Management System Engineering

RESEARCH ARTICLE

Open Access

Enhanced multi-task deep reinforcement learning for the integrated inventory-routing problem under VMI mode

Gang Lu[1], Junmin Wan[2*], Lijing Du[3] and Xiaofang Chen[3]

Abstract

Effective inventory replenishment and routing are crucial for minimizing supply chain costs and enhancing operational efficiency. In this paper, we focus on the integrated optimization of inventory replenishment and routing problems in Vendor Managed Inventory (VMI) mode and further propose an enhanced Multi-Task Proximal Policy Optimization (MTPPO) with deep reinforcement learning. The proposed model refines inventory replenishment strategies by learning from inventory status and retailer location data. Routing strategies are optimized by utilizing a Graph Isomorphism Network (GIN) to analyze the network data of retailers and formulate routing strategies based on delivery requirements and retailer network information. By jointly optimizing inventory and routing problems, the total cost is reduced. Further, experimental results demonstrate that the MTPPO outperforms heuristic algorithms, reducing inventory costs by 8.58% and total costs by 6.18%.

Keywords Inventory routing problem, Inventory replenishment, Vendor management inventory, Proximal policy optimization, Deep reinforcement learning

1 Introduction

Efficient response to customers' demands drives companies to manage their inventory and schedule distribution routes effectively across the whole supply chain in most industries. To reduce the lead time, companies should optimize their inventory levels to guarantee the availability of products under uncertain demand of customers. To reduce the total inventory in the supply chain, Vendor Managed Inventory (VMI) has been widely implemented as an effective strategy, allowing inventory information sharing between vendor and retailers in the distribution network (Liu et al. 2020). The vehicle routing problem (VRP) also deeply affects delivery performance, particularly when a large number of terminal warehouses exist in the distribution system. Recently, there has been significant attention on the integrated optimization of inventory and routing, rather than optimizing them separately, aiming at effectively reducing the overall supply chain costs. In this context, combining VMI with the Inventory Routing Problem (IRP) is particularly reasonable. VMI shifts the responsibility of inventory management from retailers to suppliers, enabling suppliers to make replenishment decisions based on real-time demand and stock information. Meanwhile, IRP explicitly considers vehicle routing in conjunction with replenishment planning. Integrating these two frameworks allows suppliers not only to coordinate replenishment quantities but also to optimize delivery routes simultaneously. This joint optimization ensures that inventory levels across retailers are maintained efficiently while

*Correspondence:
Junmin Wan
wan@fukuoka-u.ac.jp
[1] School of Management, Wuhan University of Technology, Wuhan 430070, China
[2] Faculty of Economics, Fukuoka University, Fukuoka 814-0180, Japan
[3] School of Management | Research Institute of Digital Governance and Management Decision Innovation, Wuhan University of Technology, Wuhan 430070, China

transportation resources are utilized more effectively, leading to improved service levels and reduced total logistics costs.

This paper addresses the IRP in VMI (IRP-VMI), aiming to optimize the delivery of multiple products to various retailers, ensuring efficient inventory replenishment. The objective is to minimize the total cost associated with inventory management and distribution. Decision making involves (1) determining the replenishment quantities for each retailer and (2) establishing the delivery sequence for the retailers.

As IRP-VMI is an integrated inventory control and VRP, which has been proven to be NP-hard, current research solving related problems primarily relies on classic techniques based on heuristic algorithms (Kumar et al. 2020; Zhang et al. 2024). However, heuristic algorithms tend to obtain sub-optimal solutions, and are often ineffective for handling complex, multi-dimensional (Boute et al. 2022), large-scale problems (Feng et al. 2024). Additionally, these algorithms exhibit limited flexibility when adapting to dynamic, uncertain, or real-time scenarios (Pan and Liu 2023). Deep reinforcement learning enables the development of models that autonomously learn patterns and strategies directly from data, obviating the need for expert-driven manual modeling (Panzer and Bender 2022). Moreover, deep reinforcement learning is adaptable to various objectives and capable of solving a broad spectrum of supply chain problems, thereby mitigating the limitations of heuristic algorithms (Yan et al. 2022). With the rapid advancement of machine learning and artificial intelligence, an increasing number of researchers are investigating deep reinforcement learning to address inventory replenishment and routing optimization (Rolf et al. 2023). However, to the best of our knowledge, the use of deep reinforcement learning for integrating inventory and routing optimization remains relatively unexplored. This gap is largely attributable to the inherent complexity of integrated optimization as a multi-objective problem, which poses significant challenges for modeling and training with deep reinforcement learning. Recent advancements in multi-objective deep reinforcement learning techniques have demonstrated potential, with several studies attempting to tackle integrated optimization using these approaches, and achieving notable results (Esteso et al. 2023).

Recent advancements in deep reinforcement learning have inspired novel approaches to tackling complex optimization challenges. However, existing research still faces limitations in algorithmic efficiency for IRP (Feng et al. 2024). To address this gap, this paper proposes an integrated optimization model for IRP-VMI using deep reinforcement learning. The study makes three key contributions:

1. To the best of our knowledge, this is the first integrated optimization approach for IRP-VMI based on deep reinforcement learning. The proposed method achieves a 100-fold improvement in computational efficiency compared to heuristic algorithms.
2. A Multi-Task Proximal Policy Optimization (MTPPO) algorithm is introduced to address the IRP-VMI problem. The algorithm can learn patterns from diverse retailer data and adapt to complex, multidimensional variables, demonstrating strong robustness under uncertainty and effectively addressing the challenges of integrated IRP-VMI optimization.
3. From a technical perspective, a Graph Isomorphism Network (GIN) (Xu et al. 2018) is employed to model retailer locations and inventory states within the VMI replenishment framework. This enhancement enables the MTPPO algorithm to simultaneously optimize inventory replenishment and routing decisions, resulting in improved overall performance in IRP-VMI.

The rest of this paper is organized as follows: Sect. 2 reviews relevant literature on inventory management and the routing problem to show the positioning of this study. Section 3 presents the integrated model focusing on inventory replenishment and routing. Section 4 proposes multi-task deep reinforcement learning methods for integrated optimization. Section 5 provides the results of quantitative experimental analyses. Section 6 draws conclusions, answers the research questions, and discusses limitations and future research directions.

2 Literature review

Although the integration of inventory and routing optimization is vital for supply chain management, previous research has largely been confined to local optimization rather than to the development of systematic optimization approaches. As a result, comprehensive models that simultaneously address inventory and routing problems across the entire supply chain remain rare. The relevant literature on models and methods for inventory management, the VRP, and integrated optimization is discussed below.

Inventory management has progressed through three stages. Initially, two well-known continuous review policies are the (s, S) and the (s, Q) policy (Van Horenbeek et al. 2013). Hollier et al. (2005) presented a modified (s, S) inventory model in which opportunistic replenishments are introduced to further minimize total replenishment costs. Kannan et al. (2013) implemented the Economic Ordering Quantity (EOQ) method for VMI inventory management, leading to potential reductions in overall supply chain inventory costs. Although (*s, S*) and EOQ replenishment strategies are easy to implement, they may result in high inventory levels. To tackle this challenge, academia has leveraged dynamic programming and heuristic mathematical models for the automated determination of replenishment model parameters. Sadeghi et al. (2013) proposed the utilization of Particle Swarm Optimization (PSO) algorithm to minimize the overall inventory cost within a multi-vendor, multi-retailer VMI supply chain. In a similar vein, Diabat (2014) introduced a hybrid algorithm that combines the Genetic Algorithm (GA) and Simulated Annealing to address the optimization of the two-echelon VMI within a single-vendor, multi-retailer context, with the ultimate goal of maximizing supply chain profitability. Kaasgariet et al. (2017) proposed the optimization of the overall cost of VMI supply chains through the use of GA and PSO algorithms. Their experiments showcased the superior performance of the PSO algorithm. Bertazzi et al. (2021) addressed the air freight shipments in a VMI system using a Mixed Integer Linear Programming (MILP) model. Importantly, these VMI management models were all based on heuristic algorithms, required manual modeling and often obtained suboptimal solutions.

The VRP plays a pivotal role in VMI. Duhamel et al. (2011) integrated the Greedy Randomized Adaptive Search (GRAS) algorithm and Evolutionary Local Search (ELS) algorithm to convert the core problem into a two-dimensional weighted items capacitated VRP. Leung et al. (2013) proposed using simulated annealing and Heuristic Local Search (SA-HLS) to resolve heterogeneous vehicle routing problems (VRPs). Sbai et al. (2022) adopted a combination of genetic algorithm and variable neighborhood search to handle VRPs. Çil et al. (2023) addressed the integrated distributed disassembly line balancing and VRP using MILP, mixed-integer nonlinear programming (MINLP), constraint programming (CP), and a multi-start simulated annealing algorithm. Given the urgency of demand, Zheng et al. (2023) employed a differential evolutionary algorithm to optimize the distribution of medical supplies, focusing on both fairness and timeliness. From the above, we find that heuristic algorithms are the most commonly used methods to solve VRPs in the current literature.

The IRP, which integrates inventory and routing optimization to reduce supply chain costs (Archetti and Ljubić 2022), is a prominent research area. Research directions encompass uncertain demand (Feng et al. 2024), multiple products (Coelho and Laporte 2013), multiple warehouses (Bertazzi et al. 2019), time window constraints (Liu and Zuo 2024) and time scope (Li et al. 2023). Due to the advantages of the VMI model in reducing inventory costs and improving distribution efficiency, some researchers have begun to investigate the IRP within the context of VMI. The methods used in IRP-VMI are shown in Table 1.

Heuristic methods are frequently used to solve the IRP-VMI, as shown in Table 1, but they have notable limitations, as discussed in Sect. 1. Deep reinforcement learning presents a promising alternative to address these issues. Researchers have successfully applied deep reinforcement learning to both inventory management (Boute et al. 2022; Liu and Zuo 2024; Mohamadi et al. 2024) and VRP (Qiu et al. 2022; Pan and Liu 2023).

Table 1 Investigated literature with IRP-VMI

References	Method type	Method
Coelho and Laporte (2013)	Mathematical	Mixed-integer linear programming
Rahimi et al. (2017)	Meta-heuristic	Non-dominated Sorting Genetic Algorithm II
Stellingwerf et al. (2018)	Mathematical	Mixed-integer programming
Bertazzi et al. (2019)	Mathematical	Mixed-integer linear programming
Dai et al. (2020)	Heuristic	Cuckoo algorithm with improved Clarke-Wright savings algorithm
Farias et al. (2021)	Mixed-heuristics	Branch-and-cut algorithm combined with a two-step metaheuristic
Raa and Aouam (2021)	Meta-heuristic	Sequential insertion and local search
Shang et al. (2022)	Mathematical	Mixed-integer linear programming
Kumari et al. (2023)	Meta-heuristic	Genetic algorithm
Feng et al. (2024)	Robust optimization	Distributionally robust optimization
Diabat et al. (2024)	Mathematical	Mixed-integer linear programming
Zhang et al. (2024)	Mathematical	Integer linear programming
Charaf et al. (2024)	Mathematical	Tabu search
Chiu et al. (2024)	Mathematical	Integer programming
Liu and Zuo (2024)	Mathematical	Colony optimization with variable neighborhood search
This paper	Learning	Deep reinforcement learning

Although prior studies have made pioneering contributions to inventory and vehicle routing optimization, a notable research gap remains in applying deep reinforcement learning to the integrated optimization of these two domains. To bridge this gap, this paper proposes the MTPPO algorithm, a deep reinforcement learning–based approach designed to jointly optimize VMI replenishment and routing decisions.

