Front. Eng. Manag. 2025, 12(4): 721–735
https://doi.org/10.1007/s42524-025-4170-7

RESEARCH ARTICLE

Yong ZHANG, Zhipeng YUAN, Jia DING, Feng GUO, Junyang JIN

Intelligent and efficient fiber allocation strategy based on the dueling-double-deep Q-network

© Higher Education Press 2025

Abstract Fiber allocation in optical cable production is critical for optimizing production efficiency, product quality, and inventory management. However, factors like fiber length and storage time complicate this process, making heuristic optimization algorithms inadequate. To tackle these challenges, this paper proposes a new framework: the dueling-double-deep Q-network with twin state-value and action-advantage functions (D3QNTF). First, dual action-advantage and state-value functions are used to prevent overestimation of action values. Second, a method for random initialization of feasible solutions improves sample quality early in the optimization. Finally, a strict penalty for errors is added to the reward mechanism, making the agent more sensitive to and better at avoiding illegal actions, which reduces decision errors. Experimental results show that the proposed method outperforms state-of-the-art algorithms, including greedy algorithms, genetic algorithms, deep Q-networks, double deep Q-networks, and standard dueling-double-deep Q-networks. The findings highlight the potential of the D3QNTF framework for fiber allocation in optical cable production.

Keywords optical fiber allocation, deep reinforcement learning, dueling-double-deep Q-network, dual action-advantage and state-value functions, feasible solutions

Received Jul. 21, 2024; revised Sep. 29, 2024; accepted Oct. 11, 2024

Yong ZHANG, Zhipeng YUAN, Feng GUO
School of Information Science and Engineering, Wuhan University of Science and Technology, Wuhan 430081, China

Jia DING (✉)
School of Economics and Management, East China Normal University, Shanghai 200062, China
E-mail: jding@fem.ecnu.edu.cn

Junyang JIN
HUST Wuxi Research Institute, Wuxi 214174, China; Yuanshi Autonomy Co. Ltd, Nantong 226010, China

This work was supported by the National Natural Science Foundation of China (Grant Nos. 52205519 and 62273264).

1 Introduction

Optical cables, as a fundamental component of communication networks, are essential for telephone communication, internet access, and data transmission across data centers (Liu et al., 2023). The allocation of optical fibers, a critical step in deploying these systems, faces challenges, including the labor-intensive and time-consuming nature of manual selection and suboptimal use of cable resources (Tan et al., 2024). Multi-objective optimization algorithms offer a key technological solution for efficient fiber allocation within optical cables.

Multi-objective optimization algorithms are broadly classified into three categories: exact algorithms, heuristic algorithms, and artificial intelligence algorithms. Common exact algorithms include branch and bound, decomposition methods, and Lagrangian relaxation algorithms (Silva et al., 2018). However, many multi-objective optimization problems (MOPs) exhibit nonlinearity, discontinuity, and non-differentiability, making exact algorithms difficult to apply effectively in practice (Zhong et al., 2024). As the number of optimization variables increases, solving these problems becomes more complex, time-consuming, and resource-intensive, limiting their use in large-scale, real-world applications.

To address MOPs, approximation methods that do not aim for exact solutions have become widely adopted. These include simulated annealing (Lee and Perkins, 2021), tabu search (Wang et al., 2021a), genetic algorithms (GA) (Xue et al., 2020), particle swarm optimization (Fang et al., 2024), and ant colony optimization (Martin et al., 2020). Significant advancements have been made to these intelligent optimization techniques. For example, Li et al. (2023) introduce a decomposition-based switching multi-objective whale optimizer, Ma et al. (2023) present a particle swarm optimization-assisted deep domain adaptation method, Li et al. (2024) propose a dynamic multi-objective optimization algorithm based on a hierarchical response system, and Wang et al. (2023b) develop an intelligent scheduling application integrated with MOPs

to tackle truck scheduling problems. While these methods generally yield satisfactory solutions in reasonable time frames, they do not guarantee global optimality and their performance is highly sensitive to parameter settings, with different configurations potentially producing varied results.

