Adjustable robust optimization for fleet sizing problem in closed-loop supply chains with simultaneous delivery and pickup

Ali Borumand [a,*], Ali S. Nookabadi [a]

[a] Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran

Abstract

The Fleet Sizing Problem (FSP) stands as a critical challenge within the realm of logistics and supply chain management, particularly in the context of Closed-Loop Supply Chains (CLSC). The significance of addressing the FSP lies in its direct impact on operational costs, resource utilization, and environmental sustainability. By effectively optimizing fleet size, organizations can streamline transportation operations, minimize fuel consumption, reduce carbon emissions, and ultimately enhance overall supply chain performance. Moreover, in CLSC management, where the coordination of forward and reverse logistics activities is paramount, tackling the FSP becomes even more crucial. Efficient fleet sizing enables businesses to effectively manage product returns, remanufacturing, and recycling processes, thereby fostering circular economy principles and maximizing resource utilization.

In this study, we address the FSP and vehicle routing decisions within a CLSC context. We propose an MILP model and employ a multi-stage adjustable robust optimization (ARO) formulation to handle the nondeterministic nature of demand for new products and requests for pickups of used products. We reconfigure an exact oracle-based algorithm and a heuristic search algorithm to derive upper and lower bounds on the optimal solution of the ARO problem. Additionally, we introduce a metaheuristic algorithm to function as the oracle. Our numerical experiments demonstrate that our metaheuristic approach, which is integrated with the aforementioned methods, significantly enhances both the computational efficiency and solution quality.

Keywords: Fleet sizing, Closed-loop supply chain, Uncertain demand, Adjustable robust optimization, Vehicle routing problem with simultaneous delivery and pickup.

1 Introduction

The Vehicle Routing Problem (VRP) stands at the intersection of optimization and logistics, aiming to efficiently plan and manage the delivery of goods or services using a fleet of vehicles. This challenging computational problem has significant implications for industries relying on transportation networks,

* Corresponding Author, Email: a.borumand@in.iut.ac.ir

impacting operational costs, fuel consumption, and overall environmental sustainability. The roots of VRP can be traced back to the mid-20th century when logistics and supply chain optimization became critical in the face of expanding industrial activities [1].

With the rise of e-commerce, urbanization, and global supply chains, the significance of VRP has further intensified, making it a critical element in modern logistics and transportation management. Over the years, various types of VRP have been identified and categorized to address the specific industries and applications. Two common variations include the Capacitated VRP (CVRP), and the VRP with Simultaneous Delivery and Pickup (VRPSDP). While VRP primarily focuses on optimizing routes for the delivery of goods to customers, CVRP extends the concept by incorporating the constraint of vehicle capacity explicitly into the optimization process and VRPSDP introduces a transformative layer by seamlessly incorporating both delivery and pickup operations within a single logistical framework.

In the context of Closed-Loop Supply Chains, where the management of reverse logistics and the integration of forward and reverse flows are paramount, VRPSDP becomes a natural progression. CLSCs involve not only the distribution of products to end-users but also the retrieval of items for recycling, remanufacturing, or proper disposal. VRPSDP, by accommodating simultaneous delivery and pickup, aligns with the intricacies of CLSCs, offering a more comprehensive approach to logistics optimization. The integration of VRPSDP into the CLSC framework addresses the holistic nature of modern supply chain challenges, where sustainability, cost-effectiveness, and operational efficiency are essential considerations.

In realm of VRP, fleet sizing is one of the most important long-term decisions as it needs a major investment for whole horizon planning [2]. Typically, the demand of customers is faced with uncertainty and may vary in each period of horizon planning. So, the optimal fleet size will often be different on different periods but in practice companies need to buy or rent a fleet of vehicles for the horizon planning. Hence, determining the fleet size is an important and challenging decision that ensures an adequate number of vehicles are available to meet demand while minimizing acquisition costs.

2 Literature review

This section reviews the research streams related to this study: VRPSDP in CLSC and fleet sizing in VRP. By doing this, we indicate the research gaps.

2.1 VRPSDP in CLSC

In realm of VRPSD in CLSC management, Dethloff [3] proposes a deterministic mixed-integer linear programming (MILP) model, marking one of the initial works in this domain. Guo et al. [4], in their study of a CLSC for fresh food E-commerce enterprises, focus on VRPSDP with location decisions, aiming to minimize both cost and low-carbon emissions.

Non-linearity becomes a consideration in [5], where the authors propose a deterministic mixed-integer non-linear programming (MINLP) model for a single-product CLSC. This model integrates VRPSDP with location and inventory management decisions to minimize total costs within a single-period horizon. Addressing quality defects of returned products in a multi-product CLSC, Deng et al. [6] propose a deterministic VRPSDP model with the objective of minimizing costs over a multi-period horizon.

Iassinovskaia et al. [7] tackle time windows in VRPSDP for a CLSC, presenting an MILP model for returnable transport items (RTIs) with inventory management decisions. This study extends to multi-product and multi-objective CLSCs in subsequent studies [8, 9]. Previous research have considered homogeneous vehicles but, Qiu et al. [10] advocate for a heterogeneous fleet of capacitated vehicles. They investigate production planning and VRPSDP in a CLSC, proposing an MILP formulation with deterministic parameters. This formulation accounts for irreparable returned units during the remanufacturing step.

Moreover, there are studies that consider uncertainty in Closed-Loop Supply Chain (CLSC) management. Yuchi et al. [11] investigate a single-period Closed-Loop Supply Chain (CLSC) with a manufacturer and multiple distribution and remanufacturing centers. They utilized a Vehicle Routing Problem (VRP) with a homogeneous fleet of vehicles in the distribution phase, aiming to minimize system-wide costs. The proposed MINLP model considered random demand. In contrast, Pedram et al. [12] address a CLSC with a heterogeneous fleet responsible for distributing and collecting new and end-of-life products. They incorporated fuzzy and random uncertain data using fuzzy-stochastic mathematical programming.

Multi-period CLSC optimization problems is addressed by Soysal [13] where a probabilistic MILP model is proposed to consider uncertainty in demand for returnable transport items (RTIs). Kumar et al. [14] present an MINLP model with a heterogeneous fleet for integrating RTIs and a CLSC of perishable products with uncertain demand. Shuang et al. [15] focus on profit-maximizing carbon emission control policies, utilizing a stochastic MILP model with a heterogeneous fleet in the home appliances industry. Nasiri et al. [16] develop an MINLP model minimizing lost sales, considering time windows and addressing demand uncertainty through stochastic programming.

Several studies explored multi-objective approaches. Gholizadeh et al. [17] develop a multi-period MILP model for a multi-product dairy CLSC, employing a static robust approach to handle uncertainty and considering both environmental and economic aspects. The issue of minimizing lost sales alongside cost received attention in [18], where VRPSDP with time windows and a heterogeneous fleet of vehicles was considered under uncertain conditions. Demand uncertainty in a CLSC of the cable and wire industry is addressed through a stochastic scenario-based approach in [19]. Additionally, the authors applied a carbon tax policy to reduce emissions and vehicle waiting times. Tavana et al. [20] address VRPSDV in a sustainable CLSC by a multi-objective MILP model considering uncertain demand. Authors propose a fuzzy goal programming approach to solve the model.

2.2 Fleet sizing in VRP

As we've discussed, fleet sizing decision is crucial for businesses because they need to purchase or lease vehicles at the beginning of planning horizon to satisfy the uncertain demand in all periods of the horizon.

One of the initial studies on fleet sizing is [21] where authors propose a deterministic MILP model to address the problem of routing a fleet of vehicles from a central depot to customers in which vehicle capacity is considered. Desrochers and Verhoog [22] improve the previous study by proposing a new heuristic method. The authors in [23-25] extend the previous study by considering Time windows in fleet sizing optimization. The authors enhanced existing heuristic methods and introduced a new metaheuristic approach for solving the problem.

Belfiore and Yoshizaki [26] add split delivery to fleet size optimization. In this study, customer demand can be divided and served by more than one vehicle. Authors proposed a heuristic scatter-search approach to solve the problem. Hiermann et al. [27] apply fleet size optimization to electric vehicles and Recharging Stations considering time windows. The objective of this study is to serve the customer while minimizing acquisition costs and the total distance travelled. Authors proposed an MILP model along with a branch-and-price algorithm and a hybrid heuristic to address the problem.

Synchronized visits along with time windows is considered in [28] where authors address homecare givers in the healthcare sector. In synchronized Visits VRP each customer needs be visited by more than one vehicle in the same time. The authors propose an MILP model with objective function of minimizing the total acquisition and operational costs and develop a metaheuristic algorithm to solve the problem. Pasha et al. [29] extend the work by considering multi period planning and the objective is to find the best fleet size and routing to satisfy the customer demand over a set of periods. The Authors developed a heuristic algorithm based on tabu search to solve the problem.

Multi depot problem in fleet size optimization is addressed in [30] where authors develop a bi-objective MILP model to concurrently minimize the transportation cost in the entire waste management system along with minimizing the lost capacity of transfer stations. These stations are depots which are used to store and sort wastes and send them to treatment or recycling centers. Schmidt et al. [31] propose another multi depot that take into account time interval to address traffic and congestion issues. The objective function is to minimize the sum of routing and fixed vehicle costs and the authors propose a metaheuristic algorithm to solve the problem. They show that this metaheuristic is more effective than the exact method in terms of solution quality and computational time.

