Optimization of Multi-Factory Remanufacturing Processes with Shared Transportation Resources Using the ALNS Algorithm

Jinlei Gu, Zeyu Guo, Jiacun Wang, Liang Qi, Shujin Qin and Shaoyu Zhang

***Abstract*—Decentralized manufacturing addresses growing challenges in centralized production by reducing costs for production, storage, and transportation through proximity to consumers. This study aims to optimize a multi-factory remanufacturing process, incorporating disassembly plants, manufacturing facilities, disassembly lines, and third-party logistics. The primary objective is to enhance system performance by balancing disassembly lines, optimizing transportation and routing, and minimizing workstation costs.**

Given the NP-hardness and computational complexity of the disassembly line balancing problem (DLBP) and the vehicle routing problem with pickup and delivery (VRPPD), this paper proposes a multi-objective optimization framework. The framework builds on existing disassembly strategies and employs the adaptive large neighborhood search (ALNS) algorithm to improve delivery and transportation efficiency while maximizing execution profit. To enhance practical applicability, the problem is systematically decomposed into two independent subproblems. A mixed-integer programming model is developed to optimize reverse supply chain performance and maximize profit. The model's feasibility and effectiveness are validated using the CPLEX solver, demonstrating its capability to address complex remanufacturing challenges.

***Key Words*—Multi-factory remanufacturing process optimization, Disassembly line balancing problem, Reverse supply chain optimisation, Vehicle routing problem with pickup and delivery, Adaptive large neighborhood search**

I. Introduction

In the current era of globalization, escalating environmental concerns and the imperative for resource sustainability are driving a transformative shift away from conventional linear business models towards more resource-efficient paradigms [1]. As a result, the optimization of cross-domain factory production and supply chain management is assuming paramount significance for sustainable business development [2, 3]. Within this multifaceted landscape, Multi-factory Remanufacturing processes offer a distinct advantage: they enable remote supervision and administration, underpinned by the pivotal role of networking technologies in allocating resources and equipment across heterogeneous locations [4]. This, in turn, results in enhanced productivity and cost-efficiency. Furthermore, these processes facilitate production across geographically dispersed locations, strengthening contingency management.

In recent years, significant attention has been devoted by the research community to the Reverse Supply Chain (RSC) network design problem and its various facets. The primary objective of RSC network optimization is to seamlessly coordinate the disassembly and remanufacturing of products, ultimately contributing to the sustainability and environmental image of the entire supply chain [5]. In contrast to extensively studied assembly processes, disassembly is characterized by its inherently volatile supply-side structure, necessitating effective management to mitigate avoidable costs and inefficiencies [6]. A comprehensive repository of the latest models and algorithms can be found in Akçalı *et al.*'s [7] exhaustive survey.As emphasized in the research by Paksoy *et al.* [8], it is evident that the simultaneous progress of RSC network optimization and disassembly line balancing is interdependent. The cycle times for disassembly and the total workstation count are intricately linked to variables like transportation volumes, facility capacity, collection/demolition rates, and fluctuating demand [9]. Therefore, addressing the challenges of RSC and DLBP in isolation is unfeasible, their integration is imperative. Özceylan *et al.*introduced a nonlinear mixed-integer programming model that strategically addresses the DLBP and the broader closed-loop supply chain network design problem [10]. This model places significant emphasis on optimizing the costs associated with transportation, procurement, and operations at disassembly workstations within the framework of both forward and reverse chains. However, research on the multi-factory remanufacturing processes optimization problem (MRPOP) is notably scarce. MRPOP is designed to coordinate and optimize the operations of multiple disassembly and remanufacturing factories within the RSC network. It's primary objective is to maximize the overall efficiency of the entire remanufacturing system while concurrently optimizing resource allocation, disassembly scheduling, and transportation route planning.

Manuscript received April 3, 2025; revised April 17, 2025; accepted April 18, 2025. This article was recommended for publication by Associate Editor Xiwang Guo upon evaluation of the reviewers' comments.

J. Gu is with the New Jersey of Institute of Technology, Newark, NJ, USA (e-mail: gujinlei0707@gmail.com).

Z. Guo is with the Computer and Communication Engineering College, Liaoning Petrochemical University, Fushun, 113001, China (e-mail: guozeyun@stu.lnpu.edu.cn).

J. Wang is with the Dept of Comp. Sci. and Soft. Eng. at Monmouth University, W. Long Branch, NJ, USA (e-mail: jwang@monmouth.edu).

L. Qi is with the Department of Computer Science and Technology at Shandong University of Science and Technology, Qingdao, 266590, China (qiliangsdkd@163.com).

S. Qin is with the College of Economics and Management, Shangqiu Normal University, Shangqiu 476000, China (e-mail: sjchin@vip.126.com).

S. Zhang is with Firm Foundation Christian School, Battle Ground, WA, USA (e-mail: yuaug11@gmail.com).

Corresponding Author: Liang Qi.

The rapid advancement of electronic technology has significantly improved people's quality of life. However, the frequent iteration and updates of electronic products have led to a substantial increase in end-of-life (EOL) products. Effectively managing these EOL products is of paramount importance, as highlighted by previous research [11, 12, 13, 14, 15, 16]. The crux of the matter lies in the DLBP [17, 18], which revolves around the optimal allocation of tasks at each station. Scholars have made significant contributions to DLBP research, classifying it into various types, including linear [19, 20], U-shaped [21, 22, 23], bilateral [24], and others. The importance of DLBP lies in its ability to minimize the number of workstations, equilibrium idle time, hazard index, demand index, and the change of disassembly direction, thereby enhancing the overall efficiency of disassembly while meeting demand requirements. Belassiria *et al.* [25] research the assembly line rebalancing problem by considering uncertain product demands and multi-skilled workers. They propose a mathematical model and embed a heuristic program into a genetic algorithm to maximize the production line efficiency. This work combines MRPOP with DLBP to propose a Multi-factory Remanufacturing-process-optimization Problem based on based on Vehicle Routing Problem with Pickup and Delivery. In MRPOP, disassembly and transportation are two inseparable key stages. These two major issues are interdependent and fully interactive.

Similar to the disassembly of products, factory routing and recycling transportation of components within MRPOP is critical.Introducing the vehicle routing problem (VRP) is necessary for constructing route plans for a homogeneous fleet of vehicles aiming at the shortest travel distances to serve a range of customers. In recent years, the VRP problem has been well-studied and has generated numerous variations such as the CVRP, Pickup-and-delivery problems for goods transportation (VRPPD), and The Vehicle Routing Problem with Time Windows (VRPTW) problems. The literature classifies VRPPD into three categories based on demand types and route structures: One-to-Many-to-One, Many-to-Many, and One-to-One [26].This work mainly uses the one-to-one Pickup and Delivery method. Some studies also consider the simultaneous optimization of VRP and other problems, such as production scheduling [27]. However, few studies have focused on integrating DLBP with traffic planning problems. In response to this gap, Diri Kenger *et al.* proposed the Integrated Demolition Line Balancing and Routing Problem (I-DLB-RP) [28]. This approach factors in the disassembly of EOL and combines it with VRP to chart the optimal route for transporting recyclable components to remanufacturing centers. It takes into account numerous constraints and factors to ensure the efficient distribution of recycled components, all while minimizing disassembly and transportation costs. This work proposes a combination of DLBP and VRPPD to optimize the multi-factory remanufacturing process. In this type of environment, the routing of disassembly/manufacturing plant sites and the logistics distribution of products are closely linked. This problem is essential in transportation as it aims to transport items efficiently with limited resources and time. This work combines MRPOP with DLBP and VRPPD to propose a Multi-factory Remanufacturing-process-optimization Problem based on Shared transportation and distribution service resources and discuss its specific application scenarios.

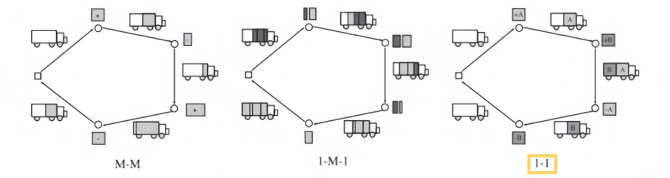


Fig. 1. The there types of PDPS.

Due to the NP-hardness and inapproximability of the DLBP and VRPPD, researchers often turn to heuristic algorithms. McGovern *et al.* [29] propose a genetic algorithm for DLBP, achieving optimal or near-optimal solutions. Feng *et al.* [30] present an enhanced multi-objective ant colony algorithm for disassembly sequence planning. Fu *et al.* [31] utilize the multiverse optimization algorithm for disassembly sequence planning, accounting for operational faults. These studies provide valuable insights and methodologies for tackling the MRPOP. The ALNS algorithm, a heuristic search approach for combinatorial optimization, amalgamates local and tabu search concepts. Its adaptability stems from dynamically tailored search strategies aligned with problem characteristics. Selected for MRPOP due to its efficiency, robustness, adaptability, and potent global search capabilities, it effectively balances exploration and exploitation tasks, yielding optimal solutions.