3 Problem description and model formulation

3.1 Problem description

The IRP-VMI is represented as a complete directed graph network $G = (N, A)$, as shown in Fig. 1, where N denotes the vertex set and $A = \{(i,j) : i,j \in \mathrm{V}, i \neq j\}$ is the set of available traffic links. N comprises N_0 and N_R, where N_0 represents the supplier and N_R represents the set of n retailers. For retailer i $\in N_R$, at time t, the replenishment is q_i^t, customer demand is d_i^t, initial inventory is I_i^0, and inventory capacity is k_i. The retailer's inventory I_i^t $(0 < I_i^t < k_i)$. Sales loss l_i^t occurs if $I_i^t < d_i^t$, and the sales loss equals to $\left(d_i^t - I_i^t\right)$, a smaller sales loss leads to a higher service level. The inventory holding cost for retailer i at time t is h_i^t. The supplier's initial inventory is I_s^0, there is no limitation on the supplier's inventory capacity. Once the replenishment quantity of the retailer is determined, the supplier is responsible for the delivery. There is one vehicle serving, and the loading capacity of the vehicle is Q. The vehicle departs from the supplier's warehouse and follows the route optimized by the algorithm to sequentially complete deliveries. Multiple round trips are allowed until the replenishment of all retailers are completely fulfilled.

3.2 Model formulation

The parameters and variables for the model are defined as below:

Sets and indices

N_0: The supplier;

N_R: Set of retailers, $i \in N_R = \{1, 2, \ldots n\}$;

N: All nodes, supplier and retailers $(N_0 \cup N_R)$;

A: Set of arcs of graph, each edge $(i,j) \in A$, $(i,j) \in N_R$;

Parameters:

I_s^0: Initial inventory of the supplier;

I_i^0: Initial inventory of the retailer i;

I_s^t: Inventory of the supplier s at time t;

I_i^t: Inventory of the retailer i at time t;

l_i^t: Lost sale of retail i at time t;

k_i: Inventory capacity of the retailer i;

p: Price of the product;

d_i^t : The customer demand of retailer i at time t;

α: Penalty factor for sales loss;

h: The holding cost of inventory;

Q: Loading capacity of the delivery vehicle;

c_d: Delivery cost per unit distance;

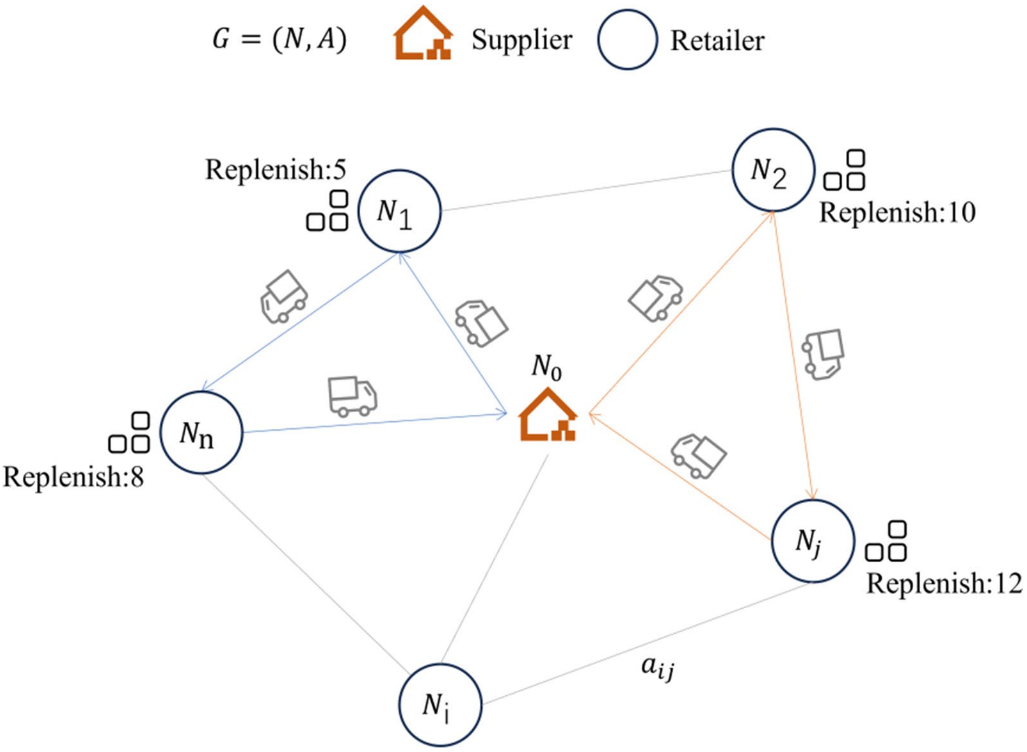

Fig. 1 Inventory Replenishment and Routing under VMI mode

$c_{i,j}$: Travel cost associated with edge *(i, j)*∈*A*;
$d_{i,j}$: Distance of edge *(i, j), (i, j)*∈*A;*
Loc_i^t: Location of the retailer i at time t.

Decision Variables:

q_i^t: Quantity of goods that the retailer i needs to replenish at time t;

$y_{i,j}^t$: Continuous flow variables representing the load capacity of the vehicle travelling from i to j (along edge *(i, j)*∈*A*) at time t;

V_i^t: 1 if the retailer i has been delivered at time t; 0, otherwise;

$X_{i,j}^t$: 1 if the vehicle travels along edge $(i,j) \in A$ at time t; 0, otherwise.

The total cost of IRP-VMI comprises both inventory management costs and distribution costs over T time periods. The inventory management costs include the retailer's inventory holding costs C_h in Eq. (1) and sales loss costs C_l in Eq. (2) in case of $(d_i^t - I_i^t) > 0$, and the distribution cost C_r in Eq. (3).

$$C_h = \sum_{t\in T}\sum_{i\in n} I_i^t * h \tag{1}$$

$$C_l = \sum\nolimits_{t\in T}\sum\nolimits_{i\in n} \max[(d_i^t - I_i^t),0]\cdot p\cdot \alpha \tag{2}$$

$$C_r = \sum_{t\in T}\sum_{(i,j)\in A} c_{i,j}X_{i,j}^t \tag{3}$$

The objectives and constraints of the integrated IRP-VMI optimization model for inventory replenishment and routing are shown as follows:

$$\min(C_h + C_l + C_r) \tag{4}$$

$$S.t.\quad I_i^t = I_i^{t-1} + q_i^t - d_i^t + l_i^t,\ i\in N^R, t\in T \tag{5}$$

$$I_i^t \geq 0,\ \ i\in N^R, t\in T \tag{6}$$

$$q_i^t \leq k_i - I_i^{t-1},\ i\in N^R, t\in T \tag{7}$$

$$q_i^t \geq 0,\ i\in N^R, t\in T \tag{8}$$

$$\sum_{i=1}^{n} q_i^t \leq I_s^t,\ i\in N^R, t\in T \tag{9}$$

$$X_{ij}^t \in \{0,1\},\ (i,j)\in N^R, t\in T \tag{10}$$

$$V_i^t \in \{0,1\},\ i\in N^R, t\in T \tag{11}$$

$$0 \leq q_i^t \leq \min\left\{Q, k_i - I_i^{t-1}\right\},\ t\in T \tag{12}$$

$$\sum_{j\in N} y_{0,j}^t = \sum_{i\in N} q_i^t,\ t\in T \tag{13}$$

$$0 \leq y_{i,j}^t \leq Q,\ (i,j)\in N^R, i\neq j, t\in T \tag{14}$$

The model's objective, as shown in Eq. (4), is to minimize the costs associated with inventory management and distribution. Equations (5) describes the inventory turnover balance for any retailer i, indicating that the current inventory level equals the previous period's inventory, adjusted for the replenishment of retailer i. Constraint (6) specifies the inventory capacity and no stock-out constraints. Equations (7) and (8) assert that replenishment and inventory levels for retailers must remain positive, with the sum of replenishment and current inventory not exceeding the retailer's storage capacity. Equation (9) states that the sum of retailer replenishments must not exceed the supplier's inventory. Equations (10) and (11) determine the visited status of retailers and routes. Constraint (12) specifies that replenishment of the retailer i must be less than vehicle capacity and inventory capacity, while Eq. (13) mandates that all retailers with replenishment are served without exception. Equation (14) asserts the current load of the vehicle is less than its maximum carrying capacity.

4 Multi-task proximal policy optimization (MTPPO)

The MTPPO framework is illustrated in Fig. 2. It incorporates two sub-tasks, inventory replenishment and routing optimization, both of which are solved using the Markov Decision Process (MDP), aiming to minimize the sum of inventory management and distribution costs. The MDP primarily consists of states, actions, and rewards, with the state at any given time t represented by s^t within the complete state space $\mathcal{S}$, $s^t \in \mathcal{S}$. Executing an action a^t triggers a transition from s^t to a subsequent state s^{t+1} and yields a reward r^t. The action a^t, drawn from the set A^t ($a^t \in A^t$), signifies all potential actions available in state s^t. The MDP's objective is to identify the optimal action a^t under the given state s^t to maximize the reward r^t. The structure of MTPPO, as illustrated in Fig. 2, integrates sub-actions for replenishment action a_1^t and routing action a_2^t. The replenishment sub-action a_1^t is determined by the output of the actor neural network, which denoting the quantity to be replenished. For the routing task, the input encompasses the current state s_2^t along with the replenishment action a_1^t and outputs the subsequent delivery node a_2^t, determined via the actor network.