Reinforcement learning, an artificial intelligence approach, adjusts strategies through continuous interaction with the environment to maximize long-term rewards. It is widely used in areas such as robotics control and optimization. Compared to heuristic algorithms, reinforcement learning offers several advantages: (1) it can learn and adapt strategies in real-time through environmental interactions, enabling effective responses to changes and uncertainties (Gui et al., 2023); (2) it is well-suited for high-dimensional and complex decision-making problems, effectively managing the increasing size of state and action spaces (Wang et al., 2021b), and (3) policy improvement is automated, reducing the need for manual parameter tuning (Zheng et al., 2023). The application of deep reinforcement learning (DRL) to MOPs has emerged as a promising approach (Mazyavkina et al., 2021). For example, an adaptive scheduling algorithm based on a deep Q network (DQN) has been developed to handle complex dynamic job-shop scheduling problems (Zhao et al., 2021). Although these methods have improved the efficiency of solving MOPs, challenges remain when applying them to optical fiber allocation. Specifically, DQN algorithms may encounter convergence issues and tend to overestimate action values, compromising strategy effectiveness.

The problem of optical fiber allocation is a typical MOP. An optimized fiber allocation plan not only minimizes fiber loss and breakage, but also improves production efficiency and product quality. In optical cable production, fiber allocation presents a complex MOP, with key factors including fiber color, length, storage time, and the number of segmented fibers. Excessive fiber segmentation can negatively affect production efficiency and inventory fiber quality. Additionally, factors such as fiber color, length, storage time, and selection sequence significantly influence inventory fiber quality. Thus, optical cable manufacturers must develop optimal fiber allocation plans to make the best use of limited inventory resources.

To address these challenges, this paper proposes an end-to-end optimization algorithm: a dueling double DQN framework with twin state-value and action-advantage functions (D3QNTF), designed to handle complex MOPs in fiber selection for optical cable production. The proposed approach automates the fiber allocation process, reducing reliance on manual operations while significantly improving production efficiency and lowering costs. The main contributions of this paper are summarized as follows:

(1) Dual action-advantage and state-value functions are used to address action value overestimation, improving the stability of the learning algorithm.

(2) A random initialization method for feasible solutions is introduced to enhance the network's learning ability, replacing the traditional approach of populating the experience pool through environmental exploration. This increases the proportion of successful samples in the early stages, significantly improving the model's decision-making capabilities.

(3) Extreme penalties for illegal actions are incorporated into the reward mechanism to increase the agent's sensitivity to and avoidance of such actions.

The structure of this paper is organized as follows. Section 2 introduces multi-objective optimization problems and reinforcement learning methods. Section 3 presents the proposed D3QNTF framework for resource allocation strategies. Section 4 provides simulation validation and a comparative discussion of the results. Finally, Section 5 offers conclusions and recommendations for future research.

2 Multi-objective optimization problems and reinforcement learning methods

2.1 Multi-objective optimization problems (MOPs)

MOPs represent a class of mathematical optimization tasks in which multiple objectives must be optimized simultaneously (Ming et al., 2023). These problems are generally formulated as follows:

$$\text{Minimize } G(\alpha) = (\varphi_1(\alpha), \varphi_2(\alpha), \ldots, \varphi_m(\alpha)), \quad (1)$$

where $\alpha = (\alpha_1, \alpha_2, ..., \alpha_n)^{\mathrm{T}}$ represents an n-dimensional candidate solution, and $G(\alpha)$ denotes an m-dimensional objective space, with $\alpha \in D$ and $D \subseteq R^n$ being an n-dimensional bounded continuous decision space. The number of objective functions is denoted by m, with $m \geqslant 2$. Since these objectives often conflict with one another, optimizing all m objectives simultaneously is challenging. To address this, the concept of Pareto dominance is applied to identify a set of Pareto-optimal solutions rather than a single optimal solution. A solution α is said to dominate a solution β (denoted as $\alpha \prec \beta$) if, for every $i \in \{1, 2, ..., m\}$, $\varphi_i(\alpha) \leqslant \varphi_i(\beta)$, and there exists at least one $j \in \{1, 2, ..., m\}$ such that $\varphi_j(\alpha) < \varphi_j(\beta)$. A solution $\alpha^* \in D$ is considered Pareto optimal if no other solution $\beta \in D$ dominates α^*. The Pareto optimal set is defined as $\Pi = \{\alpha \in D | \alpha \text{ is Paretooptimal}\}$, and the Pareto optimal front is defined as $\Lambda^* = \{G(\alpha) | \alpha \in \Pi\}$. The primary goal of MOPs is to obtain a set of Pareto optimal solutions that closely approximate Λ^* and are well-distributed along Λ^* in the objective space.