In this work, we propose a linear mixed-integer model for an MRPOP that encompasses multiple disassembly plants, manufacturing plants, disassembly lines, and third-party logistics providers. This model primarily focuses on task allocation for disassembly lines, aiming to reduce disassembly costs, idle time, and construct optimal routes for pickup and delivery in network. Moreover, it pays heed to service time and distance constraints with the dual objectives of reducing transportation costs and maximizing profitability. In summary, this study introduces a novel amalgamation of MRPOP and VRPPD into the realm of RSC network optimization—a realm hitherto underrepresented in existing literature.

The contributions of this work can be distilled into three primary facets:

1) Combining MRPOP with DLBP and VRRPD, a modeling method is described to integrate DLBP and VRRPD into the RSC network, minimizing the total cost of disassembly and transportation.

2) The problem is decomposed into two sub-problems. MRPOP is precisely solved with CPLEX. Further optimizations for delivery and transportation stages are achieved using the ALNS algorithm. Here, we extend the Shaw Move heuristic to consider membership levels and adopt a linear threshold acceptance criterion instead of simulated annealing.

3) Experimental validation of the model and algorithm effectiveness is conducted. Case testing with IBM CPLEX confirms model correctness. Optimization results and runtimes of ALNS and CPLEX are compared across various case

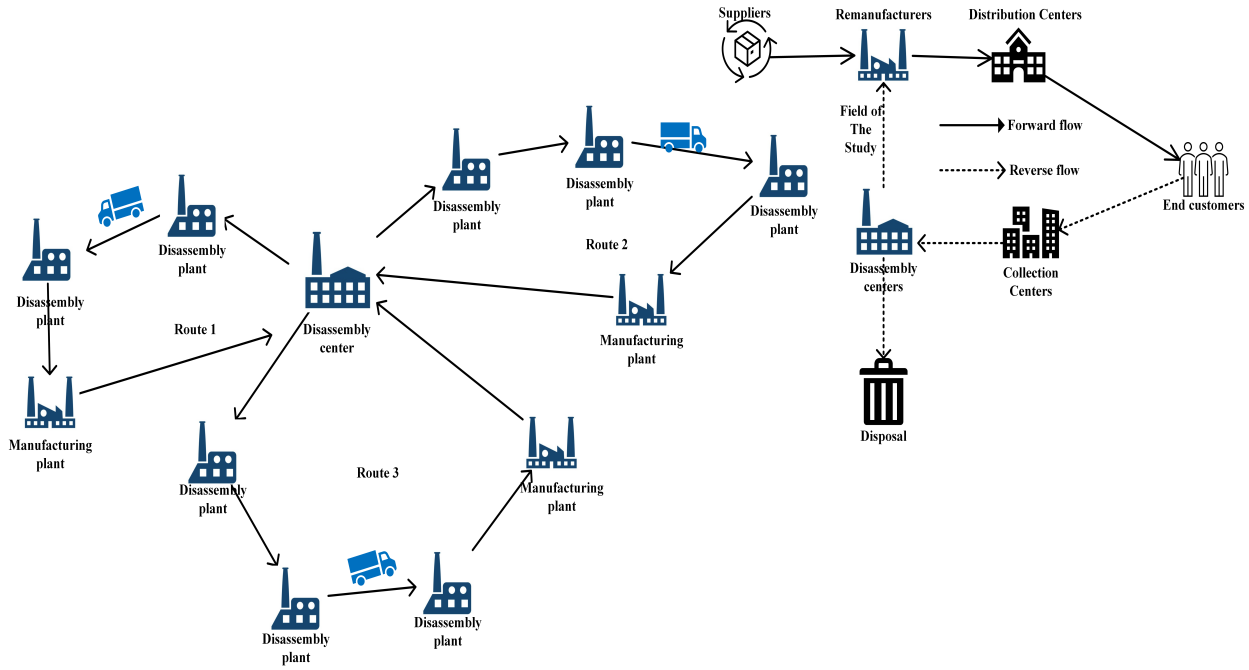


Fig. 2. The workflow of multi-factory remanufacturing process optimization.

sizes. Through extensive verification experiments, the ALNS algorithm's excellence and robustness in solving VRPPD are affirmed.

The remainder of the paper is organized as follows. Section II delves into the problem and formulates a mathematical model. Section III addresses the vehicle path planning problem. Section IV presents experimental designs and results. Finally, Section V provides a summary of the work presented in this paper.

II. Problem Description

A. Problem Statement

This section examines the integration of the DLBP and the VRPPD in MRPOP. Multi-factory remanufacturing is part of the RSC network, encompassing disassembly plants, manufacturing plants, workstations, and transportation vehicles. It aims to address various challenges including factory location, product positioning, sequencing of disassembly processes, product remanufacturing, and vehicle routing. Disassembly tasks are overseen by disassembly plants, with the disassembled subcomponents then transported to the manufacturing plants by means of vehicles. An important focal point in optimizing the RSC network is effectively designing efficient reverse logistics channels. This entails determining the number and location of disassembly and manufacturing plants, optimizing the sequence of disassembly tasks, allocating workstations, setting capacity restrictions for third-party logistics vehicle fleets, establishing service times for task nodes, and managing the flow and distance between delivery tasks.As shown in Fig. 2, The problem can be divided into three stages:

1) Disassembly plant selection and disassembly scheduling: Multiple disassembly and manufacturing plants are strategically designed and positioned across different locations. Disassembly plants consist of multiple workstations and disassembly lines, which can concurrently carry out disassembly tasks for different products. By efficiently allocating these tasks to workstations based on task priority and workstation cycle time limits, an optimal solution for the incomplete disassembly problem involving linear disassembly lines can be achieved.

2) Remanufacturing plant selection:

This stage focuses on component allocation. Based on prices of sub-components by different manufacturing plants and the distances between disassembly and manufacturing plants, the sub-components obtained from disassembling different products are assigned to appropriate manufacturing plants.

3) Planning optimal transportation routes:

This stage tackles the transportation of disassembled parts, specifically the VRPPD problem. The objective is to plan the most efficient routes from the disassembly plants to the manufacturing plants, ensuring the successful transport of all disassembled parts and satisfying the requirements of all factories along the routes. Considerations include vehicle capacity restrictions and service times, where each part request must be picked up and delivered by the same vehicle. In addition, the load of each transportation node must comply with the maximum capacity limit of the visiting vehicle. Loads can be combined and transported together to minimize transportation costs. Service time is defined as the shortest time between arrival at each node and departure, encompassing the time for loading and unloading goods. For warehouses, the service time is set at 0.

The objective of the experiments conducted is to minimize disassembly and transportation costs while maximizing overall profit. To address the incomplete disassembly problem involving linear disassembly lines, a linear mixed

integer programming model is utilized, while the VRPPD problem is optimized using the ALNS algorithm to enhance the transportation phase of the parts. This section provides a comprehensive description of the problem assumptions and mathematical formulas.

B. AND/OR Graph

During the dismantling process of EOL products, prioritization relationships exist between different dismantling tasks. DLBP aims to allocate dismantling tasks to workstations while satisfying the prioritization relationship between tasks. This task allocation process yields a dismantling sequence that satisfies both the objective function and constraints. Since the dismantling system is a discrete event system, researchers utilize Petri nets, precedence diagrams, and other formal methods to model and analyze the dismantling process and resource requirements. This study utilizes AND/OR graphs to describe task relationships.

Fig. 3 shows the schematic diagram of the fuel pump [32], which consists of 13 sub-components, whose AND/OR diagram is shown in Fig. 4. In the AND/OR diagram, the subassemblies are represented by nodes, the indexes of the subassemblies are represented by integers in pointed brackets, and each disassembly task is represented by a directed edge between the linked subassemblies. It is not difficult to find that this product has 30 disassembly tasks.

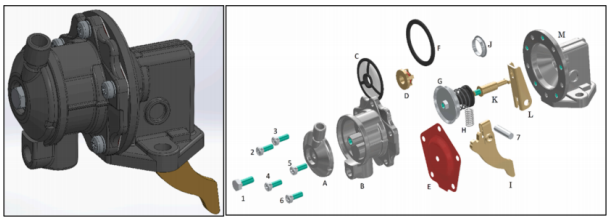


Fig. 3. A schematic of the fuel pump.