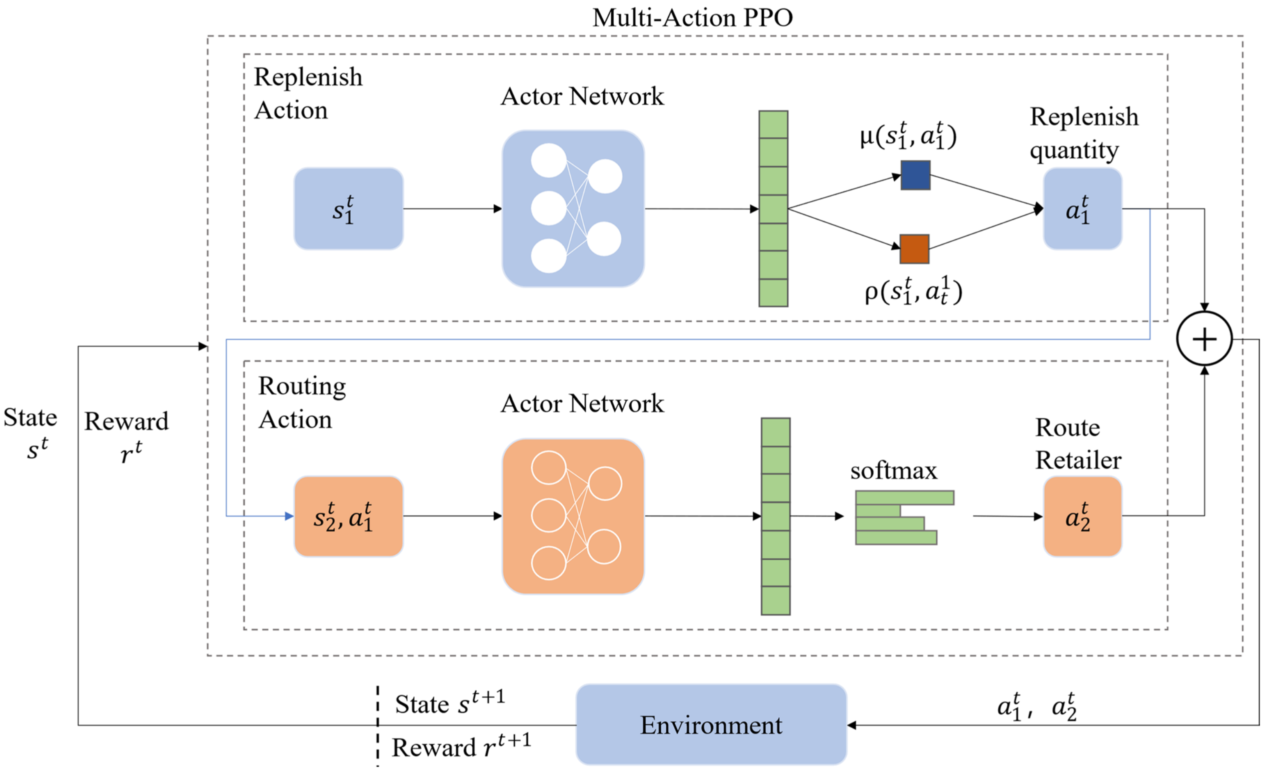

Fig. 2 Framework of the MTPPO

Upon executing the inventory replenishment action a_1^t and the routing action a_2^t, the MDP transitions to the next state and receives the corresponding reward.

The VMI inventory replenishment action and routing action consist of multiple sub-actions. As shown in Fig. 3, at time t, the inventory replenishment sub-action a_{11}^0 dictates a replenishment of 23 units for the first retailer, while sub-action a_{12}^0 dictates a replenishment of 35 units for the second retailer, and so on. Correspondingly, the routing sub-action a_{21}^0 designates retailer n_5 as the initial delivery destination, followed by sub-action a_{22}^0, which targets retailer n_2 as the subsequent delivery node,

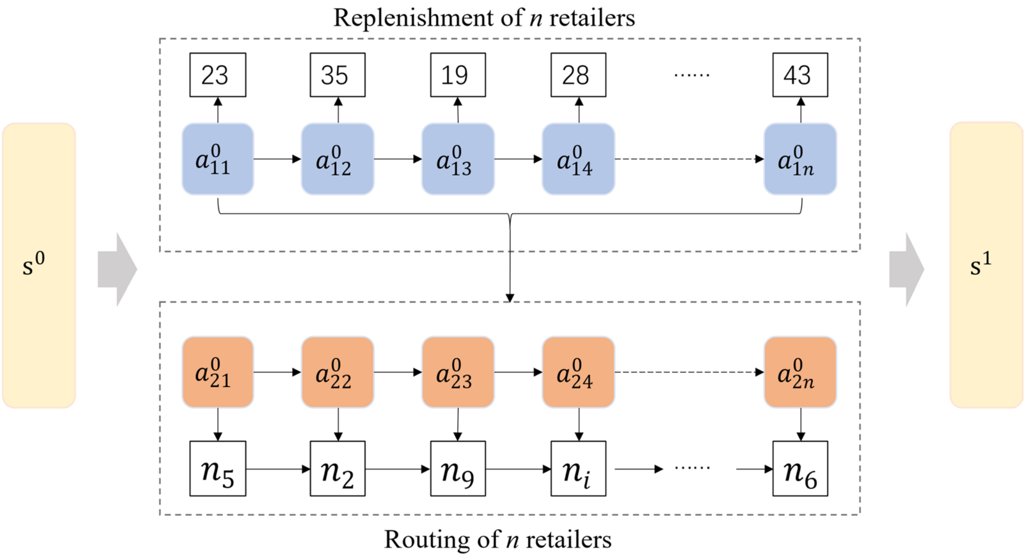

Fig. 3 Inventory replenishment and routing action decomposition

continuing in this manner until all retailers have been serviced. The following two sections will provide detailed descriptions of the inventory replenishment and routing model.

4.1 Inventory replenishment operation

The inventory replenishment strategy employs deep reinforcement learning with proximal policy optimization (PPO) to determine the optimal replenishment action based on the current inventory status. In this process, the agent obtains a corresponding reward, which is defined as the negative inventory management cost. The inventory replenishment model, including its states, actions, rewards, and policy, is described as follows.

State

The state s_i^t of retailer i includes the current inventory level, replenishment history, historical demands, as defined in Eq. (15).

$$s_i^t = \left(I_i^{t-1}, q_i^{t-T}, q_i^{t-T+1}, \ldots, q_i^{t-1}, d_i^{t-T}, d_i^{t-T+1}, \ldots, d_i^{t-1}\right) \tag{15}$$

Specifically, I_i^{t-1} represents the i-th retailer's inventory level at the time $(t-1)$, with $q_i^{t-T}, q_i^{t-T+1}, \ldots, q_i^{t-1}$ denoting the quantity replenished by the retailer at time from $(t - \mathrm{T})$ to $(t-1)$, $d_i^{t-T}, d_i^{t-T+1}, \ldots, d_i^{t-1}$ denoting the demand of the retailer during the same period.

Action

The inventory replenishment action a_i^t of retailer i represents the replenishment quantity of the retailer i at time t. The replenishment order quantity and the current inventory must remain below the retailer's storage capacity, as shown in Eq. (16).

$$a_i^t + I_i^{t-1} < k_i \tag{16}$$

Reward

The reward of retailer i at time t, r_i^t, is quantified as the inventory holding cost and sale lost taken with a negative sign. Maximizing r_i^t is equivalent to minimizing cost.

$$r_i^t = -\left(h_i^t + l_i^t\right) \tag{17}$$

The variable h_i^t (in Eq. 17) refers to the holding cost for the retailer i, while l_t^i (in Eq. 17) denotes the cost associated with sales loss incurred when inventory levels do not meet customer demand. As sales loss decreases, costs are reduced, and service level improves.

Policy model

The inventory replenishment strategy network of deep reinforcement learning is realized through the use of an attention network, as shown in the upper half of Fig. 5. This network consists of two fundamental components: an encoder and a decoder. The encoder incorporates GIN to extract features from retailer i's information, as illustrated in Fig. 4. GIN operates on a graph G defined as follows:

$$G = (V, E), V = \{r_1, \ldots r_i \ldots, r_n\}, E \subseteq V \times V \tag{18}$$

each node $r_i \in V$ has an initial feature vector $h_i^{(0)} = x_i$. At the k-th layer, the GIN updates node embeddings according to:

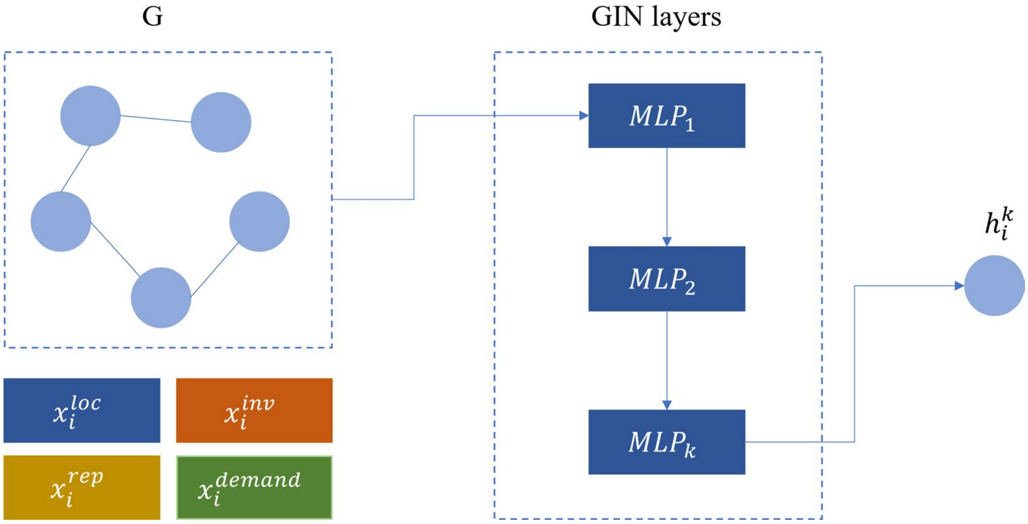

Fig. 4 GIN network

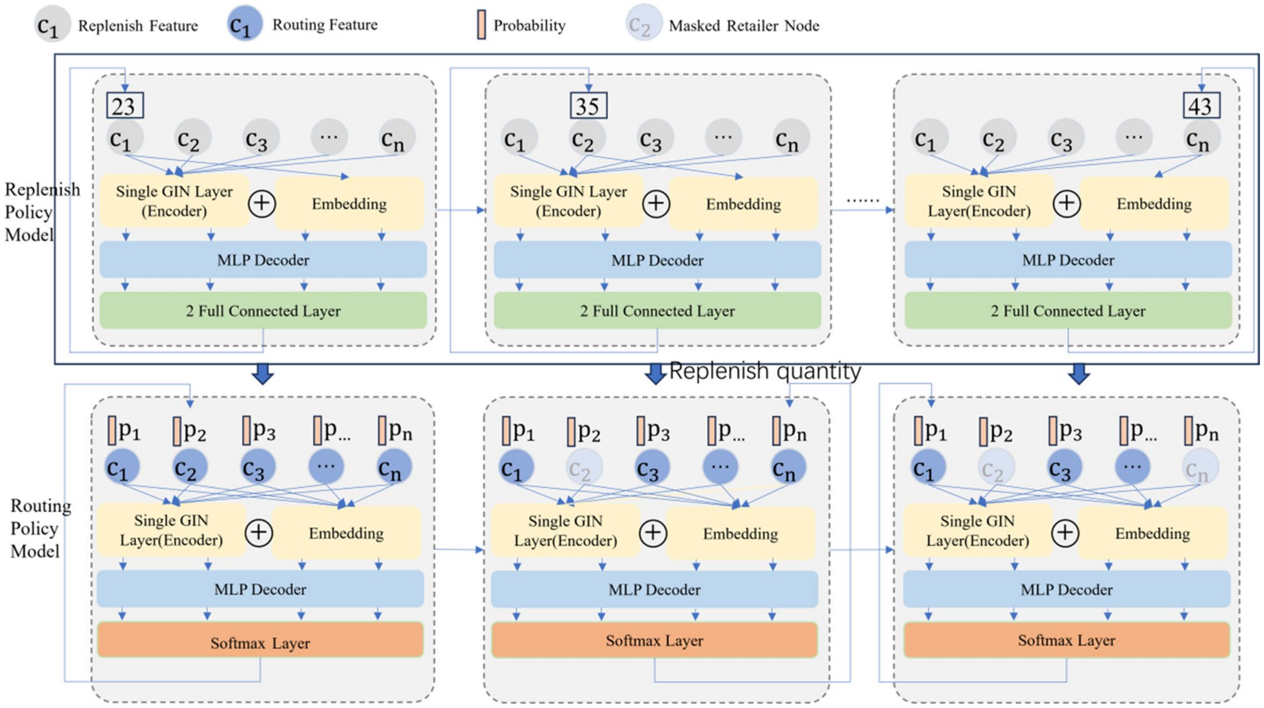

Fig. 5 Multi-Task Replenishment Strategy and Route Strategy Model

$$h_i^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+\in^{(k)}\right)\cdot h_i^{(k-1)} + \sum_{u\in\mathcal{N}(i)} h_u^{(k-1)}\right) \tag{19}$$