Due to their inherent complexity, MOPs are regarded as particularly challenging in practical applications. The

primary difficulty arises from the necessity to concurrently optimize multiple conflicting objectives, which complicates the effectiveness of any single optimization strategy in addressing all goals. Moreover, many real-world MOPs display characteristics such as nonlinearity, discontinuity, and non-differentiability, rendering traditional optimization methods unsuitable for direct application. Among the prevailing methodologies are both exact algorithms and heuristic algorithms, each possessing distinct limitations. While exact algorithms guarantee a global optimal solution through systematic search techniques, they tend to be computationally intensive, especially in large-scale, high-dimensional optimization scenarios. The substantial demand for computational resources and time makes these algorithms impractical in such contexts. Common exact algorithms, such as branch and bound, decomposition methods, and Lagrangian relaxation, perform adequately for small-scale problems, but encounter significant challenges as problem size escalates. Additionally, exact algorithms often necessitate problem-specific adaptations, resulting in limited generalizability and further constraining their application to complex MOPs.

In contrast, heuristic algorithms utilize flexible search strategies that enable them to identify near-optimal solutions within a restricted timeframe. Examples of this category include simulated annealing, tabu search, GAs, PSO, and ant colony optimization (Yao et al., 2023). Heuristic algorithms are particularly effective in tackling large-scale and complex problems, especially when the characteristics of the problem are unclear or when the solution space is extensive. However, they do not guarantee a global optimal solution, and their effectiveness is highly contingent upon parameter selection; different parameter combinations can result in considerable variations in results. Furthermore, heuristic algorithms often exhibit slower convergence and demonstrate less stable performance when applied to complex, constrained multi-objective problems due to their lack of robust theoretical foundations.

To address these limitations, artificial intelligence algorithms, particularly reinforcement learning, have attracted increasing attention in recent years for their application in MOPs. Reinforcement learning learns optimal strategies through interaction with the environment, demonstrating exceptional performance in dynamic and complex settings (Kiran et al., 2022). Compared to traditional exact and heuristic algorithms, reinforcement learning offers significant adaptability and is better suited for handling complex constraints. It does not depend on prior models and can progressively approach optimal solutions in scenarios characterized by high uncertainty and conflicting objectives. Furthermore, reinforcement learning continuously refines its strategy throughout the learning process based on received feedback, providing high flexibility and generalization capabilities. As a result, reinforcement learning effectively addresses the limitations of traditional methods, offering innovative solutions for complex real-world problems in MOP.

2.2 DQN and D3QN algorithm

Q-learning is a fundamental reinforcement learning algorithm, with its core principle centered on continuous interaction with the environment to learn the expected long-term rewards associated with various actions in different states (Moerland et al., 2023). Q-learning utilizes a Q-function to represent the value of state-action pairs and iteratively updates the Q-values using an update rule, enabling the agent to progressively learn the optimal policy. This update process is informed by the Bellman optimal equation, wherein the Q-table is updated using the current state, action, and the immediate reward given by the environment. The primary objective is to maximize cumulative rewards. Although Q-learning faces challenges in handling large-scale discrete state spaces, it lays the theoretical groundwork for DRL methods such as DQN.

DRL combines the perceptual capabilities of deep learning with the decision-making abilities of reinforcement learning (Wang et al., 2024). Its essence lies in the ability to learn multi-dimensional abstract features through multi-layer deep neural networks, facilitating the perception and understanding of complex situations in the environment while making decisions based on this understanding. Currently, some of the most prominent DRL methods include DQN, double DQN (Van Hasselt et al., 2016), deep deterministic policy gradient (DDPG) (Qiu et al., 2019), and deep recurrent Q-learning (Hausknecht and Stone, 2015).

The DQN architecture consists of a value network, a target value network, an error function, and a replay memory unit. It employs a deep neural network (DNN) to estimate the action-value function, while the experience replay mechanism and target value network are utilized to mitigate the instability and non-convergence issues that arise when approximating the action-value function using DNNs (Luo et al., 2021). DQN follows the same update formula as Q-learning but leverages a neural network to fit the Q-table, effectively transforming the elements of the state from discrete to continuous values. According to the Q-learning algorithm's update formula, the loss function for the DQN algorithm is defined as the mean squared error between the current Q value and the target Q value, as illustrated in the following formula:

$$L(\theta)=E\left[\left(r_{t+1}+\gamma\max_{a\in A}Q(s_{t+1},a;\theta^{-})-Q(s_t,a_t;\theta)\right)^2\right], \quad (2)$$

where θ and θ^{-} represent the weight parameters of the decision network and the target network, respectively. Once the loss function is obtained, the gradient descent method can be directly used to solve for the weight parameters θ of the neural network's loss function.