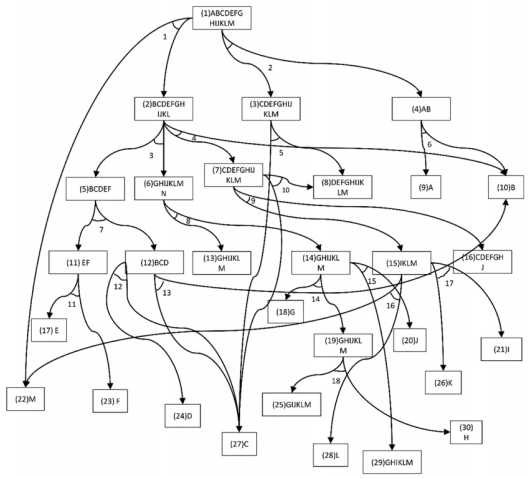


Fig. 4. The AND/OR graph of the fuel pump.

To calculate the profits of subassemblies, three matrices are used to describe the relationship between disassembly tasks and subassemblies.

1) Incidence matrix

The incidence matrix $D = [d_{pij}]$ describes the relationship between subassemblies and disassembly tasks, where i represent the subassemblies , j represent the task and p represents the product number.

$$d_{pij} = \begin{cases} 1, & \text{if subassembly } i \text{ is obainted by task } j \\ & \text{in product } p \ ; \\ -1, & \text{if subassembly } i \text{ is disassembled by task } j \\ & \text{of product } p \ ; \\ 0, & \text{otherwise.} \end{cases}$$

2) Conflict matrix

The conflict matrix $R = [r_{pj_1j_2}]$ describes the conflicting relationship between two tasks, where j_1 and j_2 represent the disassembly task, p represent the product number.

$$r_{pj_1j_2} = \begin{cases} 1, & \text{if task } j_1 \text{ of product } p \text{ has a conflicting} \\ & \text{relationship with } j_2 \ ; \\ 0, & \text{otherwise.} \end{cases}$$

3) Precedence matrix

The precedence matrix $S = [s_{pj_1j_2}]$ describes the relationship between two tasks,where j_1 and j_2 represent the disassembly task, p represents the product number.

$$s_{pj_1j_2} = \begin{cases} 1, & \text{if task } j_1 \text{ can be executed before task } j_2 \\ & \text{in product } p \ ; \\ 0, & \text{otherwise.} \end{cases}$$

The basic assumptions of the model are as follows:

1) The weight and yield of all subcomponents of the product are known
2) The demand for the components in each plant is determined and must be satisfied
3) The matrices D, S, and R are known
4) The disassembly process is partial
5) Each vehicle has a maximum capacity limit, requiring that the vehicle load after visiting each node does not exceed its maximum capacity
6) The service times of the transport vehicles when visiting the task nodes are known
7) Load requests at all task nodes in the network are known, with positive and negative indicating pickups/collections
8) The operating time of each operating workstation should not exceed the cycle time of the dismantling plant where it is located
9) The time spent on the disassembly task and the cost per unit of time are known
10) Each disassembly task can be processed on any workstation
11) To calculate the benefits of disassembled subassemblies, two matrices are used to describe the relationship between disassembled subassemblies and tasks.

C. Notations

This section shows a linear mixed integer programming model to solve the problem of simultaneously optimizing DLB and VRPPD in an RSC network. The notation and decision variables in the mathematical model satisfying all the above decisions and assumptions are defined as follows:

Sets:

$\mathbb{K}$ Set of disassembly plant indices.
$\mathbb{M}$ Set of manufacturing plant indices.
$\mathbb{P}$ Set of product indices.
$\mathbb{I}_p$ Set of all subassemblies in product p.
$\mathbb{J}_p$ Set of all tasks in production p.
$\mathbb{W}^k$ Set of linear workstations for the kth disassembly plant.
$\mathbb{P}_n$ Set of all pickup nodes in the network.
$\mathbb{D}_n$ Set of all delivery nodes in the network.
$\mathbb{N}'$ Set of all task nodes. $\mathbb{N}' = \{1, 2, \ldots, 2N\}$
$\mathbb{V}$ Set of transport vehicles. $\mathbb{V} = \{1, 2, \ldots, V\}$

Indexes:

p Product index, $p \in \mathbb{P}$.
i subassemblies index, $i \in \mathbb{I}_p$.
j Disassembly task index, $j \in \mathbb{J}_p$.
w Disassembly task index, $j \in \mathbb{W}_p$.
k Disassembly factory index. $k \in \mathbb{K}$.
m Manufacturing factory index. $m \in \mathbb{m}$.
v Transport vehicle index, $v \in \mathbb{V}$.

Parameters:

v_{mpi} The the price at which manufacturing plant m acquires component i of product p.
c_{km}^T Transportation cost from disassembly plant k to manufacturing plant m based on distance.
d_{ef} The Euclidean distance from node e to node f.
t_{ef} The Traveling time from node e to node f.
Q^v Indicates maximum load capacity of vehicle v.
d_{pij} An element in the i-th row and j-th column of D.
R Dismantling conflict matrix.
$r_{pj_1j_2}$ An element in the i-th row and j-th column of R.
S Dismantling Timing Relationship.
$s_{pj_1j_2}$ An element in the i-th row and j-th column of S.
c_k^O Unit time cost of startup the k-th disassembly factory.
c_{kw}^S Cost of startup the w-th linear workstation of the k-th disassembly factory.

Decision variables:

$$z_{pk} = \begin{cases} 1, & \text{If product } p \text{ is assigned to disassembly factory } k; \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{pjkw} = \begin{cases} 1, & \text{If task } j \text{ of product } p \text{ is assigned to a linear workstation } w \text{ in disassembly factory } k; \\ 0, & \text{otherwise.} \end{cases}$$

$$y_k = \begin{cases} 1, & \text{If the linear disassembly line of the disassembly factory } k \text{ is opened;} \\ 0, & \text{otherwise.} \end{cases}$$

$$u_{kw} = \begin{cases} 1, & \text{If the linear workstation } w \text{ on the disassembly factory } k \text{ is opened;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\alpha_{kmpi} = \begin{cases} 1, & \text{subassembly } i \text{ of product } p \text{ is shipped from disassembly factory } k \text{ to manufacturing factory } m; \\ 0, & \text{otherwise.} \end{cases}$$

$$\theta_{efv} = \begin{cases} 1, & \text{If vehicle v travelled from node e to node f;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\beta_{km} = \begin{cases} 1, & \text{Indicates that disassembly plant k to manufacturing plant m has distribution tasks;} \\ 0, & \text{otherwise.} \end{cases}$$

$$q_e \begin{cases} > 0, & \text{Denotes the pick up request of the task node;} \\ < 0, & \text{Denotes the delivery request of the task node.} \end{cases}$$

W_{kmpi}, denotes the weight of component i of product p shipped from disassembly plant K to manufacturing plant M.
Q_e^v, denotes the current load after vehicle v has visited node e
Q_{km}, denotes the load demand from dismantling plant k to at manufacturing plant m.
t_e^v, denotes the time after vehicle v has visited node e.
T_k, Cycle time of disassembly factory.

The above explains the specific meaning of the relevant variables and decision variables, and the objective function of the mathematical model and the constraints are shown below.

D. Mathematical Model

The linear mixed integer mathematical model equation satisfying all the above decisions and assumptions is defined as follows:

$$\begin{aligned} max & \sum_{k\in\mathbb{K}} \sum_{m\in\mathbb{M}} \sum_{p\in\mathbb{P}} \sum_{i\in\mathbb{I}^p} \left(v_{mpi} - c_{km}^T\right) \alpha_{kmpi} \\ & - \sum_{v\in\mathbb{V}} \sum_{e\in\mathbb{N}} \sum_{f\in\mathbb{N}} d_{ef}\theta_{efv} \\ & - \sum_{k\in\mathbb{K}} \sum_{p\in\mathbb{P}} \sum_{j\in J^p} \sum_{w\in\mathbb{W}^k} c_{kpj}^D t_{kpj} x_{pjkw} \\ & - \sum_{k\in\mathbb{K}} c_k^O T_k - \sum_{k\in\mathbb{K}} \sum_{w\in\mathbb{W}^k} c_{kw} u_{kw} \end{aligned} \tag{1}$$

$$\begin{aligned} & \sum_{m\in\mathbb{M}} \alpha_{kmpi} \le \sum_{w\in\mathbb{W}^k} \sum_{j\in\mathbb{J}^p} d_{pij} x_{pjkw}, \\ & \forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall i \in \mathbb{I}^p \setminus \{1\} \end{aligned} \tag{2}$$

$$\sum_{k\in\mathbb{K}} z_{pk} = 1, \forall p \in \mathbb{P} \tag{3}$$

$$z_{pk} \le y_k, \forall w \in \mathbb{W}^k, \forall k \in \mathbb{K} \tag{4}$$

$$u_{kw} \le y_k, \forall w \in \mathbb{W}^k, \forall k \in \mathbb{K} \tag{5}$$

$$x_{pjkw} \le z_{pk}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall w \in \mathbb{W}^k, \forall k \in \mathbb{K} \tag{6}$$