In order to approach real-world applications, there are also studies in the literature that takes into account the uncertainty to the fleet sizing VRP. Vanga and Venkateswaran [32] address reusable articles with uncertain demand and develop analytical models for optimal fleet sizing. Chang et al. [33] delve into the topic of fleet sizing in post-disaster logistics. The authors consider uncertainty in demand and road network condition and propose a two-stage stochastic programming model. In the first stage, it optimizes the location of relief goods distribution centers and the number of allocated vehicles. The second stage focuses on determining optimal vehicle and inventory routing decisions during the critical initial time window following the revelation of random factors associated with the disaster. Lei et al. [2] develop a two-stage robust optimization model for mobile facility fleet sizing and routing problem. This study accounts for uncertainty in demand within a multi-period planning framework.

2.3 Research gaps and contributions

To the best of our knowledge, no study has been conducted on fleet sizing VRP with simultaneous delivery and pickup in CLSC management considering uncertainty. In the realm of fleet sizing in Closed-Loop Supply Chains (CLSC), existing studies have overlooked the adaptability of variables to uncertain parameters, despite the multi-stage nature of the problem within a multiperiod horizon. To fill these gaps, we introduce an ARO approach aimed at addressing uncertainties in CLSC. So, here are our contributions to the literature:

- First, we propose a new deterministic MILP model to fleet sizing vehicle routing problem with simultaneous delivery and pickup in CLSC management.
- Second, we develop a multi-stage adjustable robust optimization approach to address the uncertainty in the problem.
- Additionally, we reconfigured the exact and heuristic methods from [34] to solve the ARO fleet sizing problem of our study.
- Finally, we propose a metaheuristic approach based on simulated annealing (SA) algorithm to improve the computational time and solution quality of exact and heuristic approaches.

The rest of the paper is organized as follows. Section 3 provides a detailed description of the problem. The formulation of deterministic and multi-stage ARO problems are presented in Sections 4 and 5, respectively. Section 6 contains the solution methods on the multistage ARO problem, where the customized metaheuristic method based on SA algorithm is explained. Numerical findings based on comparison among solution methods are discussed in Section 7, and we provide our conclusions in Section 8.

3 Problem description

We consider a CLSC of a producer and its related retailers. One kind of product is produced at the production site. The producer uses homogenous fleet of vehicles with capacity $C^{\mathcal{K}}$ to ship the new products to retailers and collect the used ones. $\mathcal{N}$ is the set of all nodes where $\mathcal{N}\backslash\{0\}$ is the set of all retailers (pickup and delivery locations) and $\{0\}$ is the production site.

We consider a time horizon $\mathcal{T}$ divided into T equal time periods, i.e., $\mathcal{T} = \{1, 2, \ldots, T\}$, in each of which shipment of new and used products to/from retailers take place. At the beginning of the time horizon, the producer decides on the number of vehicles with the unit cost of c^P to use for shipment activity in the time horizon.

In each period t, the producer receives a demand for d_{it} units of new products and a request to pick up d'_{it} units of used products from retailer i. In this study, we assume that d and d' are independent. Also, the demands across periods are independent. We have $d_t = [d_{it}]_{i\in\mathcal{N}\backslash\{0\}} \in D_t \subset \mathbb{R}^{|\mathcal{N}\backslash\{0\}|}$, and $d'_t = [d'_{it}]_{i\in\mathcal{N}\backslash\{0\}} \in D'_t \subset \mathbb{R}^{|\mathcal{N}\backslash\{0\}|}$, where D_t and D'_t are full-dimensional polytopes, $t \in \mathcal{T}$.

We denote the CLSC network by $\mathcal{G} = (\mathcal{N}, \mathcal{A})$, where $\mathcal{A} = \{(i,j): i,j \in \mathcal{N}, i \neq j\}$ is the set of routes connecting retailers and production site. When the demand is realized in each period, a fleet of vehicles is assigned to visit multiple retailers. Each vehicle starts its tour by leaving production site with the new products to satisfy the complete demand of chosen retailers and to pick up all the used products.

Traveling each route incurs fixed and variable costs. Fixed cost, $FixC$, includes the costs that are independent of the route. The variable cost, c_{ij}, is for traversing the route (i, j), $i, j, \in \mathcal{N}$ to deliver one unit of new product or collecting one unit of used product. At each stop at a retailer, once the new products which are demanded by the retailer are unloaded, all of the used products are picked up to be returned to the production site. At the end of each period, each vehicle finishes its tour by returning to the same production site where it started.

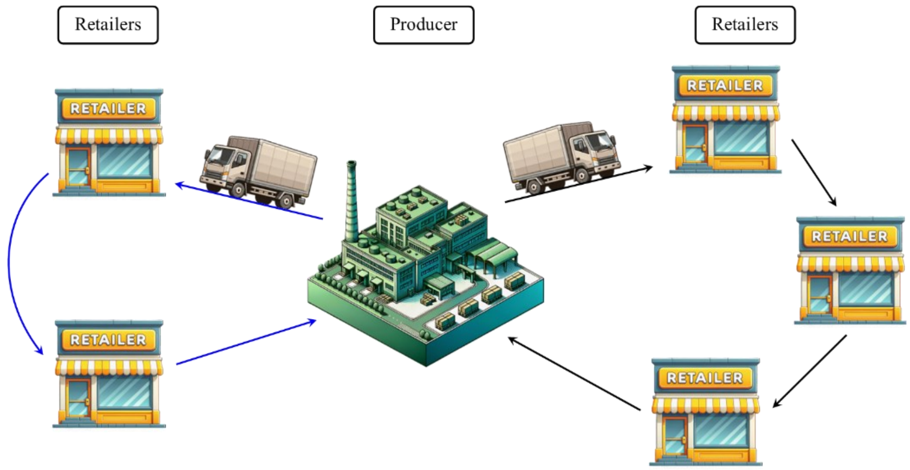


Figure 1. The schema of proposed CLSC

Figure 1 schematically illustrates the CLSC network. In this supply chain, there are two distinct echelons. The first echelon consists of the production site and the second echelon is made of retailers who demand new products and request pickups of used products. Production site and retailers are located in different geographical locations, and these locations are known. This figure also shows that homogenous vehicles may travel specific tours between production site and retailers.

The producer needs to make three types of decisions:

- the number of vehicles to be purchased at the beginning of the time horizon at production site, denoted by Q;
- whether each vehicle is assigned to travel the route $(i, j) \in \mathcal{A}$ during period t, denoted by x_{ijt};
- and the number of new and used products that each vehicle carries on the route $(i, j) \in \mathcal{A}$ during period t, denoted by v_{ijt} and u_{ijt}, respectively.

The summary of the notations and assumptions are provided below.

Sets:

$\mathcal{N}$:	Set of all locations including producer ($\{0\}$) and retailers (pickup and delivery locations) ($\mathcal{N}\backslash\{0\}$), $\mathcal{N} = \{0, 1, 2, \dots, \lvert\mathcal{N}\rvert\}$
$\mathcal{A}$:	Set of routes, $\mathcal{A} = \{(i, j): i, j \in \mathcal{N}, i \neq j\}$
$\mathcal{G}$:	Network of the CLSC, $\mathcal{G} = (\mathcal{N}, \mathcal{A})$
$\mathcal{T}$:	Set of planning periods, $\mathcal{T} = \{1, 2, \dots, \lvert\mathcal{T}\rvert\}$

Parameters:

$C^{\mathcal{K}}$:	Capacity of homogenous vehicles
c^P:	Unit purchasing cost of new vehicle
$FixC$:	Fixed cost of using each vehicle on each route
c_{ij}:	Unit shipment cost over route (i, j), $(i, j \in \mathcal{N})$
d_{it}:	Demand on new products from retailer i in period t, $(i \in \mathcal{N}\backslash\{0\})$
d'_{it}:	Quantity of used products asked by retailer i to be picked up in period t, $(i \in \mathcal{N}\backslash\{0\})$

Variables:

Q:	Number of purchased vehicles at the beginning of the time horizon
v_{ijt}:	New products quantity shipped over route (i, j) in period t, $(i, j \in \mathcal{N})$
u_{ijt}:	Used products quantity shipped over route (i, j) in period t, $(i, j \in \mathcal{N})$
x_{ijt}:	Binary variable, equal to 1 if route (i, j) is traversed in period t, 0 otherwise, $(i, j \in \mathcal{N})$

Assumptions:

1. The locations of production sites and retailers are known.
2. Demand for new products within a given period should be met entirely during that same period.
3. Requests for pickups of used products within a given period should be met entirely during that same period.
4. At each period, only one vehicle can visit a retailer to deliver the new products and pick up the used ones.
5. The uncertainty sets are full-dimensional.
6. Demand for new products and requests for pickups of used products are independent.

Assumptions 1 to 4 are taken from the CLSC literature. Assumption 5 is a usual assumption in robust optimization (Bertsimas and Goyal 2012). Assumption 12 is practically relevant because without it, there would be no need for production. Assumption 6 is the only restrictive assumption due to a lack of information regarding the periods and retailers where customers return their used products. These factors can be further investigated in future research to relax this assumption.

