
300	751	409	11.2
350	727	399	11.0
400	723	396	11.1
450	723	396	11.1
500	723	396	11.1

FIGURE 5 Visualization of the dispatching solution.



(a) BET fleet with 300 kWh battery capacity

(b) BET fleet with 400 kWh battery capacity

Acknowledgement

This study was funded by a grant from the National Center for Sustainable Transportation (NCST), supported by the U.S. Department of Transportation's University Transportation Centers Program. The contents of this paper reflect the views of the authors who are responsible for the facts and the accuracy of the information presented herein. These contents are disseminated in the interest of information exchange and do not necessarily reflect the official views or policies of the U.S. Government. The U.S. Government assumes no liability for the contents or use thereof. The authors are thankful to Mr. Troy Musgrave of Dependable Highway Express for sharing sample fleet operations data for use in this study.

Contact Information

Dongbo Peng
corresponding author
dpeng017@ucr.edu

References

1. Shaftel, H., "Overview: Weather, Global Warming and Climate Change," Climate Change: Vital Signs of the Planet, accessed May 31, 2023, https://climate.nasa.gov/global-warming-vs-climate-change.
2. US EPA, "Sources of Greenhouse Gas Emissions," December 29, 2015, accessed April 17, 2023, https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions.
3. Delgado, C., "EPA Updates Emissions Standards for Heavy-Duty Vehicles for the First Time in 20 Years," Popular Science, January 16, 2023, accessed June 1, 2023, https://www.popsci.com/environment/heavy-duty-vehicles-epa-emissions/.
4. US EPA, "Clean Trucks Plan," August 5, 2021, accessed June 1, 2023, https://www.epa.gov/regulations-emissions-vehicles-and-engines/clean-trucks-plan.
5. Erdoğan, S. and Miller-Hooks, E., "A Green Vehicle Routing Problem," *Transportation Research Part E: Logistics and Transportation Review* 48, no. 1 (2012): 100-114.
6. Keskin, M. and Çatay, B., "Partial Recharge Strategies for the Electric Vehicle Routing Problem with Time Windows," *Transportation Research Part C: Emerging Technologies* 65 (2016): 111-127.
7. Schneider, M., Stenger, A., and Goeke, D., "The Electric Vehicle-Routing Problem with Time Windows and Recharging Stations," *Transportation Science* 48, no. 4 (2014): 500-520.
8. Toth, P. and Vigo, D., "An Exact Algorithm for the Vehicle Routing Problem with Backhauls," *Transportation Science* 31, no. 4 (1997): 372-385, doi:10.1287/trsc.31.4.372.
9. Pradenas, L., Oportus, B., and Parada, V., "Mitigation of Greenhouse Gas Emissions in Vehicle Routing Problems with Backhauling," *Expert Systems with Applications* 40, no. 8 (2013): 2985-2991.
10. Solomon, M.M., "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints," *Operations Research* 35, no. 2 (1987): 254-265, doi:10.1287/opre.35.2.254.
11. Lin, J., Zhou, W., and Wolfson, O., "Electric Vehicle Routing Problem," *Transportation Research Procedia* 12 (2016): 508-521.
12. Goeke, D. and Schneider, M., "Routing a Mixed Fleet of Electric and Conventional Vehicles," *European Journal of Operational Research* 245, no. 1 (2015): 81-99.
13. Zhang, S., Gajpal, Y., Appadoo, S.S., and Abdulkader, M.M.S., "Electric Vehicle Routing Problem with Recharging Stations for Minimizing Energy Consumption," *International Journal of Production Economics* 203 (2018): 404-413, doi:10.1016/j.ijpe.2018.07.016.
14. Macrina, G., Laporte, G., Guerriero, F., and Pugliese, L.D.P., "An Energy-Efficient Green-Vehicle Routing Problem with Mixed Vehicle Fleet, Partial Battery Recharging and Time Windows," *European Journal of Operational Research* 276, no. 3 (2019): 971-982.
15. Yu, V.F., Jodiawan, P., and Gunawan, A., "An Adaptive Large Neighborhood Search for the Green Mixed Fleet Vehicle Routing Problem with Realistic Energy Consumption and Partial Recharges," *Applied Soft Computing* 105 (2021): 107251.
16. Granada-Echeverri, M., Cubides, L.C., and Bustamante, J.O., "The Electric Vehicle Routing Problem with Backhauls," *International Journal of Industrial Engineering Computations* 11, no. 1 (2020): 131-152, doi:10.5267/j.ijiec.2019.6.001.
17. Yang, S., Ning, L., Tong, L.C., and Shang, P., "Optimizing Electric Vehicle Routing Problems with Mixed Backhauls and Recharging Strategies in Multi-dimensional Representation Network," *Expert Systems with Applications* 176 (2021): 114804, doi:10.1016/j.eswa.2021.114804.
18. Xiao, J., Du, J., Cao, Z., Zhang, X. et al., "A Diversity-Enhanced Memetic Algorithm for Solving Electric Vehicle Routing Problems with Time Windows and Mixed Backhauls," *Applied Soft Computing* 134 (2023): 110025, doi:10.1016/j.asoc.2023.110025.
19. Deif, I. and Bodin, L., "Extension of the Clarke and Wright Algorithm for Solving the Vehicle Routing Problem with Backhauling," in *Proceedings of the Babson Conference on Software Uses in Transportation and Logistics Management*, Babson Park, MA, 1984, 75-96.
20. Mingozzi, A., Giorgi, S., and Baldacci, R., "An Exact Method for the Vehicle Routing Problem with Backhauls," *Transportation Science* 33 (1999): 315-329, doi:10.1287/trsc.33.3.315.
21. Ropke, S. and Pisinger, D., "A Unified Heuristic for a Large Class of Vehicle Routing Problems with Backhauls," *European Journal of Operational Research* 171, no. 3 (2006): 750-775, doi:10.1016/j.ejor.2004.09.004.