$$x_{pjkw} \le u_{kw}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, \forall w \in \mathbb{W}^k, \forall k \in \mathbb{K} \tag{7}$$

$$\sum_{k\in\mathbb{K}} \sum_{w\in\mathbb{W}^k} x_{pjkw} \le 1, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p \tag{8}$$

$$\sum_{j\in\mathbb{J}_p} t_{pj} x_{pjkw} \le T^k, \forall k \in \mathbb{K}, \forall w \in \mathbb{W}^k \tag{9}$$

$$\begin{aligned} &\sum_{w\in\mathbb{W}^k} w\left(x_{pjkw} - x_{pqkw}\right) + \\ &W^k\left(\sum_{w\in\mathbb{W}^k} x_{pqkw} - 1\right) \le 0, \\ &\forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j, q \in \mathbb{J}_p, s_{pjq} = 1 \end{aligned} \tag{10}$$

$$\begin{aligned} &\sum_{w\in\mathbb{W}^k} x_{pqkw} \le \sum_{j\in\mathbb{J}_p} \sum_{w\in\mathbb{W}^k} x_{pjkw} s_{pjq}, \\ &\forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j, q \in \mathbb{J}_p, d_{piq} = 0 \end{aligned} \tag{11}$$

$$\begin{aligned} &\sum_{w\in\mathbb{W}^k} \left(x_{pjkw} + x_{pqkw}\right) \le 1, \\ &\forall k \in \mathbb{K}, \forall p \in \mathbb{P}, \forall j, q \in \mathbb{J}_p, r_{pjq} = 1 \end{aligned} \tag{12}$$

$$z_{pk} \in \{0,1\}, \forall p \in \mathbb{P}, \forall k \in \mathbb{K} \tag{13}$$

$$x_{pjkw} \in \{0,1\}, \forall p \in \mathbb{P}, \forall j \in \mathbb{J}_p, k \in \mathbb{K}, w \in \mathbb{W}_k \tag{14}$$

$$y_k \in \{1,0\}, \forall k \in \mathbb{K} \tag{15}$$

$$u_{kw} \in \{1,0\}, \forall k \in \mathbb{K}, \forall w \in \mathbb{W} \tag{16}$$

$$\alpha_{kmpi} \in \{1,0\}, \forall k \in \mathbb{K}, \forall m \in \mathbb{M}, \forall p \in \mathbb{P}, \forall i \in \mathbb{I} \tag{17}$$

$$T_k \in \mathbb{R}_+, \forall k \in \mathbb{K} \tag{18}$$

$$\sum_{p\in\mathbb{P}} \sum_{i\in\mathbb{I}^p} w_{kmpi} = q_e, \quad \forall k \in \mathbb{K}, \forall m \in \mathbb{M}, \forall e \in \mathbb{P}_n, \beta_{km} = 1 \tag{19}$$

$$q_e \le \beta_{km} * M, \forall e \in \mathbb{P}_N \tag{20}$$

$$\theta_{efv} = 1, \forall e \in \mathbb{P}_N, \beta_{km} = 1 \tag{21}$$

$$\sum_{f\in\mathbb{N}} \theta_{fev} - \sum_{f\in\mathbb{N}} \theta_{efv} = 0, \quad \forall e \in \mathbb{N}', \forall v \in \mathbb{V} \tag{22}$$

$$\sum_{f\in\mathbb{P}_n+2*n+1} \theta_{0fv} = 1, \forall v \in \mathbb{V} \tag{23}$$

$$\sum_{e\in\mathbb{D}_n+0} \theta_{e,2*n+1,v} = 1, \forall v \in \mathbb{V} \tag{24}$$

$$\sum_{f\in\mathbb{N}} (\theta_{efv} - \theta_{f,n+e,v}) = 0, \quad \forall e \in \mathbb{P}_n, \forall v \in \mathbb{V} \tag{25}$$

$$Q_e^v + q_f - Q_f^v \le M * (1 - \theta_{efv}), \ \forall e \in \mathbb{N}, \forall f \in \mathbb{N}, \forall v \in \mathbb{V} \tag{26}$$

$$q_e \le Q_e^v \le Q_v, \ \forall v \in \mathbb{V} \tag{27}$$

$$0 \le Q_e^v \le Q_{v+q_e}, \ \forall v \in \mathbb{V} \tag{28}$$

$$Q_0^v = 0, \ \forall v \in \mathbb{V} \tag{29}$$

$$T_e^v + t_{ef} - T_f^v \le M * (1 - \theta_{efv}), \ \forall e \in \mathbb{N}, \forall f \in \mathbb{N}, \forall v \in \mathbb{V} \tag{30}$$

$$T_e^v + t_{e,n+e} \le T_{n+e}^v, \ \forall e \in \mathbb{P}_n, \forall f \in \mathbb{N}, \forall v \in \mathbb{V} \tag{31}$$

$$\theta_{efv} \in \{1,0\}, \forall e \in \mathbb{N}, \forall f \in \mathbb{N}, \forall v \in \mathbb{V} \tag{32}$$

$$Q_e^v \ge 0, \forall e \in \mathbb{N}, \forall f \in \mathbb{V} \tag{33}$$

$$T_e^v \ge 0, \forall e \in \mathbb{N}, \forall f \in \mathbb{V} \tag{34}$$

The objective function (1) represents the maximum expected profit of the entire network, comprising two main components. The first component minimizes the dismantling cost to enhance the recycling revenue of EOL products. The dismantling cost includes the fixed cost of operating dismantling factories and associated workstations, as well as the cost required for executing the dismantling tasks. The second component aims to minimize the transportation cost of transferring sub-components from the dismantling factory to the manufacturing factory, thereby maximizing the overall profit. Constraint (3) specifies that only the components obtained from dismantling can be allocated from the dismantling factory to the remanufacturing factory. Constraint (4) ensures that each product can only be assigned to a specific dismantling factory. Constraints (5)-(7) indicate that products can only be assigned to workstations in operational dismantling factories. Constraint (8) stipulates that each dismantling task for a product can only be executed once. Constraint (9) limits the work duration of each workstation on a dismantling line to the assigned cycle time. Constraints

(10)-(11) require that the allocation of dismantling tasks for each product align with their internal complexity, meeting their prioritization constraints. Constraint (12) ensures that the assignment of dismantling tasks satisfies the constraint on conflicting relationships. Constraints (13)-(18) define the ranges of decision variables. Constraint (19) determines the load request of the pickup nodes. (20) Connecting the two symbol systems, where $\beta_{km} = 1$ represents a distribution relationship between dismantling factory k and manufacturing factory m, M is a sufficiently large number. Constraint (21) ensures that each request is accessed only once. Constraint (22) guarantees flow conservation and path continuity, stating that a vehicle must both enter and leave node e. Constraints (23) and (24) ensure that each vehicle departs from the starting point and arrives at the destination. Constraint (25) ensures that pickup and delivery nodes associated with the same request are accessed by the same vehicle. Constraint (26) ensures correct load updates along the route of each vehicle, where M is a sufficiently large positive integer. Constraints (27) and (28) respectively prevent vehicles from violating the maximum capacity constraints of the pickup and delivery nodes. Constraint (29) mandates that the initial load of each vehicle is 0. Constraint (30) ensures that the arrival time at subsequent nodes is at least the sum of the arrival time and the travel time between the two points, where M is a sufficiently large positive integer. Constraint (31) ensures that pickup nodes are serviced prior to the delivery nodes associated with the same request. Lastly, constraints (32)-(34) represent the ranges of decision variables

III. Proposed Algorithm

The ALNS framework, originating from Shaw's Large Neighborhood Search (LNS) in 1998, iteratively improves solution quality by employing single removal and insertion heuristics. Unlike traditional local search, LNS focuses on exploring a limited solution space while extensively searching the vicinity through optimal elimination and reinsertion of node groups. Bent and Van Hentenryck (2006) achieved satisfactory results with LNS on VRPPDTW, highlighting the significant impact of the insertion heuristic on solution quality. To address this, Ropke and Pisinger (2006) introduced the ALNS algorithm as an alternative, enabling the use of straightforward and fast insertion heuristics while maintaining performance. In contrast to LNS's fixed heuristics, ALNS randomly selects from a predefined set of removal and insertion heuristics at each iteration. Pisinger and Ropke (2007) confirmed ALNS's robustness in handling various VRP versions, including VRPPD.

We propose the ALNS framework with two modifications based on Ropke and Pisinger (2006):

1) extending the Shaw Removal heuristic to consider membership status.
2) applying the Linear Threshold Acceptance criterion instead of Simulated Annealing.