4 Deterministic model

In this section, we formulate a deterministic MILP model, where we assume that the demand for new products and requests for pickups of used products are exactly known for all periods. Given the notation, the deterministic problem is formulated as follows:

$$min \quad \sum_{t\in\mathcal{T}} \sum_{(i,j)\in\mathcal{A}} c_{ij}\left(v_{ijt} + u_{ijt}\right) \tag{1}$$

$$+\sum_{t\in\mathcal{T}} \sum_{(i,j)\in\mathcal{A}} FixCx_{ijt} \tag{2}$$

$$+ c^{P}Q \tag{3}$$

$$s.t. \quad \sum_{i\in\mathcal{N}\backslash\{0\}} x_{0it} \leq Q, \qquad \forall t \in \mathcal{T}, \tag{4}$$

$$\sum_{i\neq j\in\mathcal{N}} x_{ijt} \leq 1, \qquad \forall j \in \mathcal{N}\backslash\{0\}, t \in \mathcal{T}, \tag{5}$$

$$\sum_{j\neq i\in\mathcal{N}} x_{ijt} - \sum_{j\neq i\in\mathcal{N}} x_{jit} = 0, \qquad \forall i \in \mathcal{N}, t \in \mathcal{T}, \tag{6}$$

$$\sum_{j\neq i\in\mathcal{N}} v_{jit} - \sum_{j\neq i\in\mathcal{N}} v_{ijt} = d_{it}, \qquad \forall i \in \mathcal{N}\backslash\{0\}, t \in \mathcal{T}, \tag{7}$$

$$\sum_{j\neq i\in\mathcal{N}} u_{ijt} - \sum_{j\neq i\in\mathcal{N}} u_{jit} = d'_{it}, \qquad \forall i \in \mathcal{N}\backslash\{0\}, t \in \mathcal{T}, \tag{8}$$

$$v_{ijt} + u_{ijt} \leq C^{\mathcal{K}} x_{ijt}, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, t \in \mathcal{T}, \tag{9}$$

$$Q \geq 0, \tag{10}$$

$$v_{ijt}, u_{ijt} \geq 0, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, t \in \mathcal{T}, \tag{11}$$

$$x_{ijt} \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, t \in \mathcal{T}. \tag{12}$$

The objective function is to minimize the total costs including variable (1) and fixed (2) shipment costs and the purchasing cost of vehicles at the beginning of the time horizon (3). Constraints (4) determines the number of vehicles which can be utilized in each period. Constraints (5) ensures that at each period a retailer can only be served once. Constraint (6) assures that if a vehicle enters a retailer, it leaves that retailer. Also, if a vehicle leaves a production site it comes back to it to complete a tour. Constraints (7) and (8) are the flow conservation constraints for new and used products, respectively. They show the quantity of inflow and outflow of new and used products for each retailer. Constraint (9) ensures that the total quantity of new and used products traveling over a route does not exceed the capacity of the vehicle. Constraints (10) - (12) are to make sure that the decision variables belong to their corresponding domain.

5 Multi-stage adjustable robust formulation

In this section, we describe the adjustable RO model formulation of the problem. According to the chronological sequence of events presented in Figure 2, in each period the first decision to be made is regarding the number of vehicles to be purchased. This decision variable is called "here-and-now" variable. Then, in each period, demand for new products and requests for pickups of used products are revealed. At the end of each period, based on the realized demand, shipment decisions are made. To formulate the multi-stage ARO problem, we use $d_{[t]}$ to denote the vector containing the demand for new products up to and including the period t and we use $d'_{[t]}$ to denote the vector containing the requests for pickups of used products up to and including the period t. Also, we denote the joint uncertain parameter up to the end of period t by $\xi_{[t]}$; that is the vector of joint $d_{[t]}$ and $d'_{[t]}$ realized until the end of period t. The uncertainty set with respect to $\xi_{[t]}$ is $\Xi_{[t]} = \times_{\tilde{t}=1}^{t}\left(d_{\tilde{t}} \times d'_{\tilde{t}}\right) \subset \mathbb{R}^{2|\mathcal{N}\setminus\{0\}|\times t}$, where $\times$ is the Cartesian product.

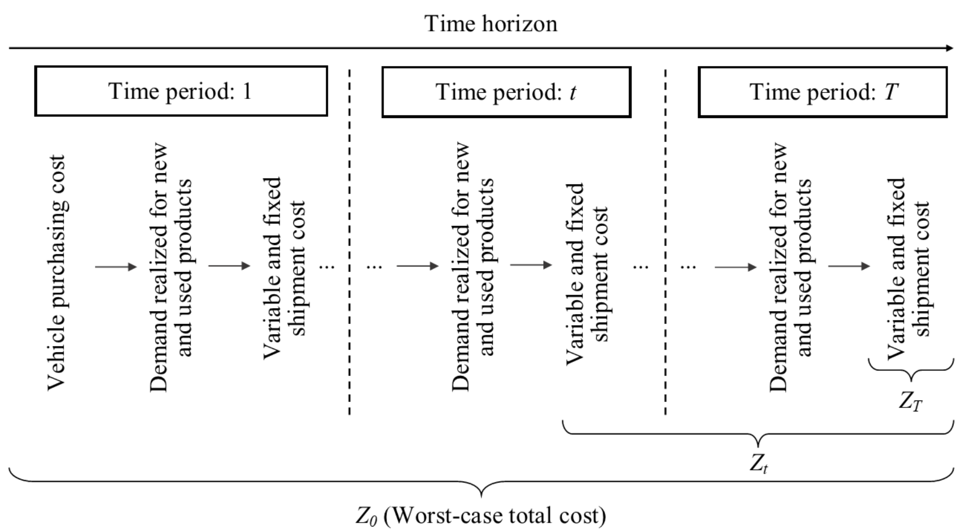


Figure 2. Events occurrence schema

In each period, the decisions are made based on the realization of demand for new products and requests for pickups of used products. As the value of variable v_{ijt} depends on all past demand for new products, we restrict their "wait-and-see" formulation to depend on the realization of d, denoted by $v_{ijt}\left(d_{[t]}\right)$. This means v_{ijt} depends on d_{it} and $v_{ij,t-1}$. Furthermore, with the same logic, the "wait-and-see" variable u_{ijt} depends on the realization of d', denoted by $u_{ijt}\left(d'_{[t]}\right)$. Finally, the "wait-and-see" variable x_{ijt} depends on the realization of both demand and pickup requests. Hence, they are denoted by $x_{ijt}\left(\xi_{[t]}\right)$.

Before presenting the multi-stage ARO formulation, we need to avoid equality constraints containing uncertain parameters. Hence, two equality constraints (7) and (8) should be reformulated and replaced with inequality constraints.

Proposition 1. In the deterministic problem, constraint (7) can be replaced by

$$\sum_{j \neq i \in \mathcal{N}} v_{jit} - \sum_{j \neq i \in \mathcal{N}} v_{ijt} \geq d_{it}, \qquad \forall i \in \mathcal{N} \backslash \{0\}, t \in \mathcal{T}. \tag{13}$$

Proof. By definition, any solution of the deterministic problem satisfies the constraint (13). Now, we prove that any optimal solution of the revised deterministic problem, where (7) is replaced by (13), satisfies constraint (7).

By contradiction, let us assume that in an optimal solution of the revised deterministic problem, constraint (7) is not satisfied. So, there exist $\bar{\iota} \in \mathcal{N} \backslash \{0\}$ and $\bar{t} \in \mathcal{T}$ such that

$$\sum_{j \neq \bar{\iota} \in \mathcal{N}} v_{j\bar{\iota}\bar{t}} - \sum_{j \neq \bar{\iota} \in \mathcal{N}} v_{\bar{\iota}j\bar{t}} > d_{\bar{\iota}\bar{t}}. \tag{14}$$

Now, let us construct a solution where all decisions are the same as the considered optimal solution except the quantity of new products shipped over routes. This variable in new solution is

$$v_{jit} =$$

$$\begin{cases} v_{jit} - \dfrac{v_{j\bar{\iota}\bar{t}} - \left(d_{\bar{\iota}\bar{t}} + v_{\bar{\iota}j'\bar{t}}\right)}{2}, & i = \bar{\iota}, t = \bar{t}, j = \{0\} \text{ or a retailer before } \bar{\iota} \text{ in a tour which inclouds } \bar{\iota}, \\ & j' = \text{a retailer after } \bar{\iota} \text{ in a tour which inclouds } \bar{\iota}, \\ v_{jit}, & \text{otherwise.} \end{cases}$$

Based on the construction, one can see that the new solution is feasible for the revised problem. The objective value of the new solution is strictly lower than the considered optimal solution, as it has fewer empty bottles in the inventory. This contradicts with the optimality of the considered solution. Therefore, any optimal solution to the revised problem satisfied (7), which concludes the proposition. □

Proposition 2. In the deterministic problem, constraint (8) can be replaced by

$$\sum_{j \neq i \in \mathcal{N}} u_{ijt} - \sum_{j \neq i \in \mathcal{N}} u_{jit} \geq d'_{it}, \qquad \forall i \in \mathcal{N} \backslash \{0\}, t \in \mathcal{T}. \tag{15}$$

Proof. Similar to proof of Proposition 1. □

As a result, by replacing constraints (7) and (8) with constraints (13) and (15), the multi-stage ARO formulation is:

$$Z_0 = \quad min \quad c^P Q \tag{16}$$

$$+ \max_{\xi_{[1]} \in \Xi_{[1]}} Z_1\left(\xi_{[1]}, Q\right) \tag{17}$$

$$s.t. \quad Q \geq 0, \tag{18}$$

where, for any $t \in \{1, 2, \ldots, T-1\}$, we have

$$Z_t(\xi_{[t]}, Q) =$$

$$min \quad \sum_{(i,j)\in\mathcal{A}} c_{ij} \left(v_{ijt}(d_{[t]}) + u_{ijt}(d'_{[t]}) \right) \tag{19}$$

$$+ \sum_{(i,j)\in\mathcal{A}} FixCx_{ijt}(\xi_{[t]}) \tag{20}$$

$$+ \max_{\xi_{[t+1]}\in\Xi_{[t+1]}} Z_{t+1}\left(\xi_{[t+1]}, Q\right) \tag{21}$$

$$s.t. \quad \sum_{i\in\mathcal{N}\setminus\{0\}} x_{0it}(\xi_{[t]}) \leq Q, \tag{22}$$

$$\sum_{i\neq j\in\mathcal{N}} x_{ijt}(\xi_{[t]}) \leq 1, \qquad \forall j \in \mathcal{N}\setminus\{0\}, \tag{23}$$