22. Salhi, S., Wassan, N., and Hajarat, M., "The Fleet Size and Mix Vehicle Routing Problem with Backhauls: Formulation and Set Partitioning-Based Heuristics," *Transportation Research Part E: Logistics and Transportation Review* 56 (2013): 22-35.
23. Chávez, J., Escobar, J., and Echeverri, M., "A Multi-objective Pareto Ant Colony Algorithm for the Multi-Depot Vehicle Routing Problem with Backhauls," *International Journal of Industrial Engineering Computations* 7, no. 1 (2016): 35-48.
24. Lin, S., Bard, J.F., Jarrah, A.I., Zhang, X. et al., "Route Design for Last-In, First-Out Deliveries with Backhauling," *Transportation Research Part C: Emerging Technologies* 76 (2017): 90-117.
25. Yu, V.F., Aloina, G., Jodiawan, P., Gunawan, A. et al., "The Vehicle Routing Problem with Simultaneous Pickup and Delivery and Occasional Drivers," *Expert Systems with Applications* 214 (2023): 119118, doi:10.1016/j.eswa.2022.119118.
26. Toth, P. and Vigo, D., *The Vehicle Routing Problem* (Philadelphia, PA: SIAM, 2002)
27. Koç, Ç. and Laporte, G., "Vehicle Routing with Backhauls: Review and Research Perspectives," *Computers & Operations Research* 91 (2018): 79-91.
28. Santos, M.J., Amorim, P., Marques, A., Carvalho, A. et al., "The Vehicle Routing Problem with Backhauls towards a Sustainability Perspective: A Review," *TOP* 28, no. 2 (2020): 358-401.
29. Bektaş, T. and Laporte, G., "The Pollution-Routing Problem," *Transportation Research Part B: Methodological* 45, no. 8 (2011): 1232-1250, doi:10.1016/j.trb.2011.02.004.
30. Ropke, S. and Pisinger, D., "An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows," *Transportation Science* 40, no. 4 (2006): 455-472.
31. Shaw, P., "Using Constraint Programming and Local Search Methods to Solve Vehicle Routing Problems," in Maher, M. and Puget, J.F. (Eds), *International Conference on Principles and Practice of Constraint Programming* (Berlin: Springer, 1998), 417-431.
32. Wen, M., Linde, E., Ropke, S., Mirchandani, P. et al., "An Adaptive Large Neighborhood Search Heuristic for the Electric Vehicle Scheduling Problem," *Computers & Operations Research* 76 (2016): 73-83, doi:10.1016/j.cor.2016.06.013.
33. Openrouteservice, accessed March 15, 2022, https://openrouteservice.org/.
34. Volvo, "VNR Electric Specifications," accessed December 16, 2022, https://www.volvotrucks.us/trucks/vnr-electric/specifications/.
35. Montoya, A., Guéret, C., Mendoza, J.E., and Villegas, J.G., "The Electric Vehicle Routing Problem with Nonlinear Charging Function," *Transportation Research Part B: Methodological* 103 (2017): 87-110.