where $h_i^{(k)} \in \mathbb{R}^d$ represents embedding of node i at layer k, $\mathcal{N}(i)$ is the neighbor set of node i, $\in^{(k)}$ is a learnable scalar controlling the balance between self-information and neighbor information, $\mathrm{MLP}^{(k)}$ is a multilayer perceptron with fully connected layers. The initial node features are defined as:

$$h_i^{(0)} = x_i = \left[x_i^{loc}, x_i^{inv}, x_i^{rep}, x_i^{demand}\right] \tag{20}$$

which corresponds to the retailer's geographical coordinates, inventory, replenishment quantity, and demand, respectively.

As shown in Eq. (21), $encoder_i^t$ denotes the encoding of the i-th retailer's graph embedding and state s_i^t. Additionally, $embedding\left(s_i^t\right)$ represents the high-dimensional characterization of the i-th retailer's state s_i^t, while $h_i^{(k)}$(in Eq. 19) characterizes the attributes from the k-th layer of the GIN.

$$encoder_i^t = concact\left(h_i^{(k)}, embedding\left(s_i^t\right)\right) \tag{21}$$

Decoding is accomplished through the use of a Multi-Layer Perceptron (MLP), as shown in Eq. (22).

$$decoder_i^t = MLP_{\theta_d}\left(encoder_i^t\right) \tag{22}$$

The inventory replenishment policy network outputs the mean and variance of a normal distribution, as shown in the formula below:

$$\mu_i^t = F_{\theta_\mu}\left(decoder_i^t\right) \tag{23}$$

$$\rho_i^t = F_{\theta_\rho}\left(decoder_i^t\right) \tag{24}$$

$$\sigma_i^t = \exp\left(\rho_i^t\right) \tag{25}$$

$$a_i^t \sim N\left(\mu_i^t, \sigma_i^t\right) \tag{26}$$

In Eqs. (23), (24), (25), and (26), F stands for a fully connected network, which outputs a Gaussian distribution with parameters $\left(\mu_i^t, \sigma_i^t\right)$, σ_i^t is the exponent of ρ_i^t, a_i^t represents the replenishment action for retailer i, which is drawn from the normal distribution $N\left(\mu_i^t, \sigma_i^t\right)$.

4.2 Routing optimization

The routing optimization adopts a MDP framework, which is implemented using the PPO deep reinforcement learning algorithm. The framework selects the most suitable delivery nodes to add to the route by considering different types of state information, such as retailers' delivery requirements, geographical positions, vehicle load capacities, and existing routes. This process continues until the route plan satisfies the delivery needs of all retailers. The subsequent sections provide a detailed description of the MDP's state variables, actions, rewards, and model formulation.

State

The state s^t encompasses the already planned route $X_{i,j}^t$, the replenishment quantity $\left(q_1^t, q_2^t, \ldots, q_n^t\right)$ and locations of all retailers $\left(Loc_1^t, Loc_2^t, \ldots, Loc_n^t\right)$, and the present load on the delivery vehicle $y_{i,j}^t$, as shown in Eq. (27).

$$s^t = \left(X_{i,j}^t, \left(q_1^t, q_2^t, \ldots, q_n^t\right), \left(Loc_1^t, Loc_2^t, \ldots, Loc_n^t\right), y_{i,j}^t\right) \tag{27}$$

Action

The action a^t refers to the next retailer for delivery, denoted by a^t. When the cumulative load for the planned route exceeds the maximum load capacity of the vehicle, it becomes mandatory for the vehicle to return to the depot. In such instances, a^t must be chosen for the supplier warehouse.

Reward

Similar to inventory replenishment, the reward for routing is defined as the negative of the delivery cost, as shown in Eq. (28). A higher reward value corresponds to a reduced cost of delivery. The aggregate cost is computed as the sum of the delivery costs incurred for the n retailers.

$$r^t = -\sum_{(i,j)\in A} d_{i,j} * c_d \tag{28}$$

Routing model

Considering that the VRP heavily relies on the topological information of retailers, this paper employs the GIN network to generate location-based graph features. The GIN network demonstrates superior ability in representing graph information compared to other graph neural networks (GNNs) and achieves the highest performance across various graph classification benchmarks. As illustrated in the lower part of Fig. 5, the routing model employs a GIN to capture the geographic coordinates of all retailers and integrates their replenishment alongside the vehicle's current load into the embedding. These features are then aggregated and fed into an attention network, which culminates in a softmax layer that generates probabilities for selecting each retailer for delivery. A greedy algorithm is utilized to select the retailer for delivery, starting from the depot and proceeding to the subsequent retailer based on the model's output. When the vehicle's load reaches zero, it returns to the depot and repeats the process until all delivery demands are satisfied. The deep reinforcement learning network incorporates a masking technique to exclude already scheduled retailers, with the mask nodes being updated iteratively to prevent the re-servicing of retailers. As indicated in Fig. 5, the model initially calculates the delivery probabilities for all retailers. Retailer c_2, with the highest probability p_2, is selected for the initial delivery. In the subsequent step, when $V_2^t = 1$, retailer c_2 is masked, and the retailer n with the highest resulting probability, p_n, is selected for delivery. This process repeats to ensure all retailers are serviced.

The attention network consists of two distinct modules: an encoder and a decoder. The input to the network is the feature set of all retailers, which includes geographical coordinates and replenishment quantity, encoded through the GIN network. The structure of the GIN is nearly identical to that illustrated in Fig. 4, except for the definition of x_i. As shown in Eq. (29), $h_i^{(k)}(x_i)$ represents the features at the k-th layer of the GIN. Unlike the inventory replenishment model, here the initial embedding $h_i^{(0)}$ (Eq. 30) only contains the retailer's geographical coordinates and replenish quantity.

$$h_i^{(k)} = \mathrm{MLP}^{(k)}\left(\left(1+ \in^{(k)}\right) \cdot h_i^{(k-1)} + \sum_{u\in\mathcal{N}(i)} h_u^{(k-1)}\right) \tag{29}$$

$$h_i^{(0)} = x_i = \left[x_i^{loc}, x_i^{rep}\right] \tag{30}$$

As shown in Eq. (31), the encoder encompasses the aggregate information of the entire retailers' network graph.

$$encoder^t = \frac{1}{n}\sum_1^n h_i^k \tag{31}$$

The decoder operates on the encoder results and mask.

$$decoder^t = MLP_{\theta_d}\left(encoder^t, mask\right) \tag{32}$$

As shown in Eq. (32), MLP indicates a Multilayer Perceptron, mask pertains to the excluded retailers. Following a softmax layer (in Eq. 33), delivery probabilities for each retailer can be described as Eq. (33).

$$\pi\left(a^t | s^t\right) = softmax\left(decoder^t\right) \tag{33}$$

4.3 Integrated optimization MTPPO

Sections 4.1 and 4.2 introduced the inventory replenishment and routing models, respectively. In this section, we present the MTPPO model, which jointly optimizes both inventory replenishment and routing. The MTPPO model employs a two-actor, one-critic structure, where the two actors are independently trained for the inventory replenishment and routing tasks. A shared critic evaluates the outputs of both actors, ensuring effective collaboration between the sub-models associated with each actor. This collaborative effort aims to maximize the total reward, thereby reducing the overall costs associated with inventory management and distribution. The architectural diagram of MTPPO is shown in Fig. 6.

MTPPO utilizes a unified critic network to assess the state value function $V(s)$ (in Eq. 34), from which the variance-reduced advantage estimator $\hat{A}^t$ is derived, where $\hat{A}^t$ is represented as (35).

$$V\left(s^t\right) = \mathbb{E}\left[G^t|s^t = s\right] \tag{34}$$

$$\begin{aligned}\hat{A}^t = &-V\left(s^t\right) + r^t + \gamma r^{t+1} + \cdots + \gamma^{T-t-1} r^{T-t} \\ &+ \gamma^{T-1} V\left(s^t\right), t \in (0, T)\end{aligned} \tag{35}$$

Here, G_t denotes the cumulative future expectation return from the state s^t, $V(s)$ representing its state value function, and r^t being the reward at time t. The parameter γ is the discount factor, typically set between 0.9 and 1. $\hat{A}^t$ is the discrepancy between the state value function and the accumulated reward from t to T, which guides the updating of the actor networks' parameters during training. The actors differ in their output types: the inventory replenishment actor generates continuous values, whereas the routing actor produces discrete actions. Nevertheless, the PPO algorithm can adapt to both continuous and discrete actor frameworks. It employs a clipped surrogate objective for the actor's network parameter optimization, as shown in Eq. (36), with ε signifying the clipping parameter and $\delta_t^k\left(\theta^k\right)$ (in Eq. 37) indicating the probability ratio between the incumbent and surrogate policies, where *inv* means inventory model, *vrp* means routing model.

$$\begin{aligned}L^{\text{CLIP}}\left(\theta^k\right) = \hat{\mathbb{E}}_t &\left[\min\left(\delta_t^k\left(\theta^k\right)\hat{A}^t\right.\right. \\ &\left.\left., \operatorname{clip}\left(\delta_t^k\left(\theta^k\right), 1-\varepsilon, 1+\varepsilon\right)\hat{A}^t\right)\right]\end{aligned} \tag{36}$$

$$\delta_t^k\left(\theta^k\right) = \frac{\pi_{\theta_k}\left(a^t\,|s^t\right)}{\pi_{\theta_k^{old}}\left(a^t\,|s^t\right)}(k \in \{inv, vrp\}) \tag{37}$$

The MTPPO algorithm's training proceeds as delineated in the pseudo-code provided.