$$\sum_{j\neq i\in\mathcal{N}} x_{ijt}(\xi_{[t]}) - \sum_{j\neq i\in\mathcal{N}} x_{jit}(\xi_{[t]}) = 0, \qquad \forall i \in \mathcal{N}, \tag{24}$$

$$\sum_{j\neq i\in\mathcal{N}} v_{jit}(d_{[t]}) - \sum_{j\neq i\in\mathcal{N}} v_{ijt}(d_{[t]}) \geq d_{it}, \qquad \forall i \in \mathcal{N}\setminus\{0\}, \tag{25}$$

$$\sum_{j\neq i\in\mathcal{N}} u_{ijt}(d'_{[t]}) - \sum_{j\neq i\in\mathcal{V}} u_{jit}(d'_{[t]}) \geq d'_{it}, \qquad \forall i \in \mathcal{N}\setminus\{0\}, \tag{26}$$

$$v_{ijt}(d_{[t]}) + u_{ijt}(d'_{[t]}) \leq C^{\mathcal{K}} x_{ijt}(\xi_{[t]}), \qquad \forall (i,j) \in \mathcal{A}, i \neq j, \tag{27}$$

$$v_{ijt}(d_{[t]}), u_{ijt}(d'_{[t]}) \geq 0, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, \tag{28}$$

$$x_{ijt}(\xi_{[t]}) \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, \tag{29}$$

and

$$Z_T(\xi_{[T]}, Q) =$$

$$min \quad \sum_{(i,j)\in\mathcal{A}} c_{ij} \left(v_{ijT}(d_{[T]}) + u_{ijT}(d'_{[T]}) \right) \tag{30}$$

$$+ \sum_{(i,j)\in\mathcal{A}} FixCx_{ijT}(\xi_{[T]}) \tag{31}$$

$$s.t. \quad \sum_{i\in\mathcal{N}\setminus\{0\}} x_{0iT}(\xi_{[T]}) \leq Q, \tag{32}$$

$$\sum_{i\neq j\in\mathcal{N}} x_{ijT}(\xi_{[T]}) \leq 1, \qquad \forall j \in \mathcal{N}\setminus\{0\}, \tag{33}$$

$$\sum_{j \neq i \in \mathcal{N}} x_{ijT}(\xi_{[T]}) - \sum_{j \neq i \in \mathcal{N}} x_{jiT}(\xi_{[T]}) = 0, \qquad \forall i \in \mathcal{N}, \quad (34)$$

$$\sum_{j \neq i \in \mathcal{N}} v_{jiT}(d_{[T]}) - \sum_{j \neq i \in \mathcal{N}} v_{ijT}(d_{[T]}) \geq d_{iT}, \qquad \forall i \in \mathcal{N} \backslash \{0\}, \quad (35)$$

$$\sum_{j \neq i \in \mathcal{N}} u_{ijT}(d'_{[T]}) - \sum_{j \neq i \in \mathcal{V}} u_{jiT}(d'_{[T]}) \geq d'_{iT}, \qquad \forall i \in \mathcal{N} \backslash \{0\}, \quad (36)$$

$$v_{ijT}(d_{[T]}) + u_{ijT}(d'_{[T]}) \leq C^{\mathcal{K}} x_{ijT}(\xi_{[T]}), \qquad \forall (i,j) \in \mathcal{A}, i \neq j, \quad (37)$$

$$v_{ijT}(d_{[T]}), u_{ijT}(d'_{[T]}) \geq 0, \qquad \forall (i,j) \in \mathcal{A}, i \neq j, \quad (38)$$

$$x_{ijT}(\xi_{[T]}) \in \{0,1\}, \qquad \forall (i,j) \in \mathcal{A}, i \neq j. \quad (39)$$

Figure 2 illustrates how the worst-case total cost is calculated based on the worst-case costs from the time period t onward, i.e., for $t = 1, \dots, T$, $Z_t(\xi_{[t]}, Q)$, is the cost of making decisions at the t^{th} time period plus incurred worst-case cost of the future.

6 Solution methods for the ARO formulation

In this section, we focus on how we can solve the multi-stage ARO problem of Section 5. We establish two exact methods (HBP and Heuristic) from [34] and develop a metaheuristic approach based on SA algorithm. The solutions are explained in two streams: upper bound solutions and lower bound solutions.

6.1 Upper bound solutions

In this section, two approaches are presented to calculate an upper bound on the optimal solution of the ARO problem.

The first one is "U-HBP" technique. The HBP technique from [34] provides both upper and lower bounds on the optimal solution of an ARO problem. U-HBP focus on the upper bound part from HBP to increase the algorithm speed. The detailed algorithm of HBP technique has been presented in Section 5.2.2 of [34]. Based on HBP algorithm, the pseudocode of U-HBP technique is presented in Algorithm 1. At the beginning of this algorithm, the first general best upper bound is infinite and leftover polytope ξ in hand is (40). This algorithm will be terminated when a time limitation arises or when there are no new leftovers to split.

Algorithm 1 Algorithm of U-HBP technique

1: Find inner and outer boxes related to set of leftover polytopes ξ in hand.

2: Calculate upper corner points of all outer boxes from step 1.

3: **for** each upper corner point from step 2 **do**

4: solve the deterministic problem of Section 4 with that upper corner point (Using CPLEX as oracle).

5: **if** the difference between the volume of the outer box and the corresponding inner box is greater than μ **then**, split the leftover polytope using its corresponding inner box.

6: Set the maximum solution as best upper bound of step 3.

7: **if** the best upper bound of step 3 is lower than general best upper bound **then**, set it as new general best upper bound.

8: Consider all leftover polytopes from step 5 as new set of leftover polytopes ξ in hand and go to step 1.

The second approach to calculate the upper bound is a customized SA metaheuristic algorithm which is integrated with U-HBP technique named "Meta-HBP". In the following, the customized metaheuristic algorithm and the way it is merged with U-HBP is explained.

6.1.1 Customized SA metaheuristic algorithm

We develop a metaheuristic algorithm along with U-HBP to increase the speed of solving the ARO problem. In this section, we present our customized metaheuristic algorithm. Two parts are essential in a metaheuristic algorithm. The first one is generating an initial solution and the second one is generating the neighborhood solutions.

Initial solution:

The first step in implementing the metaheuristic algorithm is to find a feasible initial solution based on the objective function associated with the mathematical model of the problem. In this study, a heuristic method is used to find the feasible initial solution.

Algorithm 2 Generating a feasible initial solution

1: Sort retailers that have not been assigned to any vehicle in descending order based on their demand for new products.

2: Select the first retailer from the sorted list.

3: **if** the vehicle has enough capacity for delivering/picking up new/used products to/from selected retailer **then**, assign that retailer as the next destination for the vehicle; otherwise, remove that retailer from the list and go to step 2.

4: **if** the vehicle's capacity is filled, and there are still unassigned retailers **then**, consider a new vehicle and go to step 1.

Algorithm 2 presents the pseudocode of generating a feasible initial solution. This algorithm ensures that the capacity of the vehicles is filled in descending order of the highest demand volumes. As a result, it maximizes the utilization of each vehicle's maximum capacity, leading to the deployment of fewer vehicles in each period. Example 1 shows a feasible initial solution based on steps of Algorithm 2.

Example 1. Assume there are 10 retailers with the demand for new products and requests for pickups of used products in 1 period as follows:

Table 1. Value of demand in Example 1

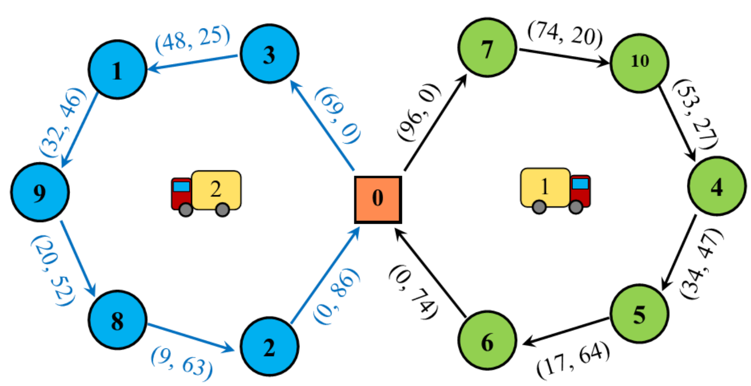
Retailer	New products	Used products	Retailer	New products	Used products
1	16	21	6	17	10
2	9	23	7	22	20
3	21	25	8	11	11
4	19	20	9	12	6
5	17	17	10	21	7

The demand is quantified in product unit, and the vehicle capacity is set at 100 product unit. In accordance with Table 1, Figure 3 illustrates a feasible initial solution.

Vehicles	Tours						
1	0	7	10	4	5	6	0
2	0	3	1	9	8	2	0

Figure 3. A feasible initial solution for Example 1

Furthermore, Figure 4 depicts tours of the initial solution. In this figure, two vehicles have been allocated to satisfy the demand. The sequence of visiting retailers on each tour is specified. Additionally, the numbers on each arrow from left to right represent v_{ijt} and u_{ijt}, respectively.



* (Num 1, Num 2): (v_{ijt}, u_{ijt})

Figure 4. Initial solution tours of Figure 3

Neighborhood solution:

The second step in implementing the metaheuristic algorithm is to generate a neighborhood solution. The neighborhood creation algorithm in this study encompasses three scenarios to make variations in the initial solution. Each of these three scenarios is probabilistically selected.

Scenario 1: In this scenario, the string of retailers assigned to vehicles is merged as one string, and subsequently randomized to match the number of vehicles. Considering the initial solution in Example 1, at first, all tours of vehicles are consecutively arranged (Figure 5).

