Dynamic Vehicle Routing for Battery Swapping in an Electric Bike-sharing System

Yi Yang[1], Yaoming Zhou[*1], Tanmoy Kundu[2], Suxiu Xu[3]
[1]Department of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]The Logistics Institute - Asia Pacific, National University of Singapore, Singapore
[3]School of Management, Jinan University, Guangzhou, China
*Corresponding Author
(kevinyang@sjtu.edu.cn, iezhou@sjtu.edu.cn, tlitk@nus.edu.sg, xusuxiu@jnu.edu.cn)

2021 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM) | 978-1-6654-3771-4/21/$31.00 ©2021 IEEE | DOI: 10.1109/IEEM50564.2021.9672818

Abstract - **In the daily operation of the emerging electric bike-sharing system, timely and efficient battery swapping is essential as there are no charging facilities at e-bike stations. This paper deals with the path planning of the trucks for battery swapping, considering dynamic battery demands at different stations. This problem is modeled as a vehicle routing problem with intermediate stops and soft time windows. A new solution method combining a dynamic planning strategy and a multiple neighborhood search algorithm is proposed to solve the problem. Numerical results on benchmark instances and a real-world case prove the effectiveness of our model and method.**

Keywords - **Electric bicycle sharing, battery swapping, dynamic vehicle routing, stochastic demand**

I. INTRODUCTION

Electric bike-sharing (EBS) is a new mode for short-distance urban travel, which operates similarly to a station-based bike-sharing system but with battery-powered motors. A user can only pick up and return an e-bike at fixed stations. There are no charging facilities at e-bike stations, and the operator needs to swap the batteries of e-bikes at different stations [1], [2]. The current strategy for battery swapping utilized by commercial companies is to divide a large EBS service area into smaller zones, each of which includes multiple stations and a battery warehouse. Inside each zone, operators drive trucks to deliver fully charged batteries from the warehouse to the stations and replace the low-power batteries on the e-bikes at the station there. This paper studies the path planning of these trucks for battery swapping.

This problem has been little studied in the literature, and it is generally modeled as a vehicle routing problem with soft time windows [3], [4]. Trucks start from the battery warehouse, visit a sequence of stations, and return to the battery warehouse after the fully charged batteries on the trucks are replaced with discharged batteries.

However, two important practical characteristics are ignored in the previous model. The first is that the battery swapping trucks may return to the battery warehouse for replenishment and perform multiple trips during their work-time. Besides, trucks do not always start from and end in the battery warehouse.

The second characteristic is that the battery demand of stations is dynamic. The demand of a station is defined as the number of e-bikes inside the station whose batteries need to be swapped. The power level of an e-bike has a significant impact on the user's willingness to ride it. In order to guarantee user satisfaction, operators usually require to replace not only the batteries of e-bikes that cannot be ridden at all but also the batteries on the e-bikes whose power is below a certain threshold but not totally exhausted. When users pick up or park the latter type of e-bikes at a station, the battery demand of that station will change.

In this paper, the battery warehouse is regarded as an intermediate stop, and the truck routing problem is modeled as a vehicle routing problem with intermediate stops and soft time windows. A multiple neighborhood search algorithm is proposed to solve this problem. We also consider the changes in battery demand of e-bike stations and use a dynamic strategy to cope with the changes when planning the routes of the trucks.

The remainder of this paper is organized as follows: Section II reviews the related literature. Section III provides the model. Section IV presents the dynamic planning strategy and multiple neighborhood search algorithm used to solve the problem. Section V shows the numerical results and sensitivity analysis. Section VI draws conclusions of this study and proposes directions for future research.

II. LITERATURE REVIEW

This section presents literature review on vehicle routing problems with intermediate stops (VRPIS) and vehicle routing problem with stochastic demand (VRPSD).

In VRPIS studies, an intermediate stop may be used for replenishing or disposing of goods or waste, as a refueling or recharging point, or as a place for rest periods and breaks [5]. In our problem, the intermediate stop, which is the battery warehouse, is a stop for reloading fully charged batteries. Time window constraints are considered in some VPRIS problems with stops for reloading and unloading goods. In the oil delivery problem [6], customers are divided into mandatory ones and optional ones. And some customers have delivery time restrictions that are modeled as hard time windows. Wang et al. [7] considered a petrol distribution problem where the petrol trucks return to the depot to reload. There are hard time windows at the petrol stations according to the stock level. Mat et al. [8] focused on a solid waste collection problem with a single depot and multiple disposal facilities. In their problem, the depot, customers, and the disposal facilities all have a pre-determined hard time window. However, for the routing problem of trucks in the EBS system, the time windows of the stations are more suitable to be set as soft time windows [4].