The dueling DQN algorithm serves as an enhancement over the original DQN algorithm by introducing a dueling network architecture (Wang et al., 2016). This architecture bifurcates the network into two streams: one dedicated to representing the state value function and the other to representing the action advantage function. These two streams are then combined through a specialized aggregation layer to produce the estimated state-action value function Q. By integrating these two networks, the dueling DQN significantly improves the efficiency and accuracy of the algorithm, making it well-suited for managing large state and action spaces. The final value function can be expressed as follows:

$$Q(s,a;\theta,\alpha,\beta)=V(s;\theta,\alpha)+\left(A(s,a;\theta,\beta)-\frac{1}{|A|}\sum_{a^*\in A}A(s,a^*;\theta,\beta)\right), \tag{3}$$

where α and β are the specific parameters for the state value function $V(s)$ and the action advantage function $A(s,a)$, respectively. $|A|$ denotes the size of the action space. The term $\frac{1}{|A|}\sum_{a^*\in A}A(s,a^*;\theta,\beta)$ ensures that the average value of the advantage function is zero, thereby stabilizing the computation of $Q(s,a)$.

The dueling-double-deep Q-network (D3QN) algorithm is an advanced DRL technique that integrates the features of double DQN and dueling DQN (Gök, 2024). By addressing the issue of Q-value overestimation, it produces more reliable Q estimates, thus facilitating enhanced decision-making and overall performance. The learning process of the D3QN algorithm is depicted in Fig. 1. The agent inputs the state s_t into the policy network, which calculates and outputs the Q values for each action. Then, using the ε-greedy strategy, the agent selects an action a_t to execute and interacts with the environment, obtaining a reward r_t and a new state s_{t+1} (Tokic, 2010). The tuple $\{s_t,r_t,a_t,s_{t+1}\}$ is stored in the experience replay buffer for network training. The ε-greedy strategy means that, with a probability of ε, a random action is selected, and with a probability of $1-\varepsilon$, the action corresponding to the highest Q value calculated by the current policy network is chosen, as shown in the following equation:

$$a_t=\begin{cases}\text{random action a}, & if\ r<\varepsilon,\\ \text{argmax}_{a\in A}Q(s_t,a;\theta), & \text{otherwise},\end{cases} \tag{4}$$

where $a\in A$, A is the set of possible actions. r is a random number uniformly distributed between 0 and 1, and ε is the probability of selecting a random action.

The loss function of the D3QN algorithm is calculated as follows:

$$L_{\text{D3QN}}=E\left[\left(r_t+\gamma Q^*\left(s_{t+1},\underset{a\in A}{\text{argmax}}Q(s_{t+1},a;\theta)\right);\theta^*\right)-Q(s_t,a_t;\theta)\right)^2\Big]. \tag{5}$$

In the D3QN algorithm, the selection of actions for determining the target Q value relies on the parameters θ of the policy network. Specifically, the action that corresponds to the maximum Q value for the present state within the policy network is identified, and the Q value for this action is then computed within the target network. This methodology effectively reduces the likelihood of Q-value overestimation.

3 DRL-based resource allocation strategy for optical cable production

3.1 Problem formulation

The challenge of optical fiber allocation in optical cable production can be defined as follows: In the context of production scheduling tasks (Wang et al., 2023a), there are a defined set of bundle numbers, N, and a corresponding set of lengths, L, alongside the total quantity of fibers

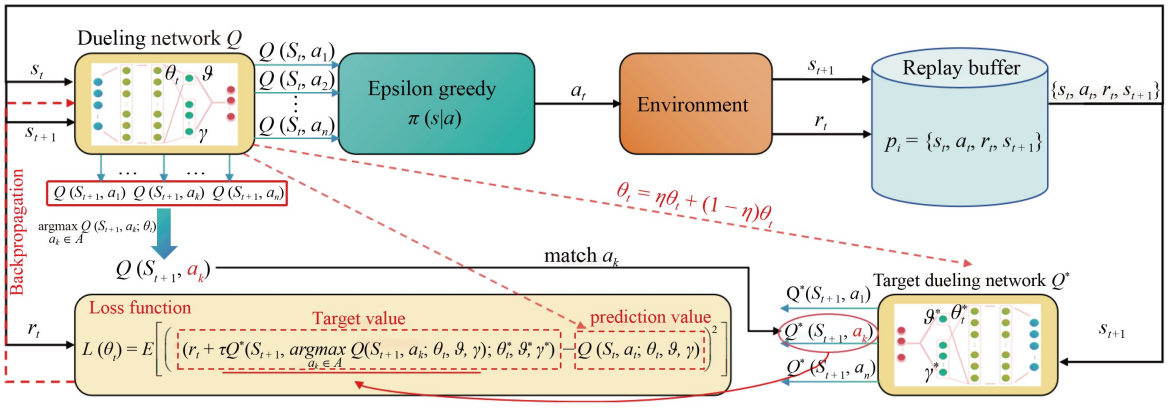

Fig. 1 The learning process of the D3QN algorithm.