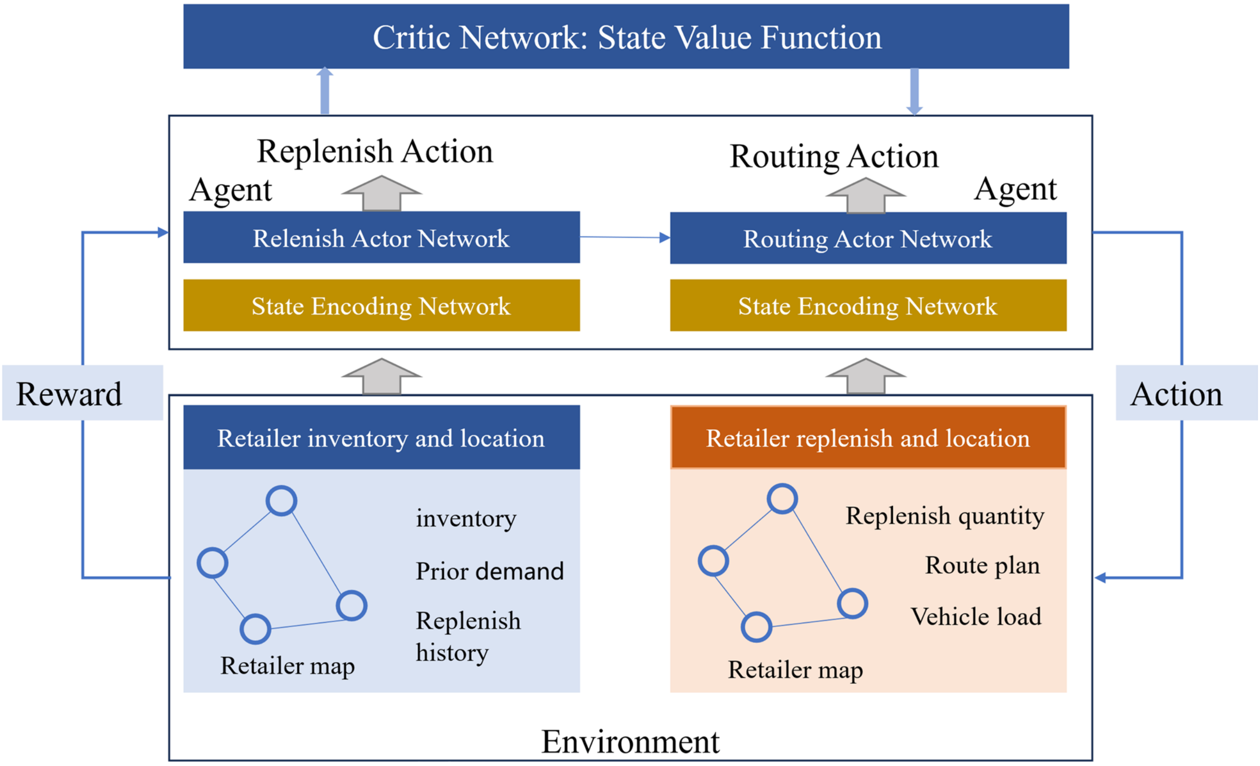

Fig. 6 The MTPPO algorithm training framework

Algorithm 1 MTPPO actor-critic style

```
for $e = 1,\cdots,\mathbf{epchos}$ do
Sampling $m$ instances from a uniform distribution;
    for $i = 1,\cdots,\boldsymbol{m}$ do
      for $t = 0,1,2,\cdots$ do
        Sample $a_{i,k}^{t}$ based on $\pi_{\theta_{k-old}}\left(a_{i,k}^{t} \mid s_{i,k}^{t}\right)\left(k \in \left(inv, vrp\right)\right)$;
        Observe reward $r_{i,k}^{t}$ and next state $s_{i,k}^{t+1}$;
        $r_{inv}^{t} = \sum_{j=1}^{n} r_{j^{inv}}^{t}\, j \in \left(1,..,n\right)\left(replenishment\, for\, n\, retailers\right)$
        $r_{vrp}^{t} = \sum_{j=1}^{h} r_{j^{vrp}}^{t}\, j \in \left(1,..,h\right)\left(routing\, for\, h\, retailers\right)$
        $\hat{A}^{t} = -V\left(s^{t}\right) + r^{t} + \gamma r^{t+1} + \cdots + \gamma^{T-t-1} r^{T-t} + \gamma^{T-1} V\left(s^{t}\right)$
        $t \in \left(0, T\right)$
        $\delta_{t}^{k}\left(\theta\right) = \frac{\pi_{\theta}^{k}(a_{k}^{t} \mid s_{l}^{t})}{\pi_{\theta_{old}}^{k}(a_{t}^{k} \mid s_{t}^{k})}\left(k \in \left(inv, vrp\right)\right)$
        if $s_{i}^{t}$ is terminal then
          break;
        End
      end for
    $L_{\mathrm{CLIP}}^{i,k}\left(\theta_{k}\right) = \hat{\mathbb{E}}_{t}\left[\mathbf{min}\left\{\delta_{i,t}^{k}\left(\theta_{\mathrm{k}}\right)\hat{A}_{i}^{t}, \mathrm{clip}\left(\delta_{i,k}^{t}\left(\theta_{k}\right), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i}^{t}\right\}\right]$
      Aggregate critic Loss:
      $L_{\mathrm{MSE}}\left(\phi\right) = \hat{\mathbb{E}}_{t}\left[\mathbf{MSE}\left(r_{k}^{t}, \hat{v}_{\phi}\left(s^{t}\right)\right)\right]; \left(\mathrm{k} \in \left\{inv, vrp\right\}\right)$;
    end for
    Update $\theta_{k}\left(\mathrm{k} \in \left\{inv, vrp\right\}\right)$ and $\phi$ by a gradient method, update $L\left(\theta_{\mathrm{k}}\right)$ and
$L_{\mathrm{MSE}}\left(\phi\right)$ by $\theta_{k}$;
    $\theta_{k}^{old} \leftarrow \theta_{\mathrm{k}}\left(k \in \left\{inv, vrp\right\}\right)$
end for
```

As depicted in Algorithm 1, the algorithm selects m instances from the training samples to fine-tune the parameters of both actor networks for inventory replenishment and routing. Lines 3 to 15 outline the actors' interactions with the environment, executes actions to gain rewards and computes the variance-reduced advantage estimator alongside the probability ratios for the new and old policies. Lines 16 to 20 involve computing the loss for the joint critic of both actors, which forms the basis for updating the parameters of the actor and critic networks individually. Subsequently, the parameters of the actor networks are replicated onto surrogate actor networks for the next iteration cycle. Once all epochs have been completed, the parameter update for both actor networks is finalized, signifying the completion of the MTPPO algorithm's training phase. This process ensures that both the inventory replenishment and routing actors are effectively optimized simultaneously, achieving an efficient strategy for managing inventory and routing within the system.

5 Results and discussion

We evaluated the performance of the inventory management and routing models independently. Following this, the performance of the integrated model was thoroughly examined. The comparative analysis includes the integrated IRP models: GA(IRP) (Vincent et al. 2022)/MTPPO, the inventory management models: GA(INV)/DDPG(Lillicrap et al. 2015)/ST-PPO-INV/MTPPO(INV), and the routing models: GA(VRP)/A3C(Babaeizadeh et al. 2016)/ST-PPO-VRP/MTPPO(VRP). Experiments were conducted across scenarios involving 20, 50, 80, and 100 retailers.

GA is a heuristic algorithm used to search for approximate optimal solutions by adjusting parameters such as population size and mutation rate. Within this context, GA(INV) denotes the model used for inventory management, GA(VRP) signifies the model applied to routing, and GA(IRP) indicates the integrated IRP model. The Deep Deterministic Policy Gradient (DDPG) is a reinforcement learning algorithm for continuous action spaces, enhancing policy gradient with an experience replay mechanism to fine-tune the Actor and Critic networks' parameters for optimal action strategies in inventory replenishment. The Asynchronous Advantage Actor-Critic (A3C) algorithm excels in managing high-dimensional state spaces and complex action scenarios, beneficial for routing. Proximal Policy Optimization (PPO) addresses challenges such as sample efficiency and stability in deep reinforcement learning, known for its ease of implementation and effectiveness in application. The ST-PPO-INV and ST-PPO-VRP models represent the division of MTPPO into single-task models for inventory management and routing, respectively. MTPPO(INV) and MTPPO(VRP) signify the inventory management and routing components of MTPPO.

The experimental computations were performed on a server configured with an Intel Xeon Silver 4310 12-Core 120W 2.1GHz processor, 256GB memory, and an NVIDIA Tesla A10 24GB GPU, operating under an Ubuntu system.

5.1 Experimental parameters

The common (Zheng et al. 2023) parameters used by all models are in Table 2. Hyper parameters for the inventory management model and the routing model are in Table 3.

5.2 Dataset and experimental metrics

The demand of each retailer is uncertain and is simulated as a random variable in the experiment, with values ranging from 0 to 50.

The geographic coordinates of retailers and suppliers were generated randomly, with the x and y-axis addresses denoted by random numbers within the 0 to 1 interval. The key metrics of the experiment are as follows:

Inv.cost: The cumulative cost of inventory management, consisting of both the holding cost and the loss from sales.

c.gap: The disparity in inventory management costs between the GA algorithm and other algorithms, utilizing GA as the reference.

Fill-rate: Order fill rate of a retailer, the percentage of orders that can be fulfilled immediately from stock without backorders or delays.

Table 2 Parameter value of the experiments

Parameter	Parameter description	Parameter value
n	Number of retailers	20,50,80,100
h_r	Holding cost per unit of product in retailer's warehouse	2
k_r	Inventory capacity of retailer	100
I_r^0	Initial inventory quantity of retailer	100
d	Retailer demand (Random)	1–50
p	Price of the product	20
α	Penalty factor for sales loss	0.3
shape	Map shape	Square
positionrange	Map coordinate range	(0,0)–(1,1)
depot	Supplier coordinates	Random
$n_{vehicle}$	Number of vehicles	1
Q	Vehicle load capacity	100
c_d	Transportation cost per unit distance	1000

Table 3 Hyper parameters of models

Model type	Algorithm	Parameter
Inventory management	DDPG	Learning rate: 0.001 Train batch size: 256 Gamma: 0.9 Replay_buffer: 5000 Mlp dimension: (64,128)
Routing	A3C	Learning rate: 0.001 Train batch size: 256 Gamma: 0.9 Mlp dimension: (64,128)
IRP (Inventory-Routing)	GA	Population size: 100 Crossover probability: 0.6 Mutation probability: 0.05 Selection pressure: 1.5 Parent and survival selection ratio: 0.5
IRP (Inventory-Routing)	MTPPO	Learning rate: 0.001 Train batch size: 256 Gamma: 0.9 GIN Layer: 3 GIN dimension: (64,128,128) MLP dimension: (128,128) Value function clip param: 0.1 The initial coefficient for KL divergence: 0.2

r.gap: The deviation in service fulfillment rates between the GA algorithm and alternative algorithms, with the GA serving as the standard.