7	10	4	5	6	3	1	9	8	2

Figure 5. Merged string of retailers from initial solution of Example 1

Following that, the string created in Figure 5 is randomly broken into segments to match the number of vehicles. For instance, Figure 6 illustrates a feasible neighborhood for the initial solution of Example 1.

Vehicles	Tours							
1	0	7	10	4	5	0		
2	0	6	3	1	9	8	2	0

Figure 6. A feasible neighborhood for the initial solution of Figure 3 in Scenario 1

Scenario 2: In this scenario, the positions of two retailers along the tour of a single vehicle are randomly swapped. For instance, Figure 7 depicts a feasible neighborhood for the initial solution of Example 1, in which the positions of retailers 10 and 5 along the tour of vehicle 1 have been exchanged.

Vehicles	Tours						
1	0	7	5	4	10	6	0
2	0	3	1	9	8	2	0

Figure 7. A feasible neighborhood for the initial solution of Figure 3 in Scenario 2

Scenario 3: In this scenario, the positions of two retailers along the tour of two vehicles are exchanged. For example, Figure 8 illustrates a feasible neighborhood for the initial solution of Example 1, where the positions of sales centers 3 and 2 along the routes of vehicles 1 and 2 have been exchanged.

Vehicles	Tours						
1	0	7	9	4	5	6	0
2	0	3	1	10	8	2	0

Figure 8. A feasible neighborhood for the initial solution of Figure 3 in Scenario 3

In the generation of a neighborhood solution, the feasibility of new solution is not guaranteed. Therefore, after generating a neighborhood, the new solution is examined for feasibility. If the new solution is not feasible, the process of generating a new neighborhood continues until a feasible solution is reached.

6.1.2 Meta-HBP

We integrate our customized SA metaheuristic algorithm with the HBP technique, naming it "Meta-HBP". This implies that in Meta-HBP, instead of employing the exact solution provided by CPLEX, we utilize our metaheuristic algorithm for upper-bound calculations. More specifically, in Algorithm 1, our customized SA method is employed as an oracle in step 4.

6.2 Lower bound solutions

In this section, two approaches are presented to calculate a lower bound on the optimal solution of the ARO problem. The first one is the heuristic search approach from [34]. In this study, we reconfigure this approach to effectively address and solve our problem. The detailed algorithm of heuristic search approach has been presented in Section 5.3 of [34].

The heuristic search approach provides a lower bound on the optimal solution of the ARO problem but as it is explained in [34], this approach needs to calculate many upper bounds in each iteration of solution procedure as well. Due to the time-consuming nature of calculating upper bounds using exact methods, we present our second approach in this section to mitigate the solution time.

The second approach is referred to as "MetaH-H" which is an integration of the heuristic search approach from [34] and our metaheuristic algorithm. We restructure the heuristic search approach to have the upper bound calculation performed by our metaheuristic algorithm rather than an exact method to expedite the calculation of these upper bounds. In the next section, the performance of the proposed solution methods in this section will be compared.

7 Numerical experiments

In this section, we show how our methods proposed in Section 6 can solve the multi-stage ARO fleet sizing problem in CLSC. The numerical experiments were run in a Virtual Machine with 8 processors 2.40 GHz and 32 GB RAM running Windows 10. Modeling the deterministic problem is done with YALMIP toolbox [35] in MATLAB R2019a. All the deterministic mixed integer linear optimization problems have been solved using ILOG CPLEX Optimization Studio 12.9.

7.1 Testbed

We randomly generate 9 instances based on [34]. Each instance contains one random coordinate $(\mathcal{X}_0, \mathcal{Y}_0)$ in $[0,500]^2$ which represent the location of production site and $|\mathcal{N}\backslash\{0\}|$ random coordinates $(\mathcal{X}_i, \mathcal{Y}_i)$ in $[0,500]^2$ which represent the locations of retailers. We refer to these locations as nodes.

We use the sets of

$$\Xi_t = \left\{ \xi_t \in \mathbb{R}^{2|\mathcal{N}\backslash\{0\}|} \middle| \begin{array}{l} 0 \le \xi_{it} \le \bar{\bar{\xi}}_{it}, \quad \forall i \in \mathcal{N}\backslash\{0\} \\ \sum_{i\in\mathcal{N}} \xi_{it} \le \bar{\bar{\xi}}_t, \end{array} \right\} \tag{40}$$

as the budget uncertainty set, where $\bar{\bar{\xi}}_{it}, \bar{\bar{\xi}}_t \in \mathbb{N}^2$ are the upper bounds of ξ_{it} and $\sum_{i\in\mathcal{N}} \xi_{it}$, respectively, for $i \in \mathcal{N}\backslash\{0\}, t \in \mathcal{T}$.

The unit of demand is generally referred to as a "product unit". The unit shipment cost between two nodes i and j is calculated as:

$$\mathrm{c}_{ij} = \frac{1\left(\frac{€}{\mathrm{km}}\right) \times Dist_{ij}\ (\mathrm{km})}{C^{\mathcal{K}}\ (\text{product unit})} \quad \left(\frac{€}{\text{product unit}}\right), \tag{41}$$

where

$$Dist_{ij} = \sqrt{\left(\mathcal{X}_i - \mathcal{X}_j\right)^2 + \left(\mathcal{Y}_i - \mathcal{Y}_j\right)^2} \quad (\mathrm{km}) \tag{42}$$

is the Euclidean distance between two nodes i and j, knowing $(\mathcal{X}_i, \mathcal{Y}_i)$ is the location of $i \in \mathcal{N}$.

A summary of the parameters considered in the testbed is presented in Table 2, where $\mathrm{U}^{\mathrm{int}}(.)$ refers to the integer uniform distribution.

Table 2. Value of parameters in the testbed

Parameter	Value	Unit
$\bar{\xi}_{it}$	$\mathrm{U}^{\mathrm{int}}([5,25]^2)$	product unit
$\bar{\xi}_t$	$\max_{j \in \mathcal{N} \setminus \{0\}} \bar{\xi}_{it}$	product unit
$C^{\mathcal{K}}$	100	product unit
c^P	50000	$\frac{€}{\text{vehicle}}$
$FixC$	50	$\frac{€}{\text{rout}}$
c_{ij}	$\frac{Dist_{ij}}{C^{\mathcal{K}}}$	$\frac{€}{\text{product unit}}$

7.2 Results

In this section, we elucidate the outcomes derived from addressing 9 randomly generated instances utilizing four distinct methods: U-HBP, Meta-HBP, heuristic, and MetaH-H, as introduced in Section 6, across two streams of upper and lower bound solutions. U-HBP and heuristic methods represent exact approaches, thus each instance is solved once using these methods. In contrast, Meta-HBP and MetaH-H are metaheuristic approaches; hence, we execute five runs for each instance, from which we derive the averages and standard deviations.

U-HBP and Meta-HBP are used to find an upper bound on the optimal solution of the ARO problem. Table 3 summarizes the results of applying these two methods to the instances. In this table, column "Ins" represents the instances' numbers and column "N-T" shows the number of retailers-periods.

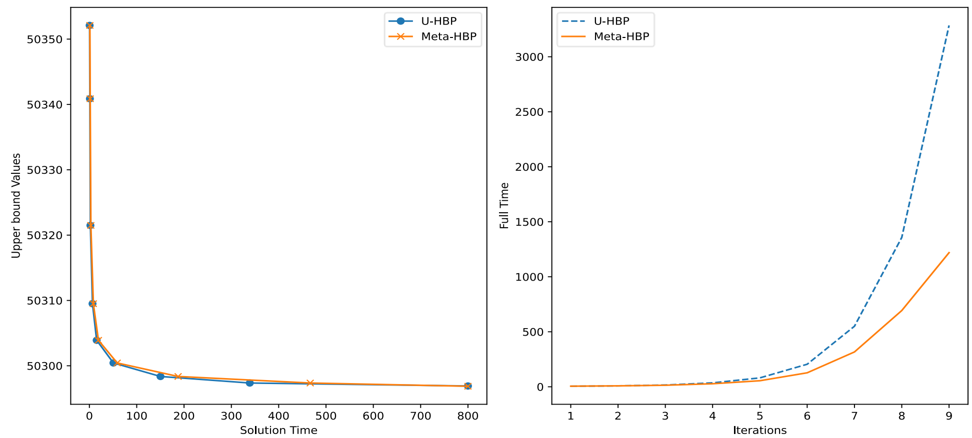
Each method comprises two primary columns: "Time", which includes "Full" and "Sol" columns, and the "Upper bound" column. "Sol Time" indicates the solution time within the constraint of $|\mathcal{N}| \times |\mathcal{T}| \times 400$ seconds, while "Full Time" reflects the total algorithm runtime for each instance, encompassing solution time along with modeling time. Modeling time refers to the duration required for the solver to model the problem, thus it is excluded from the solution time [34].

The “Deter gap” column in the U-HBP section represents the tolerance gap provided to the solver for solving deterministic problems. In the Meta-HBP section, the first number in each cell denotes the average value, and the second number, displayed within brackets, represents the standard deviation associated with runs on each instance. The final part of this table is the “Improvement”, where the numbers indicate the enhancement in time and upper bound that Meta-HBP achieves over U-HBP. Bold numbers in this table highlight the superior time and upper bound between the two methods.