The ALNS framework, outlined in Algorithm 1, begins with constructing a feasible initial solution in line 1. Lines 2-3 initialize a set of removal and insertion operators. The algorithm iterates *maxIter* times. At each iteration, ALNS removes and reinserts q requests from the solution in lines 7-8, guided by heuristics discussed in Sections Removal Heuristics and Sections Insert Heuristics. Lines 9-11 update the best-so-far solution s^* if a new solution s' has a superior objective. Lines 12-14 update the current solution s if the acceptance criterion, discussed in Section Acceptance Criterion from Linear Threshold Acceptance, is met. The search process is segmented into segments, each comprising *segIter* iterations. At the end of each segment, operators update their scores in line 16.

Algorithm 1 ALNS Framework

```
1: s* ← feasible initial solution
2: REM ← set of removal operators
3: INS ← set of insertion operators
4: q ← number of requests to be removed
5: for i := 0; i < maxIter; i++ do
6:     choose removal and insertion operator from REM and INS.
7:     (s', R_removed) ← Remove(s, q)
8:     s' ← Insert(s', R_removed)
9:     if f(s') < f(s*) then
10:        s* ← s'
11:    end if
12:    if acceptance criterion met then
13:        s ← s'
14:    end if
15:    if maxIter mod segIter = 0 then
16:        update scores for operators in REM and INS
17:    end if
18: end for
19: return s*
```

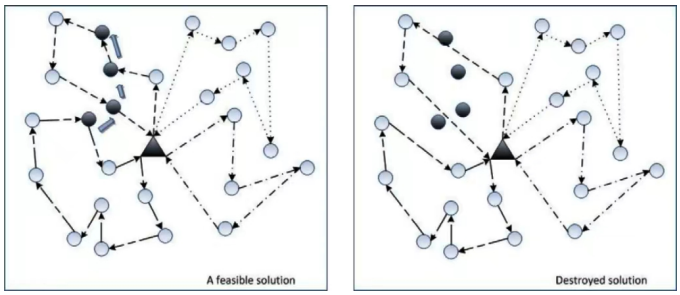


Fig. 5. Insert Delete Operation.

A. Removal Heuristics

This section introduces three distinct removal heuristics: 1) Random Removal, 2) Shaw Removal with Priority, 3) Worst Removal. Each heuristic employs a unique criterion for selecting requests to be removed and defines the function Remove(s, q) in Algorithm 1. This function, taking a feasible solution s and an integer q as input, yields the partial solution s' with q requests removed and the set of removed requests R_{removed}.

Compared to removing one or two requests from each route, it is preferable to simultaneously remove requests that are

"close" to each other. Shaw Removal quantifies the similarity between request pairs and aims to remove requests with higher similarity within one iteration.

Algorithm 2 Shaw Removal with Priority ($Solution\ s$, q)

1: $r \leftarrow$ request randomly selected from s
2: $R_{removed} \leftarrow \{r\}$
3: **while** size($R_{removed}$) $< q$ **do**
4: $r' \leftarrow$ request randomly selected from $R_{removed}$
5: $L \leftarrow$ list of requests r not in $R_{removed}$
6: $L \leftarrow L$ sorted by ascending $R(r, r')$
7: $y \leftarrow$ Random(0,1)
8: $r \leftarrow$ request at $y^p|L|$ position in L
9: $R_{removed} \leftarrow R_{removed} \cup \{r\}$
10: **end while**
11: $s' \leftarrow$ Remove(s, $R_{removed}$)
12: **return** s', $R_{removed}$

Request cost r is defined as the difference between the total objective with and without the pickup-delivery pair associated with r, denoted as $\Delta fr = f(s) - f_{-r}(s)$. Here, s represents the current solution, f denotes the objective function, and $f^{-r}(s)$ signifies the objective after removing request r from s. A larger Δfr value indicates that request r is in an unfavorable position, suggesting a higher chance of finding an improved solution by replacing such requests.

The Worst Removal algorithm is tailored to identify and eliminate requests with significant Δfr values in each iteration. Algorithm 3 outlines the algorithmic details, which share a similar approach with the Shaw Removal with Priority algorithm. However, the key difference lies in sorting the requests in each iteration based on descending order of request cost.

Similar to the Shaw Removal with Priority algorithm, the Worst Removal algorithm incorporates a randomness parameter denoted as p to introduce randomness. This randomness is crucial to prevent the algorithm from repeatedly attempting to remove the same set of requests in consecutive iterations, thereby ensuring effective exploration of the neighborhood space

Algorithm 3 Worst Removal($Solution\ s$, q)

1: $R_{removed} \leftarrow \emptyset$
2: **while** size($R_{removed}$) $< q$ **do**
3: $L \leftarrow$ list of request r not in $R_{removed}$
4: $L \leftarrow$ sort L by descending Δfr
5: $y \leftarrow$ Random$(0, 1)$
6: $r \leftarrow$ request at $y^p|L|$ position in L
7: $R_{removed} \leftarrow R_{removed} \cup \{r\}$
8: **end while**
9: $s' \leftarrow$ Remove$(s, R_{removed})$
10: **return** $s', R_{removed}$

B. Insert Heuristics

This section delineates various insert heuristics, each introducing a rule to determine the positions for requests to be inserted and specifying the function Insert(s', R_{removed}) in Algorithm 1. These heuristics take a partial solution s' and a set of requests R_{removed} as input, and output a solution with requests reinserted.

The Simple Greedy Insert (SGI) algorithm aims to identify the best request in terms of objective increase at each iteration. In this algorithm, the insertion cost Δfr_k represents the minimum objective increase resulting from including request r in route k. Algorithm 4 outlines the process to achieve this objective. Initially, in lines 2-6, we calculate the costs associated with inserting each request in each route. Subsequently, in lines 10-12, we include request r^* with the lowest insertion cost in the partial solution and remove it from the unprocessed requests set. If some requests remain unprocessed, we obtain an empty solution, indicating the algorithm's failure to reconstruct a feasible solution, as depicted in lines 14-18. Otherwise, our goal is accomplished, and the resulting solution is denoted as s'.

The SGI heuristic demonstrates limited foresight as it only considers a single step post-insertion. This tendency often results in "bad" requests being neglected, leading to their insertion with relatively high costs towards the process's end when options are scarce. In contrast, the Regret Insert strategy tackles this issue by extending its evaluation to multiple subsequent steps. Specifically, the Regret-m Insert method analyzes m additional steps to make more informed decisions.

The regret value quantifies the challenge posed by delaying the insertion of a request and potentially inserting it in subsequent iterations. The heuristic prioritizes requests yielding the highest regret value. In case of ties, preference is given to requests with lower insertion costs. Notably, the SGI corresponds to the Regret-1 Insert due to its tie-breaking mechanism.

To address requests with limited feasible insertion routes, we set $\Delta fr_{k,n} = M$. This prioritizes requests with fewer feasible insertion routes. The overall process closely resembles that of the Simple Greedy Insert, wherein we select the request maximizing the regret value c_r, as demonstrated in Algorithm 4, line 10.

In this study, the set of insertion operators comprises Regret-1 (Simple Greedy), Regret-2, Regret-3, Regret-4, and Regret-$|K|$.

Algorithm 4 Simple Greedy Insert($Solution\ s'$, $R_{removed}$)

1: **while** $R_{removed} \neq \emptyset$ **do**
2: **for** each request r in $R_{removed}$ **do**
3: **for** each route k in solution s **do**
4: calculate Δ_{frk}
5: **end for**
6: **end for**
7: **if** no feasible insertion **then**
8: break
9: **end if**
10: $r^* \leftarrow \arg \min_{r \in R_{removed}} \left(\min_{k \in K} \Delta_{frk} \right)$
11: insert request r^* in route k at position with minimum objective value increase
12: remove request r^* from $R_{removed}$
13: **end while**
14: **if** $R_{removed} = \emptyset$ **then**

15: **return** s'
16: **else**
17: **return** $\emptyset$
18: **end if**

C. Acceptance Criterion from Linear Threshold Acceptance

The acceptance criterion employed in this study originates from linear Threshold Acceptance (TA), with the end temperature set to 0. T serves as the temperature threshold governing acceptable objective gaps. In each iteration, an improved neighboring solution is invariably accepted, while a deteriorated solution is accepted if $\frac{f(s')-f(s^*)}{f(s^*)} < T$ holds true. Initially, T is set to T_{start}, and it decreases at a rate of $\Delta T = \frac{T_{\text{start}}}{\text{maxIter}}$ in each iteration.