VRP.Dist: The overall distance calculated for vehicle routing, corresponding to the direct distance between two locations.

D.Gap: The variance in total routing distances between the GA algorithm and its counterparts, with the GA's result as the benchmark.

Time: The duration of execution for the vehicle routing algorithm.

T.Gap: The divergence in execution times for the routing algorithms compared to the GA, which is set as the baseline.

VRP.Dist*1k: An estimated cost for delivery, derived by scaling the length of the delivery route by a factor of 1000.

Sum: The aggregate cost encompassing inventory management and delivery expenses.

Sum.Gap: The cost discrepancy between the GA algorithm's combined costs and that of other algorithmic combinations, with the GA's total cost as the reference point.

Table 4 Average inventory cost and service fulfillment rate for 20/50/80/100 retailers over 90 days

	Inv.cost	c.gap (%)	Fill-rate (%)	r.gap (%)	Inv.cost	c.gap (%)	Fill-rate (%)	r.gap (%)
	20_Retailers				50_Retailers			
GA(INV)	9092	–	96.7	–	23465	–	94	–
DDPG	9558	5.13	93.3	−3.4	23124	−1.45	90	−4
ST-PPO-INV	9272	1.98	93.3	−3.4	23865	1.70	92	−2
MTPPO	9186	1.03	96.7	0	23606	0.60	94	0
	80_Retailers				100_Retailers			
GA(INV)	38047	–	93.75	–	49350	–	96.00	–
DDPG	36655	−3.66	91.25	−2.50	47149	−4.46	96.00	0
ST-PPO-INV	37518	−1.39	92.50	−1.25	45806	−7.18	97.00	1
MTPPO	37080	−2.54	93.75	0	45117	−8.58	97.00	1

Note: c.Gap = relative difference in inventory cost compared with the GA(INV) baseline algorithm; r.Gap = relative difference in service fulfillment rate compared with the GA(INV) baseline algorithm. The deep reinforcement learning approach A3C and PPO are adopted in Boute et al. (2022)

Table 5 Delivery distance and algorithm execution times for different algorithms used by 20, 50, 80 and 100 retailers

	VRP.Dist	D.Gap (%)	Time	T.Gap (%)	VRP.Dist	D.Gap (%)	Time	T.Gap (%)
	20_Retailers				50_Retailers			
GA(VRP)	5.93	–	4.98	–	10.56	–	38.98	–
A3C	6.75	13.83	0.26	−94.78	12.08	14.39	0.37	−99.05
ST-PPO-VRP	6.54	10.29	0.52	−89.56	11.53	9.19	0.46	−98.82
MTPPO	6.37	7.42	0.41	−91.77	11.31	7.10	0.55	−98.59
	80_Retailers				100_Retailers			
GA(VRP)	14.57	–	82.56	–	16.91	–	113.35	–
A3C	15.53	6.59	0.43	−99.48	17.52	3.61	0.75	−99.34
ST-PPO-VRP	15.21	4.39	0.56	−99.32	17.41	2.96	0.84	−99.26
MTPPO	15.08	3.50	0.67	−99.19	17.05	0.83	0.97	−99.14

Note: D.Gap = relative difference in delivery distance compared with the GA(VRP) baseline algorithm; T.Gap = relative difference in execution time compared with the GA(VRP) baseline algorithm

5.3 Experiments results

5.3.1 Results analysis

The GA(INV) model serves as the benchmark for inventory management, and the GA (VRP) for routing. Table 4 details the 90-day average inventory costs and service satisfaction rates for scenarios with 20, 50, 80, and 100 retailers. In the 20 retailers' scenario, the GA model achieves the lowest inventory costs and the highest service satisfaction rate, peaking at 96.7%. The inventory costs associated with the Deep Deterministic Policy Gradient (DDPG), ST-PPO-INV, and MTPPO algorithms surpass those of the GA(INV). Notably, in the 100-retailer scenario, both MTPPO and ST-PPO-INV inventory costs are lower than those of the GA model. This trend illustrates the superior performance of reinforcement learning algorithms in larger-scale problems.

Table 5 presents the mean delivery distance over 90 days across scenarios with 20, 50, 80, and 100 retailers. In every testing scenario, the GA algorithm consistently achieved the shortest routing paths, with MTPPO ranking second and A3C generating the longest. In the 100 retailers' scenario, the MTPPO's delivery cost surpassed that of GA by a mere 0.83%. As the number of retailers increases, the MTPPO algorithm's performance incrementally converges with that of the GA algorithm.

Table 6 presents the aggregate costs incurred over a 90-day period, resulting from various combinations of inventory management models (GA(INV), DDPG, and ST-PPO-INV) and routing models (GA(VRP), A3C, and ST-PPO-VRP), and compares them with the integrated models MTPPO and GA(IRP). Delivery costs were approximated by multiplying the route distance by 1000. In the 20/50 retailer scenarios, the GA(IRP) incurred the lowest total cost. In the 100-retailer scenario, the MTPPO algorithm outperformed all other models, achieving a total cost that was 6.18%[1] lower than that of the GA model. As the number of retailers and the volume of data increase, the MTPPO algorithm benefits from a larger dataset for learning and training, resulting in improved performance and a consistent reduction in overall supply chain costs. Furthermore, the costs associated with MTPPO are lower than the combined costs of the individual ST-PPO-INV and ST-PPO-VRP algorithms, confirming that integrated optimization of inventory management and routing algorithms can enhance the performance of both sub-models.

5.3.2 Analysis of 20 retailers' instances

In this section, we analyze the average inventory levels and replenishment activities of 20 retailers over a 90-day period, followed by an evaluation of their routing optimization. Figure 7 illustrates that the GA(INV) algorithm exhibits significant variation in the quantity of stock replenished, with shorter intervals between restocks, resulting in the minimal total cost. The MTPPO algorithm features a slightly higher frequency of replenishment than GA(INV), with minimal variation in quantities and comparatively larger replenishment volumes, thereby accruing a higher inventory cost than the GA(INV). The DDPG algorithm and ST-PPO-INV incur more frequent replenishing with greater quantities, thus elevating the inventory costs further. Despite the MTPPO algorithm not achieving the lowest cost, it modulates its replenishment activities in accordance with demand fluctuations, ensuring that daily inventory costs are maintained within a specified threshold. The MTPPO algorithm, through model training, continually refines its inventory replenishment policy in response to environmental cues

[1] Over a 30-day period, the total cost of the MTPPO algorithm was 2.57% lower than that of the GA model.

Table 6 The combined cost of inventory management and routing algorithm combinations

Algorithms	INV.Cost	VRP.Dist*1 k	Sum	Sum.Gap (%)	INV.Cost	VRP.Dist*1 k	Sum	Sum.Gap (%)
	20_Retailers				50_Retailers			
GA(IRP)	9092	5930	15022	–	23465	10560	34025	–
GA(INV) + A3C	9092	6750	15842	5.46	23465	12080	35545	4.47
GA(INV) + ST-PPO-VRP	9092	6540	15632	4.06	23465	11530	34995	2.85
DDPG + GA(VRP)	9558	5930	15488	3.10	23124	10560	33684	−1.00
DDPG + A3C	9558	6750	16308	8.56	23124	12080	35204	3.47
DDPG + ST-PPO-VRP	9558	6540	16098	7.16	23124	11530	34654	1.85
ST-PPO-INV + GA(VRP)	9272	5930	15202	1.20	23865	10560	34425	1.18
ST-PPO-INV + A3C	9272	6750	16022	6.66	23865	12080	35945	5.64
ST-PPO-INV + ST-PPO-VRP	9272	6540	15812	5.26	23865	11530	35395	4.03
MTPPO(INV + VRP)	9186	6370	15556	3.55	23606	11310	34916	2.62
	80_Retailers				100_Retailers			
GA(IRP)	38047	14570	52617	–	49350	16910	66260	–
GA(INV) + A3C	38047	15530	53577	1.82	49350	17520	66870	0.92
GA(INV) + ST-PPO-VRP	38047	15210	53257	1.22	49350	17410	66760	0.75
DDPG + GA(VRP)	36655	14570	51225	−2.65	47149	16910	64059	−3.32
DDPG + A3C	36655	15530	52185	−0.82	47149	17520	64669	−2.40
DDPG + ST-PPO-VRP	36655	15210	51865	−1.43	47149	17410	64559	−2.57
ST-PPO-INV + GA(VRP)	37518	14570	52088	−1.01	45806	16910	62716	−5.35
ST-PPO-INV + A3C	37518	15530	53048	0.82	45806	17520	63326	−4.43
ST-PPO-INV + ST-PPO-VRP	37518	15210	52728	0.21	45806	17410	63216	−4.59
MTPPO(INV + VRP)	37080	15080	52160	−0.87	45117	17050	62167	−6.18

Note: Sum.Gap = relative difference in the combined cost (INV.Cost + VRP.Dist * 1k) compared with the GA(IRP) baseline algorithm

such as stock status and demand, thereby aligning with the dynamic nature of the environment and fostering a replenishment strategy that is more responsive to business fluctuations.

Figure 8 depicts that each of the four algorithms devised three delivery routes. Detailed routing is shown in Table 7. The GA(VRP) algorithm generated minimal aggregate distance, closely trailed by the MTPPO algorithm. The routing plans differ by only two nodes between MTPPO and GA(VRP), resulting in routes that are markedly similar. The route length calculated by the MTPPO algorithm exceeds that of GA(VRP) by 7.42%. The routes mapped out by the A3C and ST-PPO-VRP algorithms vary substantially, each extending beyond the route lengths of MTPPO. It is evident that MTPPO can consistently refine path planning strategies through reinforcement learning techniques, progressively approaching the effectiveness of heuristic algorithms.

5.3.3 Sensitivity analysis

The MTPPO algorithm leverages the transformer architecture, with a particular focus on its multi-head attention mechanism. The number of attention heads has a significant influence on the model's performance. To assess this impact, an experimental setup was conducted comparing the use of 1, 4, and 8 heads. Table 8 reveals that the model achieves peak performance with 8 heads, reflected in minimized inventory and delivery costs. A reduction in the number of heads correlates with an increase in cost. This pattern demonstrates that a higher number of heads significantly bolsters the model's capability for feature transformation and high-dimensional representation, which, in turn, enhances the learning of latent feature connections, thereby augmenting the model's performance. In the context of 100 retailers, increasing the number of heads from 1 to 4 resulted in a 4.68% reduction in total costs, while increasing the number of heads from 4 to 8 led to an additional 1.61% cost reduction. Although model performance improved with the increase in heads, the gains were marginal beyond 4 heads.