Table 3. Numerical results of upper bound solutions

Ins	N-T	U-HBP				Meta-HBP			Improvement (%)		
		Deter gap	Time		Upper bound	Time		Upper bound	Time		Upper bound
			Full	Sol		Full	Sol		Full	Sol	
1	2-1	1e-04	3284.88	800	50296.9064	**1219.19** [18.52]	800 [0]	**50296.8658** [0]	**62.88%**	0.00%	**0.0001%**
2	2-2	1e-04	7329.09	1600	**50731.6537**	**1984.09** [17.03]	1600 [0]	50735.9738 [0]	**72.93%**	0.00%	-0.0085%
3	2-3	1e-04	11529.54	2400	**50921.7577**	**2859.98** [16.06]	2400 [0]	50929.4664 [0]	**75.19%**	0.00%	-0.0151%
4	6-1	1e-04	6369.99	2400	51113.0652	**3095.41** [22.85]	2400 [0]	**51099.7844** [0]	**51.41%**	0.00%	**0.0260%**
5	6-2	1e-04	7905.46	4800	**52673.6016**	**5590.13** [22.42]	4800 [0]	52695.6374 [19.22]	**29.29%**	0.00%	-0.0418%
6	6-3	1e-04	11618.82	7200	**102782.986**	**9789.57** [98.92]	7200 [0]	103346.3621 [3.8]	**15.74%**	0.00%	-0.5481%
7	10-1	0.015	11630.29	4000	**101974.411**	**4491.7** [16.39]	4000 [0]	102014.0908 [4.78]	**61.38%**	0.00%	-0.0389%
8	10-2	0.015	11935.18	8000	104004.149	**11519.69** [84.45]	8000 [0]	**103659.9084** [22.91]	**3.48%**	0.00%	**0.3310%**
9	10-3	0.015	12463.43	12000	107066.468	**7263.11** [122.71]	**6868.77** [115.55]	**106958.4349** [106.54]	**41.72%**	**42.76%**	**0.1009%**
10	20-5	0.1	**45499.4**	40000	218835.5894	45640.97114 [66.79]	40000 [0]	**218622.9971** [155.02]	-0.31%	0.00%	**0.0971%**

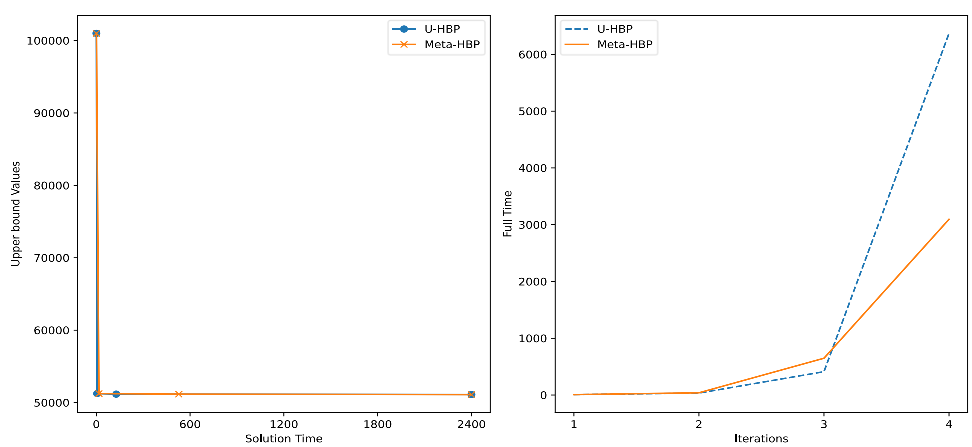
Figure 9 illustrates the iterations of instance 1 solved by U-HBP and Meta-HBP. Part (a) demonstrates the descending trend towards the upper bound and convergence to the optimal solution based on solution time. It is evident that the metaheuristic algorithm, Meta-HBP, outperforms the exact method, U-HBP, given the limitations on solution time. Part (b) of this figure depicts the full time running of both methods. According to this figure, Meta-HBP demonstrates a 62.88% improvement in total runtime compared to U-HBP.



(a) (b)

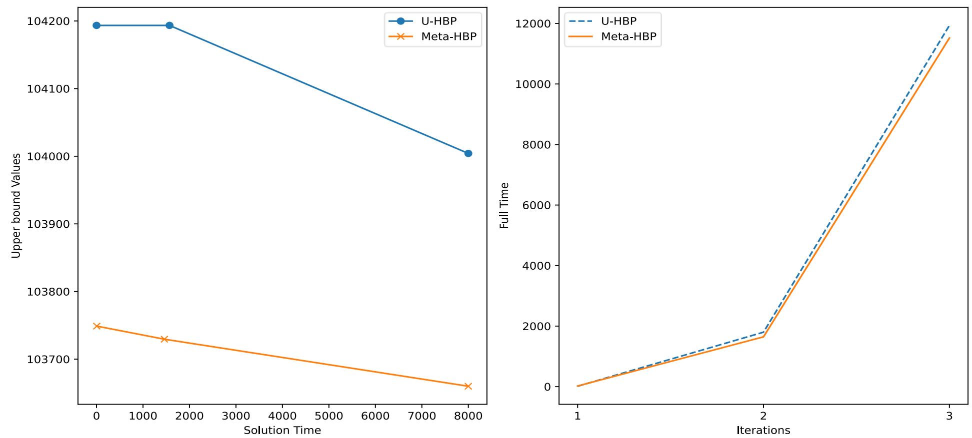
Figure 9. Solving Ins 1 using U-HBP and Meta-HBP ((a): Upper bound values, (b): Full Time)

As Table 3 indicates, instances 1 through 9 demonstrate an improvement in full time. In instance 10, a "Deter gap" of 0.1 has been considered to enable U-HBP to solve the problem, thus resulting in a shorter full time compared to Meta-HBP. However, this has also allowed Meta-HBP to achieve a better upper bound within the specified time frame. Additionally, improvements in the upper bound are observed in instances 1, 4, 8, 9, and 10, along with a noticeable enhancement in the solution time of instance 9. Instance 10 stands out as the sole instance that has achieved marginally better full time result in U-HBP. This can be attributed to the necessity of setting a "Dete gap" of 0.1, allowing U-HBP to solve the problem within the specified time limit. Figure 10 to Figure 12 depict the iterations of instances 4, 8, and 9.



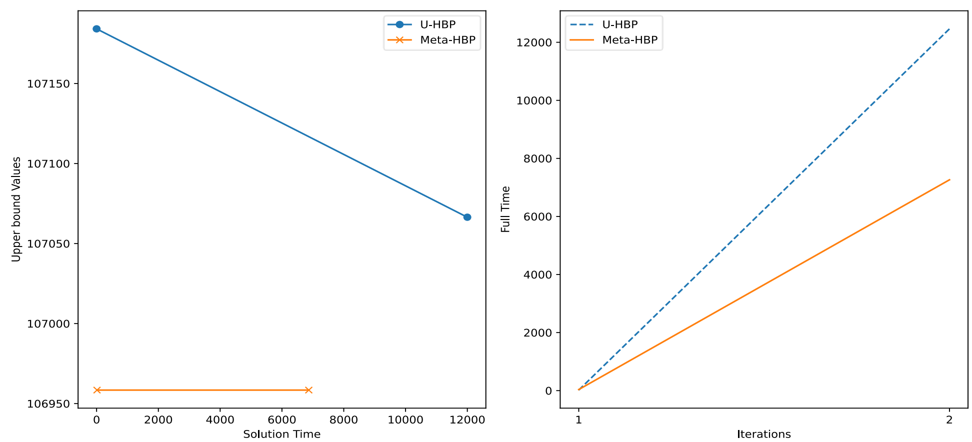
(a) (b)

Figure 10. Solving Ins 4 using U-HBP and Meta-HBP ((a): Upper bound values, (b): Full Time)



(a) (b)

Figure 11. Solving Ins 8 using U-HBP and Meta-HBP ((a): Upper bound values, (b): Full Time)



(a) (b)

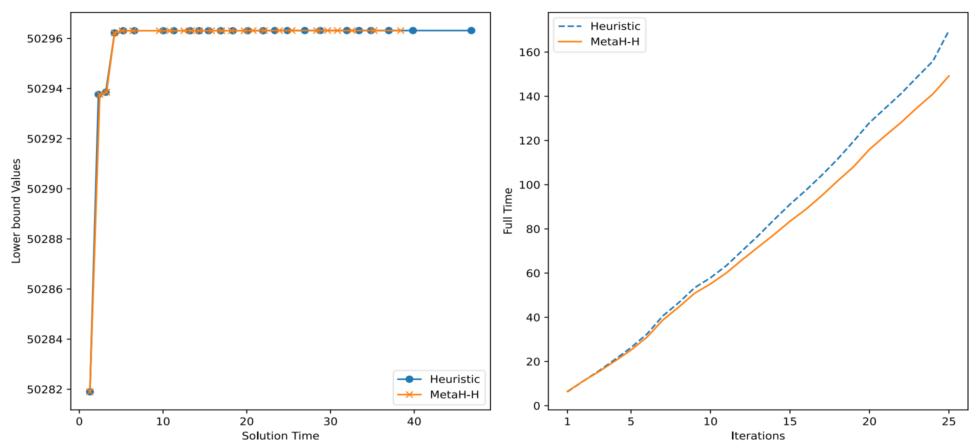
Figure 12. Solving Ins 9 using U-HBP and Meta-HBP ((a): Upper bound values, (b): Full Time)

Heuristic and MetaH-H are used to find a lower bound on the optimal solution of the ARO problem. Table 4 summarizes the results of applying these two methods to the instances. This table shares the same columns and instances as Table 3, with the distinction of presenting lower bounds.