Linear TA was first introduced by Dueck and Scheuer (1990) and has demonstrated enhanced computational efficiency and the ability to yield high-quality solutions. Another commonly employed acceptance criterion stems from simulated annealing (SA), where superior solutions are always accepted, and deteriorated solutions are accepted with a probability of $e^{-\frac{f(s')-f(s)}{T}}$. Santini et al. (2018) have noted comparable performance between both criteria when applied within the Adaptive Large Neighborhood Search (ALNS) framework. The selection of TA for this study was driven by (1) the straightforward comparison of acceptance with the threshold T, eliminating the need for random number generation and probability computation, and (2) the direct calculation of the cooling rate through tuning a single parameter, T_{start}.

IV. Experimental Studies

A. Experimental Cases and Parameter Settings

To assess the accuracy of the model and the effectiveness of the proposed algorithm, the experimental cases were solved using IBM ILOG CPLEX Optimization Studio to obtain the standard optimal solution. Additionally, ALNS was employed to solve the same cases in order to compare the experimental results. The computational experiments were carried out on a computer with an Intel(R) Core(TM) i9-13900HX processor running at 5.40GHz and equipped with 32.00GB of RAM.

In order to make the experimental study more comprehensive, three distinct product sizes were chosen: washing machine, computer, radio. These products were combined in different configurations to create multiple product cases for testing.Table I provides the specific size information of the combined cases. Regarding the work environment,three disassembly factories and three manufacturing factories were established, each with a maximum of five workstations on the disassembly lines within the factories.The specific parameter settings are shown in Table I.

B. Model Validation and Analysis

We used CPLEX to test experimental cases, and the results are shown in Table II. In the table, "w" represents washing machines, "c" represents computers, and "r" represents radios. Taking Case 6 as an example: washing machines are assigned to Disassembly Plants 2 and 3 for disassembly, with components recovered by Remanufacturing Plants M1, M2, and M3; computers are assigned to Disassembly Plant 3, with components recovered by Manufacturing Plants 2 and 3; radios are assigned to Disassembly Plants 1 and 2, with components solely recovered by Manufacturing Plant 1. For instance, the disassembly results of radio P8 show that it is assigned to Disassembly Plant K1, with disassembled parts 9, 19, 23, 24, and 26 subsequently transported to Remanufacturing Plant M1. The total weight of the transported parts is 79.

In the MRPOP analysis, Case 6 involves 10 products: 3 washing machines, 3 computers, and 4 radios. The experimental results are shown in Table II and Table III. A comparison reveals that the results from CPLEX only allocate and transport disassembled parts based on the profit provided by the manufacturing plants, indicating a significant weakness in CPLEX in the comprehensive transportation process. Figure 7 shows the optimal solution obtained by re-inputting the CPLEX output into MRPOP as the parameters required for solving the two-stage VRPPD model. Comparing Figures 6 and 7 clearly shows that the two-stage transportation model optimization significantly reduces delivery costs, total vehicle distance, and the number of vehicles required compared to the single-stage CPLEX solution. This alignment better meets the actual efficiency requirements of a multi-factory remanufacturing environment.

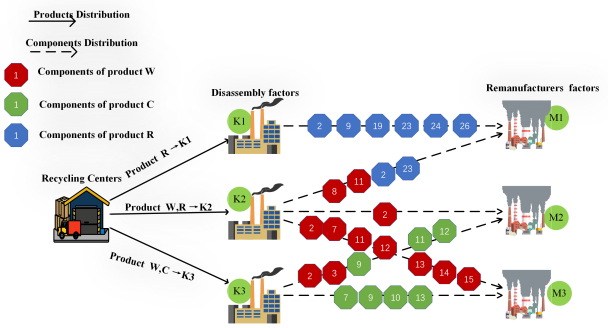


Fig. 6. Factory distribution method for case 1.

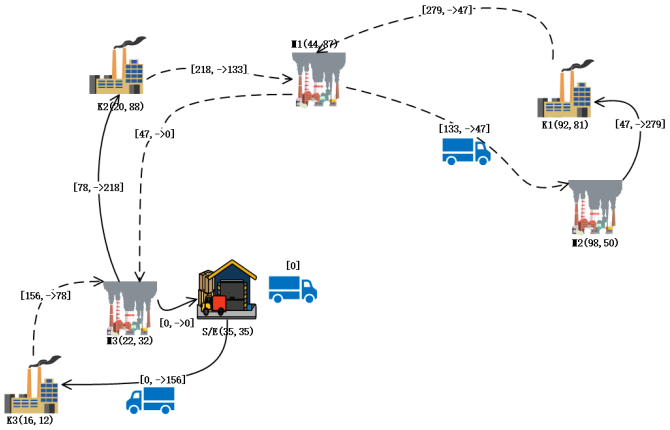


Fig. 7. Optimal delivery solution for case1.

TABLE I Product parameter set.

Product	Num. of parts	Num. of task	Num of subassembly	Subassembly profit	Component weight	Disassembly cost	Disassembly time
Washing machine	6	13	15	106 ~ 179	3 ~ 33	3 ~ 8	4 ~ 10
comuputer	5	13	25	102 ~ 166	6 ~ 41	2 ~ 7	5 ~ 11
Radio	10	30	29	63 ~ 162	5 ~ 74	2 ~ 6	4 ~ 11

TABLE II CPLEX solutions 1

Product ID	Disassembly allocation	Remanufacturing allocation	Component weight
P1	K_2	$(< 2 \rightarrow M_2 > / < 7, 11, 12 \rightarrow > M_3)$	8/25
P2	K_3	$(< 2, 3 \rightarrow M_2 >)$	33
P3	K_2	$(< 8, 11 \rightarrow M_1 > / < 2, 13, 14, 15 \rightarrow M_3 >)$	11/22
P4	K_3	$(< 9, 12 \rightarrow M_2 > / < 7, 10 \rightarrow M_3 >)$	19/22
P5	K_3	$(< 9, 11, 12 \rightarrow M_2 > / < 10, 13 \rightarrow M_3 >)$	26/15
P6	K_3	$(< 5, 9, 13 \rightarrow M_3 >)$	41
P7	K_2	$(< 2, 23 \rightarrow M_1 >)$	74
P8	K_1	$(< 9, 19, 23, 24, 26 \rightarrow M_1 >)$	79
P9	K_1	$(< 2, 23 \rightarrow M_1 >)$	74
P10	K_1	$(< 9, 19, 23, 24, 26 \rightarrow M_1 >)$	79

TABLE III Experimental parameter setting

Request	Task Node	Demand	Service Time	Location
	S	0	0	(35, 35)
R_1	K_1	232	10	(92, 81)
	M_1	-232	10	(44, 87)
R_2	K_2	85	10	(20, 88)
	M_1	-85	10	(44, 87)
R_3	K_2	8	10	(20, 88)
	M_2	-8	10	(98, 50)
R_4	K_2	47	10	(20, 88)
	M_3	-47	10	(22, 32)
R_5	K_3	78	10	(16, 12)
	M_2	-78	10	(98, 50)
R_6	K_3	78	10	(16, 12)
	M_3	-78	10	(22, 32)
	E	0	0	(35, 35)

C. Comparison with CPLEX OPL solver

To examine the performance of the proposed ALNS framework to VRPPD, we generated and evaluated smaller instances. The results were compared with those obtained by CPLEX within a 60-minute time constraint. The experiments utilized N pairs of randomly extracted requests from the adjusted instances, forming a smaller dataset. Experiments were conducted for instances with N values of 5, 10, and 15, and the results are displayed in Tables IV, V, VI.

For $N = 5$, CPLEX's computation time slightly outperformed ALNS, but ALNS still achieved optimal solutions for all instances at a relatively fast speed. When $N = 10$, ALNS found superior solutions within 5 seconds for six cases, with the best performance in case 5, where the target value computation time was 183.116 times longer than ALNS. In two cases, CPLEX failed to find the optimal solution within the 60-minute time constraint. Among the cases where the optimal solution was found, ALNS achieved speeds 145.844 to 2264.15 times faster than CPLEX. In other cases, both ALNS and CPLEX were able to find optimal solutions, with ALNS being faster. For $N = 15$, CPLEX was only able to solve 5 cases, while ALNS found feasible solutions for all instances without a notable increase in computation time. In case 3, CPLEX's objective value exhibited significant differences from ALNS, effectively demonstrating the algorithm's superiority. As ALNS yielded identical results to the successfully found optimal objectives in all instances in OPL, we conclude that our ALNS is robust when applied to the model proposed in our study. Additionally, in the experiments with $N = 10$ and 15, ALNS required much less time than CPLEX, and the difference in computation time increased as the problem size grew. We have reason to believe that ALNS has higher computational efficiency in experiments with larger problem sizes.