6 Conclusion

In this paper, we present an MTPPO model for the integrated optimization of inventory and routing within the VMI framework. The MTPPO model utilizes GIN to capture spatial relationships and key information among retailers, while the transformer mechanism is employed to learn latent patterns in inventory management and routing. The model's multi-task structure, featuring two

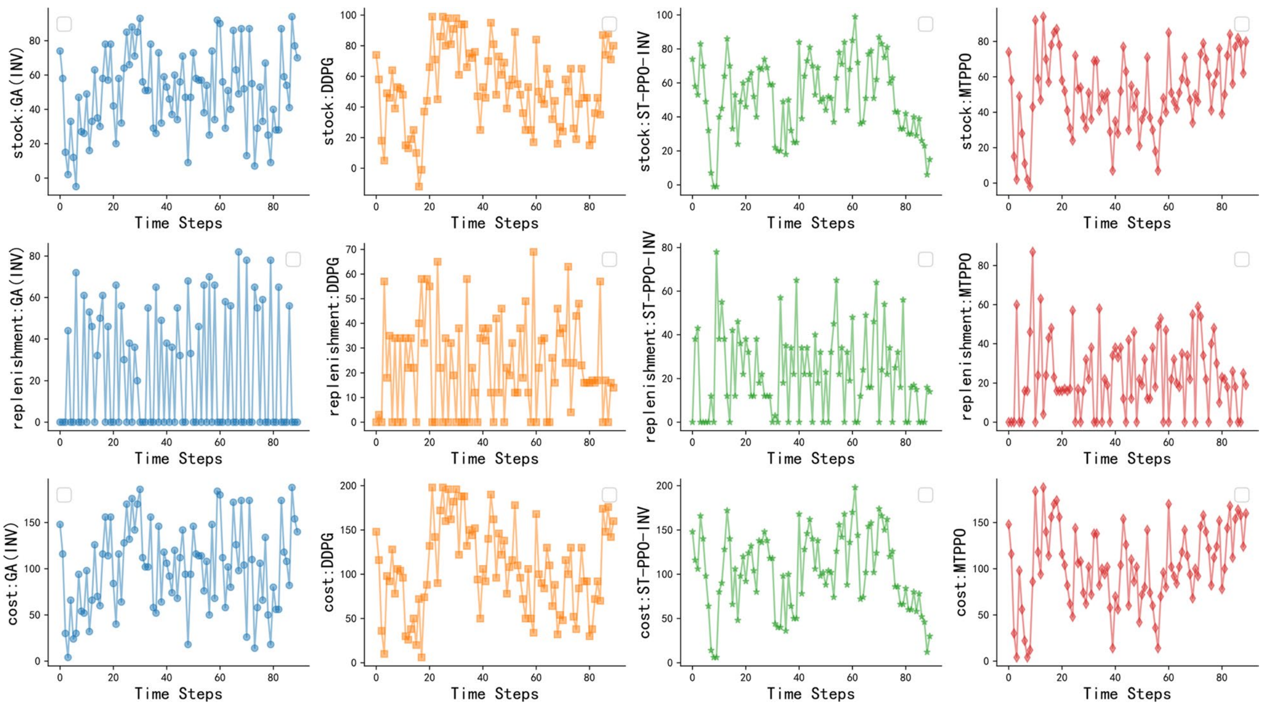

Fig. 7 The inventory levels, replenishment requirements, and inventory cost trends of retailers within 90 days (4 algorithms)

Table 7 Routing for 4 algorithms

Algorithm	Route length	Routing
ST-PPO-VRP	6.62	0->14->12->15->5->3->20->19->11->18->2 0->17->7->4->1->13->8 0->16->6->10->9->0
A3C	7.42	0->19->17->11->7->4->1->13->10 0->20->3->16->8->5->6->18->12 0->15->14->9->2->0
MTPPO	6.44	0->14->5->3->15->20->17 0->12->19->11->7->4->1->13->16 0->8->6->10->9->18->2->0
LKH3	6.14	0->14->5->3->15->20->17 0->12->19->11->7->4->1->13->16 0->18->8->6->10->9->2->0

actors and one critic, enables simultaneous optimization of inventory replenishment and routing. To the best of our knowledge, this is the first study to apply a deep reinforcement learning method to address the IRP under the VMI model (IRP-VMI). Experimental results show that the MTPPO model achieves an 8.58% reduction in inventory costs and a 6.18% reduction in total costs compared to popular heuristic algorithms for IRP.

From a practical standpoint, this study offers several managerial implications: First, integrated optimization of IRP-VMI outperforms addressing inventory replenishment and routing as separate tasks. Second, MTPPO outperforms traditional approaches such as the GA, offering the additional advantage of eliminating the need for manual model adjustments. Furthermore, the algorithm achieves up to a 100-fold improvement in efficiency. Lastly, as data volume increases and the deep learning network becomes more complex, the model's performance improves. However, the marginal improvements diminish once the model's complexity exceeds a certain threshold (four heads).

Despite the effectiveness and innovation of the MTPPO model, several areas warrant further investigation and refinement. First, the current MTPPO framework

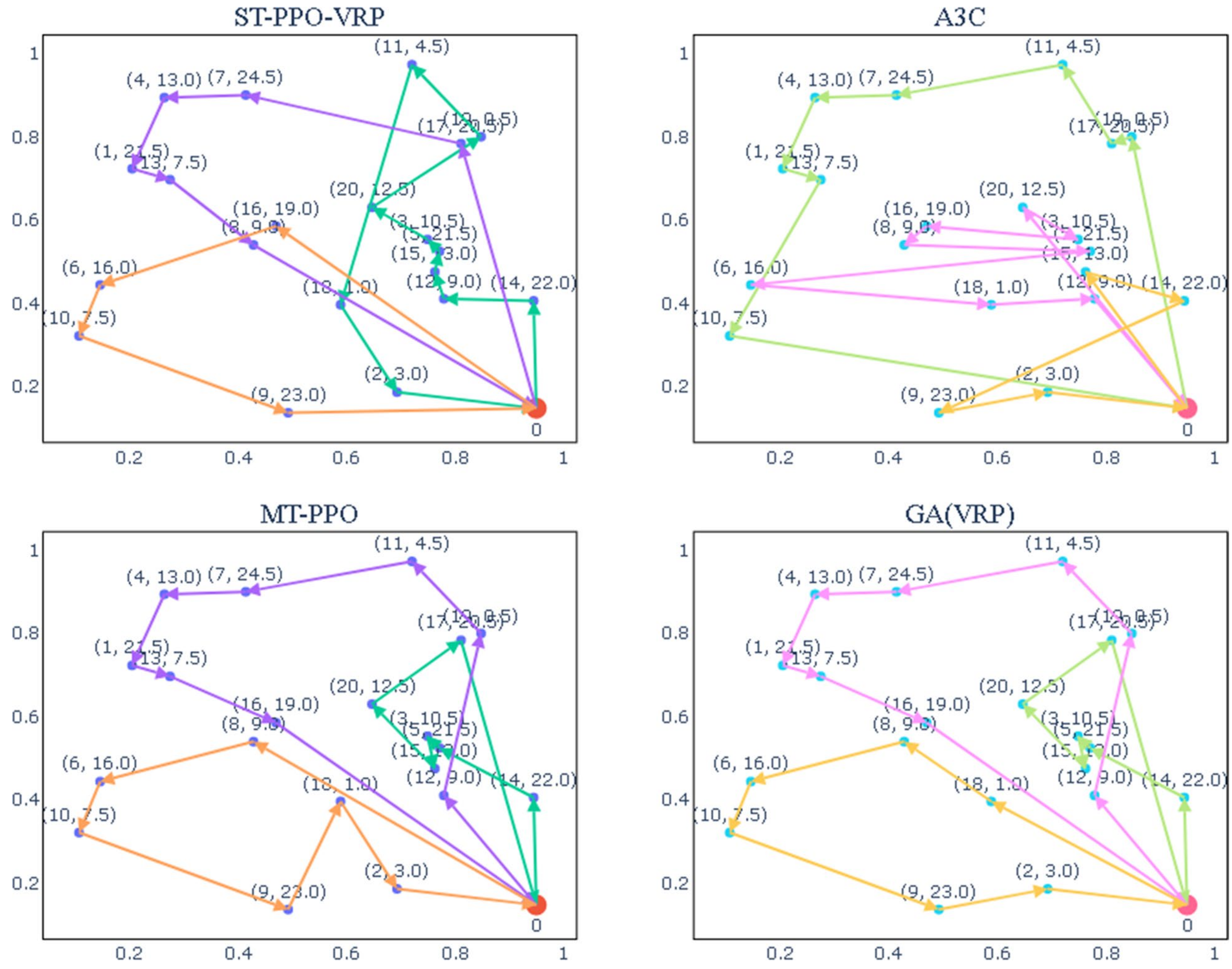

Fig. 8 Routing for 20 Retailers Using 4 Algorithms

Table 8 Inventory costs and delivery distances under different attention head counts

	Inv.cost	VRP.Dist*1 k	Sum	Inv.cost	VRP.Dist*1 k	Sum
	20_Retailers			50_Retailers		
1 Head	9753	6830	16583	25184	11840	37024
4 Heads	9504	6450	15954	24388	11570	35958
8 Heads	9186	6370	15556	23606	11310	34916
	80_Retailers			100_Retailers		
1 Head	40415	15180	55595	48836	17290	66126
4 Heads	38854	15120	53974	46036	17130	63166
8 Heads	37080	15080	52160	45117	17050	62167

Note: Sum = Inv.Cost + VRP.Dist * 1k

supports only a single supplier warehouse. As the scale of retailers expands, future research should explore the integration of multiple supplier warehouses to enhance distribution efficiency. Second, with the growth in retailer networks, reliance on a single vehicle will become inadequate. Future studies should address multi-vehicle routing problems. Third, in this study, delivery times are determined solely by suppliers. Future models will incorporate delivery time window requirements specified by retailers. Fourth, with a focus on environmental sustainability, future research will investigate routing strategies for heterogeneous fleets composed of both electric and conventional vehicles.

Author contributions

All authors contributed to the study conception and design. Material preparation, data collection and analysis were performed by Gang Lu and Lijing Du. The first draft of the manuscript was written by Gang Lu and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding

This study was funded by National Natural Science Foundation of China project "Media Attention and Contract Governance of Mixed-Ownership Enterprises: Effects, Mechanisms, and Pathways" (No. 72172113), and the Ministry of Education Humanities and Social Sciences Research Planning Fund Project "Study on the Impact of Party Organization Embedding on the Performance Improvement of Private Enterprises in the Chinese Context" (No. 21YJAZH008).

Data availability

Data will be made available on reasonable request.

Declarations

Ethics approval and consent to participate

Not applicable.

Competing interests

The author Xiaofang Chen is the member of Editorial Board for Management System Engineering but was not involved in the journal's review, or any decisions, related to this submission.

Received: 30 June 2025 Revised: 13 September 2025 Accepted: 15 September 2025
Published online: 01 October 2025

References

Archetti, C., and I. Ljubić. 2022. Comparison of formulations for the inventory routing problem. *European Journal of Operational Research* 303:997–1008. https://doi.org/10.1016/j.ejor.2021.12.051.

Babaeizadeh, M., I. Frosio, and S. Tyree et al. 2016. Reinforcement learning through asynchronous advantage actor-critic on a gpu. arXiv preprint arXiv:161106256. https://doi.org/10.48550/arXiv.1611.06256.