Table 4. Numerical results of lower bound solutions

Ins	N-T	Deter gap	Heuristic Time Full	Heuristic Time Sol	Heuristic Lower bound	MetaH-H Time Full		MetaH-H Time Sol		MetaH-H Lower bound		Improvement (%) Time Full	Improvement (%) Time Sol	Improvement (%) Lower bound
1	2-1	1e-04	169.75	46.83	50296.315	**149.2**	[2.84]	**38.38**	[2.02]	50296.315	[0]	**12.11%**	**18.04%**	0.0000%
2	2-2	1e-04	2483.16	577.25	50643.7226	**2234.05**	[42.66]	**508.96**	[39.62]	50643.7226	[0]	**10.03%**	**11.83%**	0.0000%
3	2-3	1e-04	10615.58	**2334.35**	50806.2012	**9346.95**	[92.21]	2339.66	[52.03]	50806.2012	[0]	**11.95%**	-0.23%	0.0000%
4	6-1	1e-04	8008.48	**2255.02**	50608.9097	**7345.59**	[650.07]	2255.6	[185.98]	**50608.9475**	[0.05]	**8.28%**	-0.03%	**0.0001%**
5	6-2	1e-04	5710.7	1724.3	**51305.4459**	**4027.02**	[247.14]	**964.2**	[108.7]	51304.8451	[0.49]	**29.48%**	**44.08%**	-0.0012%
6	6-3	1e-04	23650.65	7200	51869.9201	**19145.31**	[1012.86]	**5047.3**	[369.08]	**51871.4166**	[1.49]	**19.05%**	**29.90%**	**0.0029%**
7	10-1	0.015	705.11	127.08	51320.3628	**684.17**	[8.48]	**122.31**	[6.33]	**51398.889**	[46.23]	**2.97%**	**3.75%**	**0.1530%**
8	10-2	0.015	17374.27	8000	52136.5334	**11033.72**	[735.06]	**1431.84**	[125.31]	**52232.3311**	[10.88]	**36.49%**	**82.10%**	**0.1837%**
9	10-3	0.015	35873.81	12000	53109.4428	**27977.04**	[1372.49]	**4371.18**	[359.23]	**53132.025**	[13.43]	**22.01%**	**63.57%**	**0.0425%**
10	20-5	0.1	**42188.9**	**6117.76**	61936.829	45440.75	[3258.52]	7908.48	[686.62]	**62082.59**	[20.66]	-7.71%	-29.27%	**0.2353%**

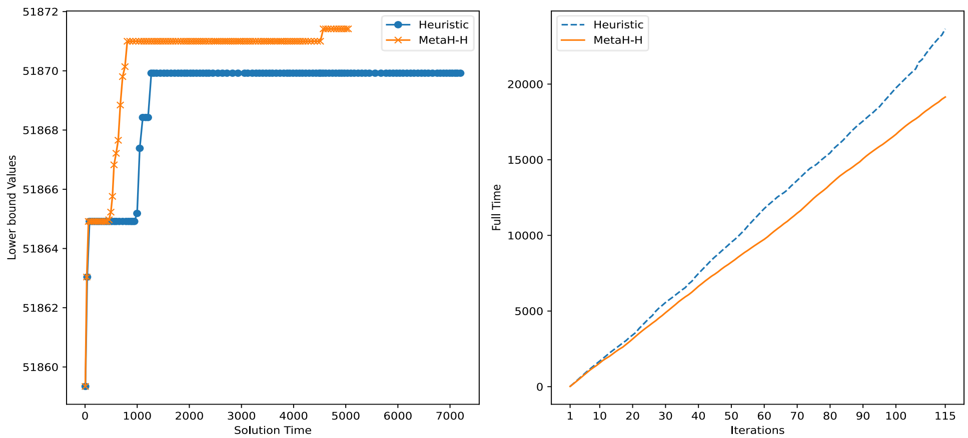
Figure 13 illustrates the iterations of instance 1 solved by Heuristic and MetaH-H. Part (a) displays the upward trend towards the lower bound and the convergence to the optimal solution based on solution time. As it can be seen, the metaheuristic algorithm, MetaH-H, attains the lower bound established by the Heuristic method while enhancing solution time by 18.04%. Part (b) of this figure depicts the full time running of both methods. According to this figure, MetaH-H demonstrates a 12.11% improvement in total runtime compared to Heuristic. This trend is almost true for instances 2 and 5 as well.



(a) (b)

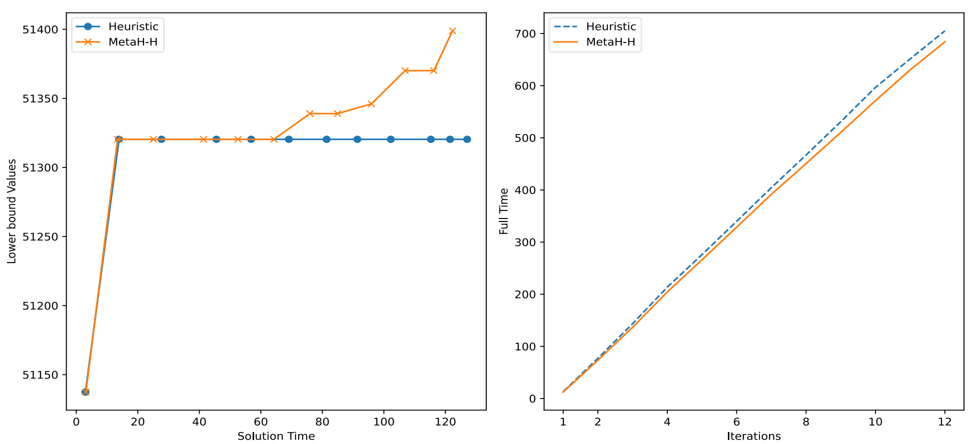
Figure 13. Solving Ins 1 using Heuristic and MetaH-H ((a): Lower bound values, (b): Full Time)

As evidenced in Table 4, improvements are observed in both the full time and solution time across instances 6 through 9, along with enhancements in the lower bound values. These improvements are illustrated in Figure 14 to Figure 17. Similar to Table 3, in Table 4, instance 10 has attained lower full and solution times due to the larger "Deter gap" compared to all other instances, which was necessary for the Heuristic method to solve the problem.



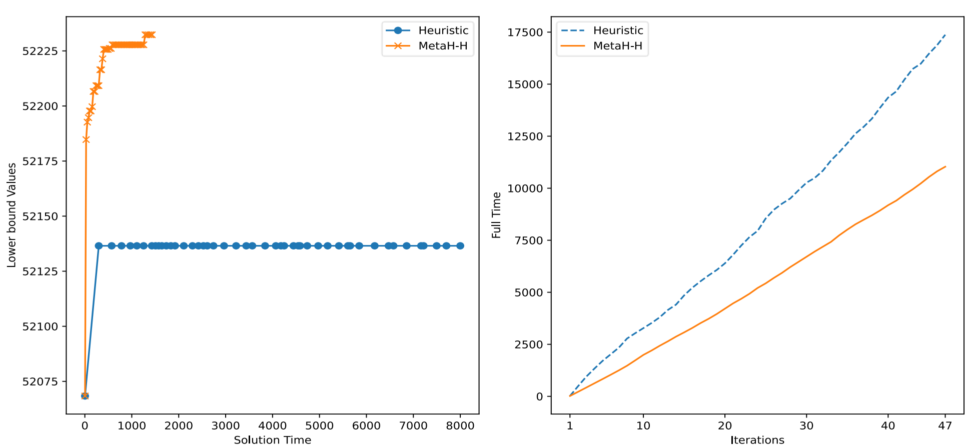
(a) (b)

Figure 14. Solving Ins 6 using Heuristic and MetaH-H ((a): Lower bound values, (b): Full Time)



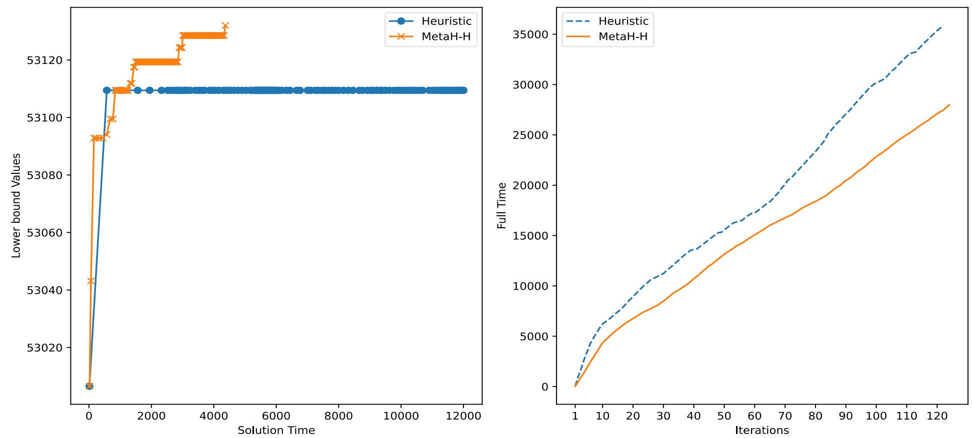
(a) (b)

Figure 15. Solving Ins 7 using Heuristic and MetaH-H ((a): Lower bound values, (b): Full Time)



(a) (b)

Figure 16. Solving Ins 8 using Heuristic and MetaH-H ((a): Lower bound values, (b): Full Time)



(a) (b)

Figure 17. Solving Ins 9 using Heuristic and MetaH-H ((a): Lower bound values, (b): Full Time)

8 Conclusions and future research

The Fleet Sizing Problem (FSP) emerges as a critical component in optimizing logistics and supply chain management, particularly in the realm of Closed-Loop Supply Chains (CLSC). Addressing the FSP directly impacts operational costs, resource utilization, and environmental sustainability, offering opportunities to streamline transportation operations, minimize fuel consumption, and enhance overall supply chain performance.

In this paper, we have delved into the complexities of the FSP within a CLSC context, proposed a Mixed Integer Linear Programming (MILP) model, and employed a multi-stage adjustable robust optimization (ARO) formulation to handle the uncertainties associated with demand for new products and requests for pickup of used products. We have reconfigured an exact oracle-based algorithm and a heuristic search algorithm to derive upper and lower bounds on the optimal solution of the ARO problem. Additionally, we have introduced a metaheuristic algorithm based on simulated annealing (SA) algorithm to function as the oracle. Through numerical experiments, we have illustrated the efficacy of our approach, demonstrating significant improvements in both computational efficiency and solution quality when integrated with existing methods.