Studies on the vehicle routing problem with stochastic demands (VRPSD) mainly concentrate on designing static routes with minimum expected objective value. It is assumed that the distribution of demands is known, and recourse policies are used to handle possible failures during the execution of the routes. A combination of memetic algorithm and greedy randomized adaptive search procedure is proposed to solve the classic VRPSD problem in [9]. Erera et al. [10] considered the VRPSD with a tour duration constraint. They designed a heuristic algorithm to calculate the longest possible duration of a route when the distribution of demands is known. Then, routes are designed while making sure they never exceed the maximum tour duration. An open vehicle routing problem where the demands of customers are described by triangular fuzzy numbers are proposed in [11], which is solved by a differential evolution algorithm integrated with a stimulation algorithm for estimating the additional distance caused by route failures.

It is difficult to characterize the battery demands of e-bike stations accurately. However, the demand of a station at any time during the process can be obtained with a short delay. Therefore, instead of designing static routes, we propose a dynamic planning strategy that originates from the period re-optimization method in the dynamic vehicle routing problem (DVRP) [12]. For a comprehensive review on DVRP, the reader is referred to [13][14]. In this strategy, static problems are constructed with updated demand information, and the paths of trucks are dynamically reoptimized to reduce transportation costs.

III. MODEL

A. Problem Description

The battery swapping operation in the service zone is completed by a fleet of homogeneous trucks with a fixed maximum capacity Q. Trucks start from their origins loaded with fully charged batteries, then visit the stations in the zone to replace the low-power batteries on the e-bikes. Trucks are allowed to go back to the battery warehouse during their trips for replenishment. The entire process must be completed within the time span H. Stations have soft time windows. The lower limit of a time window is always the start time 0, and the upper limit of a time window is less than H. If a truck arrives at an e-bike station later than the upper limit of its time window, a certain penalty cost will be incurred.

Due to the influence of users' riding behavior, the demands of e-bike stations change over time. When a truck performs battery swapping at an e-bike station, the number of batteries replaced cannot exceed the demand of that station at that time.

The following assumptions are made in our study.

1. The demand at stations cannot be accurately estimated in advance.
2. The demand of an e-bike station is always less than the maximum capacity of a truck.
3. The time window, maximum penalty, and service time of each station are known and fixed.

B. Sets, Parameters, and Decision Variables

The main parameters used in this paper are as follows:

Sets

N—the set of stations requiring battery replacement.
N_0—the set of duplicate nodes of the battery warehouse.
N_+, N_-—the set of starting/ending points of the trucks.
N_{all}—$N_{\text{all}} = N \cup N_0 \cup N_+ \cup N_-$.
A—the set of directed arcs. An arc (i,j) can only starts from $i \in N_+$ and ends in $j \in N \cup N_0$, or starts from $i \in N$ and ends in $j \in N \cup N_0 \cup N_-$, or starts from $i \in N_0$ and ends in $j \in N \cup N_-$.
V—the set of trucks.

Parameters

n_{v+}, n_{v-}—the starting/ending point of truck $v \in V$.
c_{ij}—the distance between $i \in N_{\text{all}}$ and $j \in N_{\text{all}}$.
t_{ij}—the time distance between $i \in N_{\text{all}}$ and $j \in N_{\text{all}}$.
σ—the time truck needs to replenish at the warehouse.
$(0, E_i)$—the soft time window of station $i \in N$.
p_i—the maximum penalty for violating the time window of station $i \in N$.
s_i—the time trucks need to stay for at node i. For $i \in N_0$, we have $s_i = \sigma$; for $i \in N$ it equals the service time at station i; and for $i \in N_+ \cup N_-$ we have $s_i = 0$.
d_i—the demand of station i.

Decision Variables

$x_{ij}^{(v)}$—if truck $v \in V$ traverses arc $(i,j) \in A$ it equals 1, otherwise 0.
l_i—the load of truck when it leaves node i. when $i \in N_+$ it is the initial load of the truck, which is known in advance.
t_i—the time truck arrives at node $i \in N_{\text{all}}$.