Bertazzi, L., L. C. Coelho, A. De Maio, and D. Laganà. 2019. A matheuristic algorithm for the multi-depot inventory routing problem. *Transportation Research Part e: Logistics and Transportation Review* 122:524–544. https://doi.org/10.1016/j.tre.2019.01.005.

Bertazzi, L., S. D. Moezi, and F. Maggioni. 2021. The value of integration of full container load, less than container load and air freight shipments in vendor–managed inventory systems. *International Journal of Production Economics* 241:108260. https://doi.org/10.1016/j.ijpe.2021.108260.

Boute, R. N., J. Gijsbrechts, W. Van Jaarsveld, and N. Vanvuchelen. 2022. Deep reinforcement learning for inventory control: A roadmap. *European Journal of Operational Research* 298:401–412. https://doi.org/10.1016/j.ejor.2021.07.016.

Charaf, S., D. Taş, S. D. P. Flapper, and T. Van Woensel. 2024. A matheuristic for the two-echelon inventory-routing problem. *Computers & Operations Research* 171:106778. https://doi.org/10.1016/j.cor.2024.106778.

Chiu, A., G. Angulo, and H. Larrain. 2024. Optimizing the long-term costs of an inventory routing problem using linear relaxation. *Transportation Research Part e: Logistics and Transportation Review* 183:103447. https://doi.org/10.1016/j.tre.2024.103447.

Çil, Z. A., H. Öztop, Z. D. Kenger, and D. Kizilay. 2023. Integrating distributed disassembly line balancing and vehicle routing problem in supply chain: Integer programming, constraint programming, and heuristic algorithms. *International Journal of Production Economics* 265:109014. https://doi.org/10.1016/j.ijpe.2023.109014.

Coelho, L. C., and G. Laporte. 2013. A branch-and-cut algorithm for the multi-product multi-vehicle inventory-routing problem. *International Journal of Production Research* 51:7156–7169. https://doi.org/10.1080/00207543.2012.757668.

Dai, Z., K. Gao, and B. C. Giri. 2020. A hybrid heuristic algorithm for cyclic inventory-routing problem with perishable products in VMI supply chain. *Expert Systems with Applications* 153:113322. https://doi.org/10.1016/j.eswa.2020.113322.

Diabat, A. 2014. Hybrid algorithm for a vendor managed inventory system in a two-echelon supply chain. *European Journal of Operational Research* 238:114–121. https://doi.org/10.1016/j.ejor.2014.02.061.

Diabat, A., N. Bianchessi, and C. Archetti. 2024. On the zero-inventory-ordering policy in the inventory routing problem. *European Journal of Operational Research* 312:1024–1038. https://doi.org/10.1016/j.ejor.2023.07.013.

Duhamel, C., P. Lacomme, A. Quilliot, and H. Toussaint. 2011. A multi-start evolutionary local search for the two-dimensional loading capacitated vehicle routing problem. *Computers & Operations Research* 38:617–640. https://doi.org/10.1016/j.cor.2010.08.017.

Esteso, A., D. Peidro, J. Mula, and M. Díaz-Madroñero. 2023. Reinforcement learning applied to production planning and control. *International Journal of Production Research* 61:5772–5789. https://doi.org/10.1080/00207543.2022.2104180.

Farias, K., K. Hadj-Hamou, and C. Yugma. 2021. Model and exact solution for a two-echelon inventory routing problem. *International Journal of Production Research* 59:3109–3132. https://doi.org/10.1080/00207543.2020.1746428.

Feng, Y., A. Che, and N. Tian. 2024. Robust inventory routing problem under uncertain demand and risk-averse criterion. *Omega (Westport)* 127:103082. https://doi.org/10.1016/j.omega.2024.103082.

Hollier, R. H., K. L. Mak, and K. F. C. Yiu. 2005. Optimal inventory control of lumpy demand items using (s, S) policies with a maximum issue quantity restriction and opportunistic replenishments. *International Journal of Production Research* 43:4929–4944. https://doi.org/10.1080/00207540500218967.

Kaasgari, M. A., D. M. Imani, and M. Mahmoodjanloo. 2017. Optimizing a vendor managed inventory (VMI) supply chain for perishable products by considering discount: Two calibrated meta-heuristic algorithms. *Computers & Industrial Engineering* 103:227–241. https://doi.org/10.1016/j.cie.2016.11.013.

Kannan, G., M. C. Grigore, K. Devika, and A. Senthilkumar. 2013. An analysis of the general benefits of a centralised VMI system based on the EOQ model. *International Journal of Production Research* 51:172–188. https://doi.org/10.1080/00207543.2011.653838.

Kumar, A., S. K. Mangla, P. Kumar, and S. Karamperidis. 2020. Challenges in perishable food supply chains for sustainability management: A developing economy perspective. *Business Strategy and the Environment* 29:1809–1831. https://doi.org/10.1002/bse.2470.

Kumari, M., P. K. De, P. Narang, and N. H. Shah. 2023. Integrated optimization of inventory, replenishment, and vehicle routing for a sustainable supply chain utilizing a novel hybrid algorithm with carbon emission regulation. *Expert Systems with Applications* 220:119667. https://doi.org/10.1016/j.eswa.2023.119667.

Leung, S. C. H., Z. Zhang, D. Zhang, et al. 2013. A meta-heuristic algorithm for heterogeneous fleet vehicle routing problems with two-dimensional loading constraints. *European Journal of Operational Research* 225:199–210. https://doi.org/10.1016/j.ejor.2012.09.023.

Li, R., Z. Cui, Y.-H. Kuo, and L. Zhang. 2023. Scenario-based distributionally robust optimization for the stochastic inventory routing problem. *Transportation Research Part e: Logistics and Transportation Review* 176:103193. https://doi.org/10.1016/j.tre.2023.103193.

Lillicrap, T. P., J. J. Hunt, and A. Pritzel et al. 2015. Continuous control with deep reinforcement learning. arXiv preprint arXiv:150902971. https://doi.org/10.48550/arXiv.1509.02971.

Liu, Z., and X. Zuo. 2024. Inventory routing problem with split delivery and variable time windows for customers with small capacity and large sales. *IEEE Transactions on Intelligent Transportation Systems* 25:10375–10388. https://doi.org/10.1109/TITS.2024.3357648.

Liu, W., G. Y. Ke, J. Chen, and L. Zhang. 2020. Scheduling the distribution of blood products: A vendor-managed inventory routing approach. *Transportation Research Part e: Logistics and Transportation Review* 140:101964. https://doi.org/10.1016/j.tre.2020.101964.

Mohamadi, N., S. T. A. Niaki, M. Taher, and A. Shavandi. 2024. An application of deep reinforcement learning and vendor-managed inventory in perishable supply chain management. *Engineering Applications of Artificial Intelligence* 127:107403. https://doi.org/10.1016/j.engappai.2023.107403.

Pan, W., and S. Q. Liu. 2023. Deep reinforcement learning for the dynamic and uncertain vehicle routing problem. *Applied Intelligence* 53:405–422. https://doi.org/10.1007/s10489-022-03456-w.

Panzer, M., and B. Bender. 2022. Deep reinforcement learning in production systems: A systematic literature review. *International Journal of Production Research* 60:4316–4341. https://doi.org/10.1080/00207543.2021.1973138.

Qiu, H., S. Wang, Y. Yin, et al. 2022. A deep reinforcement learning-based approach for the home delivery and installation routing problem. *International Journal of Production Economics* 244:108362. https://doi.org/10.1016/j.ijpe.2021.108362.

Raa, B., and T. Aouam. 2021. Multi-vehicle stochastic cyclic inventory routing with guaranteed replenishments. *International Journal of Production Economics* 234:108059. https://doi.org/10.1016/j.ijpe.2021.108059.

Rahimi, M., A. Baboli, and Y. Rekik. 2017. Multi-objective inventory routing problem: A stochastic model to consider profit, service level and green criteria. *Transportation Research Part e: Logistics and Transportation Review* 101:59–83. https://doi.org/10.1016/j.tre.2017.03.001.

Rolf, B., I. Jackson, M. Müller, et al. 2023. A review on reinforcement learning algorithms and applications in supply chain management. *International Journal of Production Research* 61:7151–7179. https://doi.org/10.1080/00207543.2022.2140221.

Sadeghi, J., S. M. Mousavi, S. T. A. Niaki, and S. Sadeghi. 2013. Optimizing a multi-vendor multi-retailer vendor managed inventory problem: Two tuned meta-heuristic algorithms. *Knowledge-Based Systems* 50:159–170. https://doi.org/10.1016/j.knosys.2013.06.006.

Sbai, I., S. Krichen, and O. Limam. 2022. Two meta-heuristics for solving the capacitated vehicle routing problem: The case of the Tunisian Post Office. *Operational Research*. https://doi.org/10.1007/s12351-019-00543-8.

Shang, X., G. Zhang, B. Jia, and M. Almanaseer. 2022. The healthcare supply location-inventory-routing problem: A robust approach. *Transportation Research Part e: Logistics and Transportation Review* 158:102588. https://doi.org/10.1016/j.tre.2021.102588.

Stellingwerf, H. M., G. Laporte, F. C. A. M. Cruijssen, et al. 2018. Quantifying the environmental and economic benefits of cooperation: A case study in temperature-controlled food logistics. *Transportation Research Part d: Transport and Environment* 65:178–193. https://doi.org/10.1016/j.trd.2018.08.010.

Van Horenbeek, A., J. Buré, D. Cattrysse, et al. 2013. Joint maintenance and inventory optimization systems: A review. *International Journal of Production Economics* 143:499–508. https://doi.org/10.1016/j.ijpe.2012.04.001.

Vincent, F. Y., N. Y. Salsabila, N. Siswanto, and P.-H. Kuo. 2022. A two-stage genetic algorithm for joint coordination of spare parts inventory and planned maintenance under uncertain failures. *Applied Soft Computing* 130:109705. https://doi.org/10.1016/j.asoc.2022.109705.

Xu, K., W. Hu, J. Leskovec, S. Jegelka. 2018. How powerful are graph neural networks? arXiv preprint arXiv:181000826. https://doi.org/10.48550/arXiv.1810.00826.

Yan, Y., A. H. F. Chow, C. P. Ho, et al. 2022. Reinforcement learning for logistics and supply chain management: Methodologies, state of the art, and future opportunities. *Transportation Research Part e: Logistics and Transportation Review* 162:102712. https://doi.org/10.1016/j.tre.2022.102712.

Zhang, Y., F. Chu, A. Che, and Y. Li. 2024. Closed-loop inventory routing problem for perishable food with returnable transport items selection. *International Journal of Production Research* 62:501–521. https://doi.org/10.1080/00207543.2023.2275639.

Zheng, F., L. Du, X. Li, et al. 2023. Multi-objective medical supplies distribution open vehicle routing problem with fairness and timeliness under major public health emergencies. *Management System Engineering* 2:5. https://doi.org/10.1007/s44176-023-00015-6.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.