For future research, considering environmental factors such as carbon emissions and sustainability metrics in fleet sizing decisions could contribute to more eco-friendly and socially responsible supply chain operations. Furthermore, beyond addressing the large number of deterministic problems tackled by our proposed method, the computation of leftover polytopes in large-scale problems remains a time-consuming part of the presented algorithm. Consequently, exploring more efficient methodologies to compute these leftovers necessitates further investigation in future research. Additionally, there is potential for research to extend beyond the current focus on CLSCs and encompass other echelons of the supply chain, including suppliers, recovery centers, and disposal centers, to further enhance overall supply chain efficiency and sustainability.

References

[1] G. B. Dantzig and J. H. Ramser, "The truck dispatching problem," *Management science,* vol. 6, no. 1, pp. 80-91, 1959.

[2] C. Lei, W.-H. Lin, and L. Miao, "A two-stage robust optimization approach for the mobile facility fleet sizing and routing problem under uncertainty," *Computers & Operations Research,* vol. 67, pp. 75-89, 2016.

[3] J. Dethloff, "Vehicle routing and reverse logistics: The vehicle routing problem with simultaneous delivery and pick-up," *OR-Spektrum,* journal article vol. 23, no. 1, pp. 79-96, February 01 2001, doi: 10.1007/pl00013346.

[4] J. Guo, X. Wang, S. Fan, and M. Gen, "Forward and reverse logistics network and route planning under the environment of low-carbon emissions: A case study of Shanghai fresh food E-commerce enterprises," *Computers & Industrial Engineering,* vol. 106, pp. 351-360, 2017/04/01/ 2017, doi: https://doi.org/10.1016/j.cie.2017.02.002.

[5] H. Guo, C. Li, Y. Zhang, C. Zhang, and Y. Wang, "A nonlinear integer programming model for integrated location, inventory, and routing decisions in a closed-loop supply chain," *Complexity,* vol. 2018, 2018.

[6] S. Deng, Y. Li, H. Guo, and B. Liu, "Solving a Closed-Loop Location-Inventory-Routing Problem with Mixed Quality Defects Returns in E-Commerce by Hybrid Ant Colony Optimization Algorithm," *Discrete Dynamics in Nature and Society,* vol. 2016, p. 6467812, 2016/06/30 2016, doi: 10.1155/2016/6467812.

[7] G. Iassinovskaia, S. Limbourg, and F. Riane, "The inventory-routing problem of returnable transport items with time windows and simultaneous pickup and delivery in closed-loop supply chains," *International Journal of Production Economics,* vol. 183, pp. 570-582, 2017/01/01/ 2017, doi: https://doi.org/10.1016/j.ijpe.2016.06.024.

[8] Y. Zhang, A. Che, and F. Chu, "Improved model and efficient method for bi-objective closed-loop food supply chain problem with returnable transport items," *International Journal of Production Research,* vol. 60, no. 3, pp. 1051-1068, 2022.

[9] Y. Zhang, F. Chu, and A. Che, "Closed-loop Inventory Routing Problem for Perishable Food with Multi-type Returnable Transport Items," *IFAC-PapersOnLine,* vol. 55, no. 10, pp. 2828-2833, 2022.

[10] Y. Qiu, M. Ni, L. Wang, Q. Li, X. Fang, and P. M. Pardalos, "Production routing problems with reverse logistics and remanufacturing," *Transportation Research Part E: Logistics and Transportation Review,* vol. 111, pp. 87-100, 2018/03/01/ 2018, doi: https://doi.org/10.1016/j.tre.2018.01.009.

[11] Q. Yuchi, N. Wang, Z. He, and H. Chen, "Hybrid heuristic for the location-inventory-routing problem in closed-loop supply chain," *International Transactions in Operational Research,* 2018.

[12] A. Pedram, S. Sorooshian, F. Mulubrhan, and A. Abbaspour, "Incorporating Vehicle-Routing Problems into a Closed-Loop Supply Chain Network Using a Mixed-Integer Linear-Programming Model," *Sustainability,* vol. 15, no. 4, p. 2967, 2023.

[13] M. Soysal, "Closed-loop Inventory Routing Problem for returnable transport items," *Transportation Research Part D: Transport and Environment,* vol. 48, pp. 31-45, 2016.

[14] M. Kumar, D. Kumar, P. Saini, and S. Pratap, "Inventory routing model for perishable products toward circular economy," *Computers & Industrial Engineering,* vol. 169, p. 108220, 2022.

[15] Y. Shuang, A. Diabat, and Y. Liao, "A stochastic reverse logistics production routing model with emissions control policy selection," *International Journal of Production Economics,* vol. 213, pp. 201-216, 2019.
[16] M. M. Nasiri, H. Mousavi, and S. Nosrati-Abarghooee, "A green location-inventory-routing optimization model with simultaneous pickup and delivery under disruption risks," *Decision Analytics Journal,* vol. 6, p. 100161, 2023.
[17] H. Gholizadeh, H. Jahani, A. Abareshi, and M. Goh, "Sustainable closed-loop supply chain for dairy industry with robust and heuristic optimization," *Computers & Industrial Engineering,* vol. 157, p. 107324, 2021.
[18] Z. Zarrat Dakhely Parast, H. Haleh, S. Avakh Darestani, and H. Amin-Tahmasbi, "Green reverse supply chain network design considering location-routing-inventory decisions with simultaneous pickup and delivery," *Environmental Science and Pollution Research,* pp. 1-22, 2021.
[19] K. Govindan, F. Salehian, H. Kian, S. T. Hosseini, and H. Mina, "A location-inventory-routing problem to design a circular closed-loop supply chain network with carbon tax policy for achieving circular economy: An augmented epsilon-constraint approach," *International Journal of Production Economics,* vol. 257, p. 108771, 2023.
[20] M. Tavana, H. Kian, A. K. Nasr, K. Govindan, and H. Mina, "A comprehensive framework for sustainable closed-loop supply chain network design," *Journal of Cleaner Production,* vol. 332, p. 129777, 2022.
[21] B. Golden, A. Assad, L. Levy, and F. Gheysens, "The fleet size and mix vehicle routing problem," *Computers & Operations Research,* vol. 11, no. 1, pp. 49-66, 1984.
[22] M. Desrochers and T. Verhoog, "A new heuristic for the fleet size and mix vehicle routing problem," *Computers & Operations Research,* vol. 18, no. 3, pp. 263-274, 1991.
[23] F.-H. Liu and S.-Y. Shen, "The fleet size and mix vehicle routing problem with time windows," *Journal of the Operational Research society,* vol. 50, pp. 721-732, 1999.
[24] M. Dell'Amico, M. Monaci, C. Pagani, and D. Vigo, "Heuristic approaches for the fleet size and mix vehicle routing problem with time windows," *Transportation Science,* vol. 41, no. 4, pp. 516-526, 2007.
[25] W. Dullaert, G. K. Janssens, K. Sörensen, and B. Vernimmen, "New heuristics for the fleet size and mix vehicle routing problem with time windows," *Journal of the Operational Research Society,* vol. 53, pp. 1232-1238, 2002.
[26] P. Belfiore and H. T. Yoshizaki, "Heuristic methods for the fleet size and mix vehicle routing problem with time windows and split deliveries," *Computers & Industrial Engineering,* vol. 64, no. 2, pp. 589-601, 2013.
[27] G. Hiermann, J. Puchinger, S. Ropke, and R. F. Hartl, "The electric fleet size and mix vehicle routing problem with time windows and recharging stations," *European Journal of Operational Research,* vol. 252, no. 3, pp. 995-1018, 2016.
[28] M. A. Masmoudi, M. Hosny, and Ç. Koç, "The fleet size and mix vehicle routing problem with synchronized visits," *Transportation Letters,* vol. 14, no. 4, pp. 427-445, 2022.
[29] U. Pasha, A. Hoff, and L. M. Hvattum, "Simple heuristics for the multi-period fleet size and mix vehicle routing problem," *INFOR: Information Systems and Operational Research,* vol. 54, no. 2, pp. 97-120, 2016.
[30] H. Asefi, S. Shahparvari, P. Chhetri, and S. Lim, "Variable fleet size and mix VRP with fleet heterogeneity in Integrated Solid Waste Management," *Journal of Cleaner Production,* vol. 230, pp. 1376-1395, 2019.
[31] C. E. Schmidt, A. C. Silva, M. Darvish, and L. C. Coelho, "Time-dependent fleet size and mix multi-depot vehicle routing problem," *International Journal of Production Economics,* vol. 255, p. 108653, 2023.
[32] R. Vanga and J. Venkateswaran, "Fleet sizing of reusable articles under uncertain demand and turnaround times," *European Journal of Operational Research,* vol. 285, no. 2, pp. 566-582, 2020.

[33] K.-H. Chang, Y.-C. Chiang, and T.-Y. Chang, "Simultaneous location and vehicle fleet sizing of relief goods distribution centers and vehicle routing for post-disaster logistics," *Computers & Operations Research,* vol. 161, p. 106404, 2024.

[34] A. Borumand, A. Marandi, A. S. Nookabadi, and Z. Atan, "An oracle-based algorithm for robust planning of production routing problems in closed-loop supply chains of beverage glass bottles," *Omega,* vol. 122, p. 102939, 2024.

[35] J. Lofberg, "YALMIP: A toolbox for modeling and optimization in MATLAB," in *2004 IEEE international conference on robotics and automation (IEEE Cat. No. 04CH37508)*, 2004: IEEE, pp. 284-289.