C. Optimization Model

$$Min\, Z = \sum_{v \in V} \sum_{(i,j) \in A} c_{ij} * x_{ij}^{(v)} + \sum_{i \in N} p_i * \frac{m_i}{H - E_i} \tag{1}$$

$$m_i \geq t_i - E_i, i \in N \tag{2}$$

$$m_i \geq 0, i \in N \tag{3}$$

$$\sum_{(n_{v+},j) \in A} x_{n_v^+ j}^{(v)} = 1, \forall v \in V, n_{v+} \in N_+ \tag{4}$$

$$\sum_{i \in N_+ - \{n_{v+}\}} \sum_{(i,j) \in A} x_{ij}^{(v)} = 0, \forall v \in V \tag{5}$$

$$\sum_{(j,n_{v-}) \in A} x_{jn_v^-}^{(v)} = 1, \forall v \in V, n_{v-} \in N_- \tag{6}$$

$$\sum_{i \in N_- - \{n_{v-}\}} \sum_{(i,j) \in A} x_{ji}^{(v)} = 0, \forall v \in V \tag{7}$$

$$\sum_{(i,j) \in A} x_{ji}^{(v)} = \sum_{(i,j) \in A} x_{ij}^{(v)}, \forall i \in N_0 \cup N, v \in V \tag{8}$$

$$\sum_{v \in V} \sum_{(i,j) \in A} x_{ij}^{(v)} = 1, \forall i \in N \tag{9}$$

$$\sum_{v\in V}\sum_{(i,j)\in A} x_{ij}^{(v)} \leq 1, \forall i \in N_0 \quad (10)$$

$$t_i + s_i + t_{ij} - M\left(1 - \sum_{v\in V} x_{ij}^{(v)}\right) \leq t_j, \forall (i,j) \in A \quad (11)$$

$$l_i + d_j - M\left(1 - \sum_{v\in V} x_{ij}^{(v)}\right) \leq l_j, \forall (i,j) \in A, j \notin N_0 \quad (12)$$

$$0 \leq t_i \leq H, \forall i \in N_0 \cup N \cup N_+ \cup N_- \quad (13)$$

$$0 \leq l_i \leq Q, \forall i \in N \cup N_+ \cup N_- \quad (14)$$

$$l_i = 0, \forall i \in N_0 \quad (15)$$

The objective function (1) linearized with constraints (2)-(3) and an intermediate variable m_i minimizes the total travel distance and the penalty of violating the time windows of e-bike stations. Constraints (4)-(7) guarantee that trucks start from their initial positions and eventually return to their pre-determined destinations. Constraint (8) ensures that the same vehicle enters and leaves the same node. Equation (9) indicates that a station must be visited exactly once. Inequality (10) indicates that a duplicate node of the warehouse can be visited at most once. The consistency of vehicle load and arrival time at nodes is ensured by (11) and (12). Load feasibility and tour duration feasibility are enforced by (13) and (14). Constraint (15) forces the truck load at the duplicate nodes of battery warehouse to be 0.

Notice that d_i is dependent on t_i and cannot be accurately predicted, the model above thus cannot be directly solved. We can only use an estimation of demands $\widetilde{d_\iota}$ instead of the actual demands d_i. We add (16) and construct a static problem instance.

$$d_i = \widetilde{d_\iota}, \forall i \in N \quad (16)$$

IV. SOLUTION

A. Dynamic Planning Strategy

A flow chart of the dynamic planning strategy is shown in Fig. 1. The entire horizon is sliced into anticipation horizons with the same time length t_a. A static version of the problem is constructed and solved in each anticipation horizon.

At the beginning of each anticipation horizon, the best routing plan obtained in the last anticipation horizon is sent to the trucks to perform during the next t_a time. The following steps are carried out to construct a new instance.

Step 1: The stations where the trucks are in or going to at the end of this horizon are used as their new starting points. The available time of each truck is updated accordingly.
Step 2: The loads of the trucks at the new starting points are predicted based on their current loads and routing plans.
Step 3: The stations that are not served by the end of this anticipation horizon made up the new N. Their current demands are obtained, and their $\widetilde{d_\iota}$ are updated based on their demand information in the past anticipation horizons.

After these steps, a new static problem consisting of (1)-(14) is obtained. During the rest of the anticipation horizon, the problem can be solved. The solution to this problem can be directly carried out by the trucks in the next anticipation horizon.