D. Verification of Algorithm With Different Scales of Cases

To evaluate the efficacy and robustness of the ALNS framework in addressing the VRPPD problem,we conducted experiments on all instances of size 100 from the dataset by Li and Lim (2003). We evaluated all deletion operators

TABLE IV Comparison of results between CPLEX and ALNS N=5

Test Instance	Objective		Time(s)	
	CPLEX	ALNS	CPLEX	ALNS
1	196.499	196.499	0.693s	1.145s
2	265.647	265.647	0.479s	1.139s
3	211.094	211.094	0.204s	0.712s
4	275.335	275.335	0.305s	1.001s
5	518.822	518.822	0.207s	0.825s
6	356.407	356.407	0.506s	1.229s

TABLE V Comparison of results between CPLEX and ALNS N=10

Test Instance	Objective		Time(s)	
	CPLEX	ALNS	CPLEX	ALNS
1	781.062	763.768	*3600s	4.158s
2	603.877	603.877	*3600s	1.59s
3	354.714	354.714	1860.5s	0.97s
4	547.470	547.470	720.47s	4.94s
5	433.965	433.965	300.31s	1.64s
6	326.235	326.235	1260.59s	3.595s

within ALNS, including Random Destructor, Worst Destructor, and Shaw Destructor, along with various rebuilding strategies utilizing Regret-M Rebuilder.

For each operator and case, we performed five independent experiments, recording the best objective value achieved in each trial and calculating their average. We analyzed five distinct operator combination strategies and present the average objective values for each case in the table VII below. Overall, different operator combination strategies exhibited similar performance across cases. The Random Destructor demonstrated wider exploration abilities but was prone to local optima. The Shaw Destructor showed a tendency to converge to better solutions more quickly, though it might be influenced by the initial solution quality. On the other hand, the Worst Destructor had the potential to escape local optima faster in certain scenarios, albeit with increased computational cost.

TABLE VI Comparison of results between CPLEX and ALNS N=15

Test Instance	Objective		Time(s)	
	CPLEX	ALNS	CPLEX	ALNS
1	572.504	557.224	*3600s	1.112s
2	———	659.206	*3600s	1.963s
3	2824.479	611.233	*3600s	5.598s
4	902.876	544.052	*3600s	4.909s
5	1053.413	986.798	*3600s	1.824s
6	1216.911	1049.695	*3600s	3.150s

V. Conclusions

The multi-factory remanufacturing processes optimization problem is a research hotspot in the field of supply chain. This paper proposes, for the first time, the MRPOP considering distribution services and establishes a mixed integer programming model aiming to maximize profit to describe this problem. We systematically decompose the problem into two sub-problems: disassembly scheduling and transportation route planning. Furthermore, we enhance the ALNS algorithm to better explore optimal solutions and avoid local optima. To validate the model's accuracy, we utilize IBM CPLEX to solve the model on small-scale instances and conduct algorithm comparisons. In large-scale cases, we employ the modified ALNS algorithm, demonstrating its efficiency.

However, several limitations warrant further discussion. First, the model assumes static and known parameters such as demand, disassembly task times, and transportation times. In real-world applications, these inputs are often uncertain or subject to dynamic changes. Inaccurate forecasts or real-time travel delays could significantly degrade system performance. Additionally, while the model performs well on benchmark scenarios, its scalability and responsiveness in real-time environments with dynamic task arrivals and disruptions remains an open challenge.

To address these limitations, future work will explore incorporating stochastic elements and real-time data streams into the modeling framework. Approaches such as robust optimization or scenario-based planning could help mitigate uncertainty. Moreover, applying reinforcement learning to adapt vehicle routing policies or hybrid heuristic/metaheuristic strategies could allow the system to respond dynamically to changes in demand and system states. Finally, integrating customer prioritization—based on service-level agreements or value-based criteria—could further refine task scheduling and delivery strategies, enhancing both efficiency and customer satisfaction.

References

[1] S. Qin, S. Zhang, J. Wang, S. Liu, X. Guo, and L. Qi, "Multiobjective multiverse optimizer for multirobotic u-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 2, pp. 882–894, 2023.

[2] S. Wang, J. Wan, D. Li, and C. Zhang, "Implementing smart factory of industrie 4.0: an outlook," *International journal of distributed sensor networks*, vol. 12, no. 1, p. 3159805, 2016.

[3] F. T. S. Chan and S. H. Chung, "Distributed scheduling in multiple-factory production with machine maintenance," *Process planning and scheduling for distributed manufacturing*, pp. 243–267, 2007.

[4] X. Wang, M. Zhou, Q. Zhao, S. Liu, X. Guo, and L. Qi, "A branch and price algorithm for crane assignment and scheduling in slab yard," *IEEE Transactions on Automation Science and Engineering*, vol. 18, no. 3, pp. 1122–1133, 2020.

[5] A. Mishra, P. Dutta, S. Jayasankar, P. Jain, and K. Mathiyazhagan, "A review of reverse logistics and closed-loop supply chains in the perspective of circular economy," *Benchmarking: An International Journal*, vol. 30, no. 3, pp. 975–1020, 2023.

[6] M. Ullah, "Impact of transportation and carbon emissions on reverse channel selection in closed-loop supply chain management," *Journal of Cleaner Production*, vol. 394, p. 136370, 2023.

[7] E. Akçalı and S. Çetinkaya, "Quantitative models for inventory and production planning in closed-loop supply chains," *International Journal of Production Research*, vol. 49, no. 8, pp. 2373–2407, 2011.

TABLE VII Algorithm performance comparison

Case ID	Optimal					Vehicle					Calculation time(s)				
	Random -M	Worst -1	Worst -M	Shaw -1	Shaw -M	Random -M	Worst -1	Worst -M	Shaw -1	Shaw -M	Random -M	Worst -1	Worst -M	Shaw -1	Shaw -M
lr111	1307.54	1307.55	1307.59	1308.99	1308.78	13.6	13	13.4	13	13.6	13.45	21.69	22.12	22.13	13.87
lr112	1726.34	1766.51	1730.71	1759.16	1735.75	20	20.4	20	20	20	7.10	7.75	13.08	13.14	9.76
lr201	1323.81	1347.44	1327.74	1334.62	1332.29	5.2	5.4	5.4	5.4	5	47.85	40.60	48.01	59.55	65.10
lr202	1391.14	1425.33	1407.37	1401.78	1399.12	5.8	5.2	5.4	5	5	63.16	66.87	69.04	99.33	70.46
lr203	1162.48	1186.56	1173.87	1173.42	1164.94	4	4	4.2	4.8	4	100.68	104.18	95.07	109.26	90.72
lr204	1111.17	1126.78	1129.11	1145.73	1114.66	4	4	4	4	4	68.51	75.41	71.01	81.01	68.84
lr205	1179.23	1218.59	1206.09	1194.89	1182.33	4	4.2	4.2	4	4	81.71	124.77	110.58	103.82	82.70
lr206	1219.59	1248.81	1217.44	1229.76	1224.17	5.6	5.2	5.4	5.2	4.8	79.43	58.44	56.82	58.96	54.75
lr207	1232.56	1296.53	1236.35	1240.78	1687	4.2	5	4.6	4.6	4.2	71.21	73.51	73.50	68.34	74.17
lr208	1071.28	1101.20	1073.41	1084.03	1070.83	4	4.4	4.2	4.4	4.6	88.67	85.13	89.34	81.55	83.23
lr209	1113.55	1140.08	1120.98	1122.25	1115.24	5.2	5.2	4.8	4.6	5	76.51	77.81	73.44	63.63	70.81
lr210	1083.11	1088.78	1092.99	1090.10	1082.64	4	4	4	4	4	112.98	69.23	70.32	69.12	82.73
lr211	1372.63	1402.99	1378.90	1378.71	1382.67	6	6	6	6	6	60.63	45.31	39.64	37.34	38.37

[8] E. Özceylan, T. Paksoy, and T. Bektaş, "Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing," *Transportation research part E: logistics and transportation review*, vol. 61, pp. 142–164, 2014.

[9] G. Xu, Z. Zhang, Z. Li, X. Guo, L. Qi, and X. Liu, "Multi-objective discrete brainstorming optimizer to solve the stochastic multiple-product robotic disassembly line balancing problem subject to disassembly failures," *Mathematics*, vol. 11, no. 6, p. 1557, 2023.

[10] E. Özceylan, T. Paksoy, and T. Bektaş, "Modeling and optimizing the integrated problem of closed-loop supply chain network design and disassembly line balancing," *Transportation research part E: logistics and transportation review*, vol. 61, pp. 142–164, 2014.

[11] X. Guo, S. Liu, M. Zhou, and G. Tian, "Dual-objective program and scatter search for the optimization of disassembly sequences subject to multiresource constraints," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 3, pp. 1091–1103, 2017.

[12] S. Qin, J. Li, J. Wang, X. Guo, S. Liu, and L. Qi, "A salp swarm algorithm for parallel disassembly line balancing considering workers with government benefits," *IEEE Transactions on Computational Social Systems*, 2023.