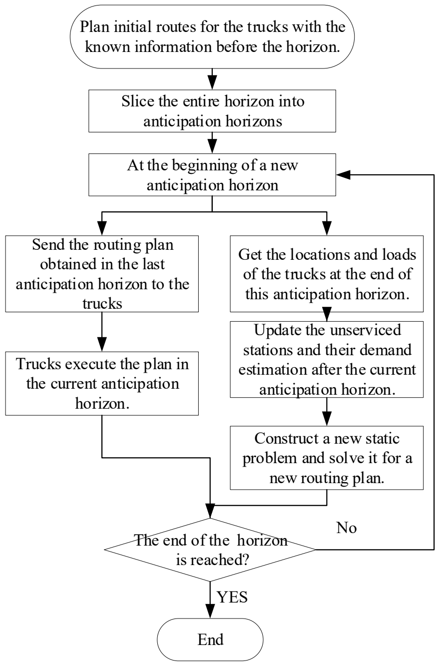

Fig. 1. Flow chart of the dynamic planning strategy

B. Multiple Neighborhood Search

The static model proposed in III is tested on gurobi. The static problem constructed in each anticipation horizon is a vehicle routing problem with intermediate stops and soft time windows, which is $\mathcal{NP}$-hard. Therefore, a multiple neighborhood search algorithm is proposed to solve the problem. A solution to the static problem is a set of tours for the trucks. The initial solution is built using an insertion algorithm. The algorithm works as follows.

Algorithm 1. Insertion
Begin:
Step 1: An empty tour is built for the unassigned truck.
Step 2: Continuously insert new stations at locations in the tour that leads to minimum cost increase without violating the duration and capacity constraints.
Step 3: If the capacity constraint prevents a new station from being inserted, try to insert a node representing the warehouse at a location on the tour that yields the smallest cost increase while preserving the duration feasibility. The tour is terminated if this attempt fails; otherwise, go back to step 2.
Step 4: If the duration constraint prohibits further insertions, the tour is also terminated.
Step 5: If all the stations are inserted or there are no other available vehicles, this algorithm terminates and outputs the tours, otherwise we select another available truck and go back to step 1.
End

The initial solution is then improved using a multiple neighborhood search similar to that in [15]. We use four neighborhoods.

Swap. Exchange two stations on the same tour or two different tours.

Relocation. Remove a station from its current location and insert it into another location on the same or a different tour.

Single-tour 2-opt. Select two stations on the same tour and reverse the part of the tour between them.

Inter-tour 2-opt. Select two stations that belong to two different tours and exchange the portions behind them.

The procedures of the multiple neighborhood search are shown below. The objective value of an infeasible solution is penalized by multiplying it with a penalty factor that is always larger than 1. When an infeasible neighbor is accepted as the new incumbent, the penalty factor is increased by a certain step; otherwise, it is reduced. The original objective value of a solution is used in the ban list.

Algorithm 2. Multiple Neighborhood Search
Begin:
Step 1: With the initial solution $p^{(\text{ini})}$, set the best feasible solution $p^{(\text{best})} = p^{(\text{ini})}$ and the current incumbent $p^{(\text{cur})} = p^{(\text{ini})}$.
Step 2: For each of the neighborhoods, repeat steps 3-7.
Step 3: Search all neighbors of $p^{(\text{cur})}$, penalize the objective value of infeasible neighbors with the penalty factor.
Step 4: Set the neighbor with the minimum objective value that is not in the ban list as the new $p^{(\text{cur})}$, if it is feasible and better than $p^{(\text{best})}$, set $p^{(\text{best})} = p^{(\text{cur})}$.
Step 5: Update the penalty factor based on the feasibility of $p^{(\text{cur})}$, update the ban list.
Step 6: The number of iterations in one neighborhood n_{niter}+1.If n_{niter} is larger than the maximum number of iterations in a neighborhood, or after a certain number of iterations, $p^{(\text{best})}$ is not improved, go to step 8, otherwise go back to step 3.
Step 7: End the current round of search with the neighborhood, reset $p^{(\text{cur})} = p^{(\text{best})}$, and reset the penalty factor.
Step 8: The number of iterations with all the neighborhoods n_{iter}+1. If n_{iter}+1 is larger than the maximum number of iterations or after a certain number of iterations, $p^{(\text{best})}$ is not improved, output $p^{(\text{best})}$ as the results; otherwise, go back to step 2.
End

V. NUMERICAL RESULTS

A. Algorithm Analysis on adjusted datasets

In order to verify the effectiveness of the multiple neighborhood search algorithm on solving the static problems, the Solomon benchmark instances [16] are used. The original dataset for vehicle routing problem with hard time windows is modified to conform to our problem.

TABLE I
COMPARISON OF THE COMPUTATIONAL RESULTS ON THE ADJUSTED SOLOMON INSTANCES

Instance	MaxV	MaxH	MNS	OR-Tools	Gap
R101.25	2	460	370.1	435.9	15.1%
R201.25	2	3 000	3 599.7	4 389.9	18.0%
C101.25	2	2 472	1 950.9	2 002.2	2.6%
C201.25	2	3 390	2 708.4	2 944.4	8.0%
RC101.25	2	480	310.8	380.4	18.3%
RC201.25	2	2 400	3 037.8	3 083.1	1.5%

We took and adjusted two instances, respectively, from the R class, C class, and RC class. The six instances are solved by our algorithm and the routing module in the optimization tool OR-Tools of Google, which is widely used in the industry. The results are shown in TABLE I. MaxV and MaxH denote the maximum number of trucks allowed and the maximum tour duration in the adjusted instances. The best cost obtained by our algorithm is smaller than that of OR-Tools on all instances, which proves the effectiveness of our algorithm.

B. Real-world Case Analysis

The real-world case analysis considers an e-bike service zone in Shanghai of HelloBike. The zone consists of 1 warehouse and 40 stations, as shown in Fig. 2. Data concerning our problem are from the company. We set: $H = 1\text{h}$, $|V| = 2$, and $Q = 200$.

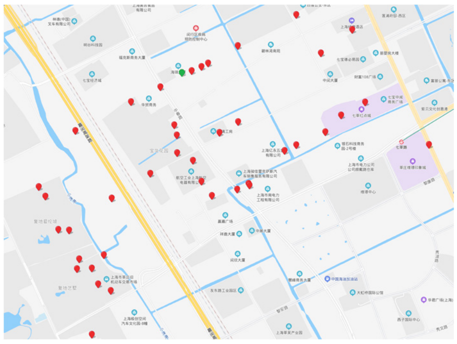

Fig. 2. Locations of the stations and warehouse in the studied zone

We compare the dynamic planning strategy with the static planning strategy. In the dynamic strategy, t_a is set to be 10 minutes and the $\tilde{d}_i$ of unserved stations are updated to their newest demand at the beginning of that anticipation horizon. The process of the static strategy is similar to that of the dynamic strategy except without the demand update. The static problems are all solved by the multiple neighborhood search algorithm.

We use the following index to evaluate a routing plan.

RC—total travel distance of the trucks.

OL—lost riding orders during the horizon.

PSN—the number of batteries to swap in the plan.

ASN—the actual number of batteries swapped.

OBN—the number of e-bikes whose power is below the threshold in the zone at the end of the time horizon.

The results under the two different strategies are compared in TABLE II. Compared with the case of no battery swapping, the routing plan obtained by using our model and algorithm significantly reduces the order loss and the number of low-power e-bikes with reasonable routing cost.

The dynamic planning strategy reduced the routing cost by 11.5% compared to the static strategy. From the perspective of PSN and ASN, under the dynamic programming strategy, the estimation of the number of e-bikes demanding battery swapping at each station is more accurate.

TABLE II
COMPUTATIONAL RESULTS OF THE CASE

Strategy	RC	OL	PSN	ASN	OBN
no swapping	-	48	-	-	939
static planning strategy	13781	27	594	535	524
dynamic planning strategy	12196	25	565	539	519

C. Sensitivity Analysis

we change Q to explore its impact on the dynamic planning strategy. As is shown in TABLE III, when Q is set to 150, the routing cost of the dynamic planning strategy is still improved by 6.2% compared with the static strategy. The difference between the RC of the two strategies further decreases when Q is 150 and disappears when Q equals 250. The dynamic strategy brings lesser improvement on the routing cost when Q is relatively too large or too small.

TABLE III
COMPARISON BETWEEN THE DYNAMIC STRATEGY AND STATIC STRATEGY WITH DIFFERENT TRUCK CAPACITY

Q	Strategy	RC	OL	ASN	OBN
100	static planning strategy	19155	13	521	536
	dynamic planning strategy	18467	16	524	535
150	static planning strategy	14723	17	526	552
	dynamic planning strategy	13814	19	548	534
250	static planning strategy	11449	13	531	546
	dynamic planning strategy	11449	11	547	525

VI. CONCLUSION

This paper investigated the routing of battery swapping trucks in an EBS system. In practice, trucks are allowed to return to the battery warehouse to reload batteries and the uncertain riding behavior of the users often leads to changes in the battery demand of e-bike stations. To consider the above two characteristics, a dynamic strategy is proposed, which slices the entire horizon and constructs a new static problem at each stage which is solved by a multiple neighborhood search algorithm. The algorithm is compared with commercial software on the modified Solomon datasets. The solutions obtained by the multi-neighborhood search algorithm are better than those obtained by commercial software. A case study based on an EBS service zone in Shanghai was conducted to illustrate the effectiveness of the dynamic strategy. The results show that the dynamic strategy reduces the transportation cost by approximately 11.5% compared to the static strategy.