[13] X. Guo, Z. Zhang, L. Qi, S. Liu, Y. Tang, and Z. Zhao, "Stochastic hybrid discrete grey wolf optimizer for multi-objective disassembly sequencing and line balancing planning in disassembling multiple products," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 1744–1756, 2021.

[14] X. Cui, X. Guo, M. Zhou, J. Wang, S. Qin, and L. Qi, "A discrete whale optimization algorithm for disassembly line balancing with carbon emission constraint," *IEEE Robotics and Automation Letters*, 2023.

[15] X. Guo, T. Wei, J. Wang, S. Liu, S. Qin, and L. Qi, "Multiobjective u-shaped disassembly line balancing problem considering human fatigue index and an efficient solution," *IEEE Transactions on Computational Social Systems*, 2022.

[16] X. Guo, M. Zhou, A. Abusorrah, F. Alsokhiry, and K. Sedraoui, "Disassembly sequence planning: A survey," *IEEE/CAA Journal of Automatica Sinica*, vol. 8, no. 7, pp. 1308–1324, 2021.

[17] A. Güngör and S. M. Gupta, "Disassembly line in product recovery," *International Journal of Production Research*, vol. 40, no. 11, pp. 2569–2589, 2002.

[18] N. Deniz and F. Ozcelik, "An extended review on disassembly line balancing with bibliometric & social network and future study realization analysis," *Journal of Cleaner Production*, vol. 225, pp. 697–715, 2019.

[19] C. B. Kalayci, A. Hancilar, A. Gungor, and S. M. Gupta, "Multi-objective fuzzy disassembly line balancing using a hybrid discrete artificial bee colony algorithm," *Journal of Manufacturing Systems*, vol. 37, pp. 672–682, 2015, reverse Supply Chains.

[20] M. L. Bentaha, O. Battaïa, and A. Dolgui, "Lagrangian relaxation for stochastic disassembly line balancing problem," *Procedia Cirp*, vol. 17, pp. 56–60, 2014.

[21] S. Avikal, R. Jain, P. Mishra, and H. Yadav, "A heuristic approach for u-shaped assembly line balancing to improve labor productivity," *Computers Industrial Engineering*, vol. 64, no. 4, pp. 895–901, 2013.

[22] S. Qin, S. Zhang, J. Wang, S. Liu, X. Guo, and L. Qi, "Multi-objective multi-verse optimizer for multi-robotic u-shaped disassembly line balancing problems," *IEEE Transactions on Artificial Intelligence*, 2023.

[23] Z. Li, I. Kucukkoc, and Z. Zhang, "Iterated local search method and mathematical model for sequence-dependent u-shaped disassembly line balancing problem," *Computers Industrial Engineering*, vol. 137, p. 106056, 2019.

[24] J. Liang, S. Guo, B. Du, Y. Li, J. Guo, Z. Yang, and S. Pang, "Minimizing energy consumption in multi-objective two-sided disassembly line balancing problem with complex execution constraints using dual-individual simulated annealing algorithm," *Journal of Cleaner Production*, vol. 284, p. 125418, 2021.

[25] I. Belassiria, M. Mazouzi, S. ELfezazi, A. Cherrafi, and Z. ELMaskaoui, "An integrated model for assembly line re-balancing problem," *International Journal of Production Research*, vol. 56, no. 16, pp. 5324–5344, 2018.

[26] P. Toth and D. Vigo, *Vehicle routing: problems, methods, and applications*. SIAM, 2014.

[27] F. Qiu, N. Geng, and H. Wang, "An improved memetic algorithm for integrated production scheduling and vehicle routing decisions," *Computers & Operations Research*, vol. 152, p. 106127, 2023.

[28] Z. Diri Kenger, Ç. Koç, and E. Özceylan, "Integrated disassembly line balancing and routing problem," *International Journal of Production Research*, vol. 58, no. 23, pp. 7250–7268, 2020.

[29] S. M. McGovern and S. M. Gupta, "A balancing method and genetic algorithm for disassembly line balancing," *European journal of operational research*, vol. 179, no. 3, pp. 692–708, 2007.

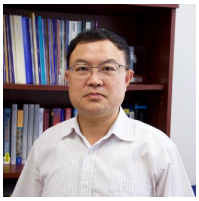
[30] Y. Feng, M. Zhou, G. Tian, Z. Li, Z. Zhang, Q. Zhang, and J. Tan, "Target disassembly sequencing and scheme evaluation for cnc machine tools using improved multiobjective ant colony algorithm and fuzzy integral," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 12, pp. 2438–2451, 2018.

[31] Y. Fu, M. Zhou, X. Guo, L. Qi, and K. Sedraoui, "Multiverse optimization algorithm for stochastic biobjective disassembly sequence planning subject to operation failures," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 52, no. 2, pp. 1041–1051, 2021.

[32] M. L. Bentaha, A. Dolgui, O. Battaïa, R. J. Riggs, and J. Hu, "Profit-oriented partial disassembly line design: dealing with hazardous parts and task processing times uncertainty," *International Journal of Production Research*, vol. 56, no. 24, pp. 7220–7242, 2018.

Jinlei Gu received his B.S. degree in computer science from Changshu Institute of Technology, China, in 2021, M.S. degree in computer science, from Monmouth University, New Jersey, U.S., in 2023. He is currently a Ph.D student in the New Jersey Institute of Technology. His research interests include reinforcement learning and intelligent optimization algorithms.

Zeyu Guo received his B.S. in IoT Engineering from The College of Institute of Disaster Prevention, China in 2022. Currently, he is a graduate student in the School of Computer and Communication Engineering at Liaoning University of Petroleum and Chemical Technology in Fushun, China. His research interests include Vehicle routing problems and intelligent optimization algorithms.



Jiacun Wang received the Ph.D. degree in computer engineering from Nanjing University of Science and Technology (NUST), China, in 1991. He is currently a Professor of software engineering at Monmouth University, West Long Branch, New Jersey, USA. From 2001 to 2004, he was a member of scientific staff with Nortel Networks in Richardson, Texas. Prior to joining Nortel, he was a research associate of the School of Computer Science, Florida International University at Miami. His research interests include software engineering, discrete event systems, formal methods, machine learning, and real-time distributed systems. He authored Timed Petri Nets: Theory and Application Kluwer, 1998), Real-time Embedded Systems (Wiley, 2018) and Formal Methods in Computer Science (CRC, 2019), edited Handbook of Finite Stat Based Models and Applications (CRC, 2012), and published over 130 research papers in journals and conferences. Dr. Wang was an Associate Editor of IEEE Transactions on Systems, Man and Cybernetics, Part C, and is currently Associate Editor of IEEE/CAA Journal of Automatica Sinica. He has served as general chair, program chair, and special sessions chair or program committee member for many international conferences. He is a senior member of IEEE.

Liang Qi received his B.S. degree in Information and Computing Science and M.S. degree in Computer Software and Theory from Shandong University of Science and Technology, Qingdao, China, in 2009 and 2012, respectively, and Ph.D. degree in Computer Software and Theory from Tongji University, Shanghai, China in 2017. He is currently an associate professor of Computer Science and Technology at Shandong University of Science and Technology, Qingdao, China. From 2015 to 2017, he was a visiting student in the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ, USA. He has authored over 100+ technical papers in journals and conference proceedings, including IEEE Transactions on System, Man and Cybernetics: Systems, IEEE Transactions on Intelligent Transportation Systems, and IEEE/CAA Journal of Automatica Sinica. He received the Best Student Paper Award-Finalist in the 15th IEEE International Conference on Networking, Sensing and Control (ICNSC'2018). His current research interests include Petri nets, discrete event systems and optimization algorithms.

Shujin Qin received his B.S. degree in Information and Computing Science from Tianjin Polytechnic University, Tianjin, China in 2008, M.S. degree in Operational Research and Cybernetics from University of Science and Technology Liaoning, Anshan, China in 2011, and Ph. D. degree in System Engineering from Northeastern University, Shenyang, China in 2019. He joined Shangqiu Normal University, Shangqiu, China in 2019, and is now a lecturer of logistics management. His current research interests include large-scaled integer programming, cutting stock problem, vehicle routing problem, traveling repairman problem and intelligent optimization algorithm.

Shaoyu Zhang is currently a high school student at Firm Foundation Christian School in the United States. Having spent the first fifteen years of life studying in China, Shaoyu developed a strong academic foundation before moving abroad in 2023 to further pursue interdisciplinary studies. She will be participating in the Running Start Program at Lower Columbia College, where she aims to explore advanced coursework in STEM fields.Her academic interests include applied mathematics, data analysis, and intelligent optimization algorithms, with a growing enthusiasm for their applications in healthcare and technological innovation. She aspires to integrate technological advancements into practical fields such as nursing and healthcare management, aiming to contribute to innovative solutions in patient care and medical logistics.