The future research direction is to consider optimizing the dynamic planning strategy by combining it with a more accurate method of forecasting the demands at stations.

REFERENCES

[1] T. Bieliński, Ł. Dopierała, M. Tarkowski, and A. Ważna, "Lessons from Implementing a Metropolitan Electric Bike Sharing System," *Energies*, vol. 13, no. 23, p. 6240, 2020.

[2] A. A. Campbell, C. R. Cherry, M. S. Ryerson, and X. Yang, "Factors influencing the choice of shared bicycles and shared electric bikes in Beijing," *Transp. Res. Part C Emerg. Technol.*, vol. 67, pp. 399–414, 2016.

[3] P. Zhou, C. Wang, Y. Yang, and X. Wei, "E-sharing: Data-driven online optimization of parking location placement for dockless electric bike sharing," *Proc. - Int. Conf. Distrib. Comput. Syst.*, vol. 2020-November, pp. 474–484, 2020.

[4] C. FENG, B. QIN, and L. U. YE, "A Research on Distribution Problem of Electric Bicycle-sharing Batteries," *Ind. Eng. J.*, vol. 22, no. 3, 2019.

[5] M. Schiffer, M. Schneider, G. Walther, and G. Laporte, "Vehicle routing and location routing with intermediate stops: A review," *Transp. Sci.*, vol. 53, no. 2, pp. 319–343, 2019.

[6] E. Prescott-Gagnon, G. Desaulniers, and L.-M. Rousseau, "Heuristics for an oil delivery vehicle routing problem," *Flex. Serv. Manuf. J.*, vol. 26, no. 4, pp. 516–539, 2014.

[7] B. Wang, Y. Liang, M. Yuan, H. Zhang, and Q. Liao, "A metaheuristic method for the multireturn-to-depot petrol truck routing problem with time windows," *Pet. Sci.*, vol. 16, no. 3, pp. 701–712, 2019.

[8] N. A. Mat, A. M. Benjamin, and S. Abdul-Rahman, "Enhanced heuristic algorithms with a vehicle travel speed model for time-dependent vehicle routing: a waste collection problem," *J. Inf. Commun. Technol.*, vol. 17, no. 1, pp. 55–78, 2017.

[9] A. Gutierrez, L. Dieulle, N. Labadie, and N. Velasco, "A hybrid metaheuristic algorithm for the vehicle routing problem with stochastic demands," *Comput. \& Oper. Res.*, vol. 99, pp. 135–147, 2018.

[10] A. L. Erera, J. C. Morales, and M. Savelsbergh, "The vehicle routing problem with stochastic demand and duration constraints," *Transp. Sci.*, vol. 44, no. 4, pp. 474–492, 2010.

[11] E. Cao and M. Lai, "The open vehicle routing problem with fuzzy demands," *Expert Syst. Appl.*, vol. 37, no. 3, pp. 2405–2411, 2010.

[12] X. Xiang, J. Qiu, J. Xiao, and X. Zhang, "Demand coverage diversity based ant colony optimization for dynamic vehicle routing problems," *Eng. Appl. Artif. Intell.*, vol. 91, p. 103582, 2020.

[13] V. Pillac, M. Gendreau, C. Guéret, and A. L. Medaglia, "A review of dynamic vehicle routing problems," *Eur. J. Oper. Res.*, vol. 225, no. 1, pp. 1–11, 2013.

[14] U. Ritzinger, J. Puchinger, and R. F. Hartl, "A survey on dynamic and stochastic vehicle routing problems," *Int. J. Prod. Res.*, vol. 54, no. 1, pp. 215–231, 2016.

[15] I. Markov, S. Varone, and M. Bierlaire, "Integrating a heterogeneous fixed fleet and a flexible assignment of destination depots in the waste collection VRP with intermediate facilities," *Transp. Res. part B Methodol.*, vol. 84, pp. 256–273, 2016.

[16] M. M. Solomon, "Algorithms for the vehicle routing and scheduling problems with time window constraints," *Oper. Res.*, vol. 35, no. 2, pp. 254–265, 1987.