required for cable manufacturing. Each product is associated with a specific length that corresponds to a particular bundle number, establishing a one-to-one relationship between the bundle and length sets. To fulfill production scheduling requirements, manufacturers are tasked with devising an efficient fiber allocation strategy that considers several factors, including the quantity and length of available fibers, the total scheduled production length, the number of tubes, and the number of cores. The total number of fibers needed for cable production is calculated by multiplying the number of tubes by the number of cores.

To address the complexities of fiber allocation under various constraints, a mathematical model has been developed (Huang et al., 2023). This model ensures that multiple selection principles are integrated into the allocation process: the length of a single optical fiber must not exceed the maximum length available in inventory; the production length limit of the tube is determined by its major, minor, and outer diameters; and priority is given to fibers that are 8 km or shorter, colored fibers, return-to-inventory fibers, and those with extended storage durations. The model's objective is to maximize the proportion of high-quality fibers in inventory while minimizing the number of segmented fibers and fiber allocation operations.

The definitions of the symbols used in the model are presented in Table 1. The mathematical model for fiber allocation constraints is outlined as follows:

$$\text{s.t.} \qquad l_i \leqslant \chi, \tag{6}$$

$$l_i \leqslant (R_b - R_s - 5) \times (R_b + R_s) \times \delta \times \pi \times \left(\frac{1}{\hbar}\right)^2 \times 0.0025 * 0.8 * 0.001, \tag{7}$$

$$L_i = \beta(l_i), \beta(l_i) = \{0, l_i \leqslant 8; 1, l_i > 8\}, \forall i \in I, \tag{8}$$

$$C_i = \alpha(c_i), \alpha(c_i) = \{0, c_i \in G; 1, c_i \in M\}, \forall i \in I, \tag{9}$$

$$U_i = J(k_i), J(k_i) = \left\{0, k_i \in Q; 1, k_i \in Q^{'}\right\}, \forall i \in I, \tag{10}$$

$$D_i = \frac{1}{1 + (t^{'} - t)}, \forall i \in I, \tag{11}$$

$$\Psi = \left(\frac{1}{|4I|}\right) \sum_{i \in I} (r_1 \times C_i + r_2 \times U_i + r_3 \times L_i + r_4 \times D_i), \forall i \in I, \tag{12}$$

Table 1 Nomenclature

Symbol	Definition
π	pi
δ	width, the distance between the side plates of the winding spool
R_b	major diameter, the diameter of the circular plates on both sides of the spool
R_s	minor diameter, the diameter of the hollow portion at the center of these plates
$\hbar$	outer diameter of the tube, the diameter of the tube's outermost layer
χ	the maximum length of stock optical fiber
I	the aggregate set of optical fiber inventory quantities, $i = 1, 2, ..., \lvert I \rvert$
G	the set of colors {B, OR, G, BR, GR, R, BL, Y, V, P, AQ}, where B = blue, OR = orange-yellow, G = green, BR = brown, GR = white, R = red, BL = black, Y = yellow, V = blue-purple, *P* = purple, and AQ = light green
M	the set of color $\{N\}$, where N = colorless
C_i	quantitative score of color dimension for the *i*th fiber
c_i	the color of the *i*th fiber
Q	the set of return-to-inventory fiber
$Q^{'}$	the set of non-return-to-inventory fiber
k_i	whether the *i*th fiber is a return-to-stock fiber
U_i	quantitative score of return-to-stock dimension for the *i*th fiber
l	the length of fiber
L_i	quantitative score of length dimension for the *i*th fiber
t	the time of fiber entry into inventory
$t^{'}$	the time of fiber usage
D_i	quantitative score of time dimension for the ith fiber
r_1, r_2, r_3, r_4	weighting coefficients for calculating the overall score of a single fiber
Ψ	the score of inventory

Equation (6) ensures that the length of a single optical fiber does not surpass the maximum available fiber length in inventory; Eq. (7) defines the relationship between the production length limit of the tube and its major, minor, and outer diameters; Eqs. (8) – (11) prioritize the use of fibers based on specific characteristics: fibers shorter than 8 km (Eq. (8)), colored fibers (Eq. (9)), return-to-inventory fibers (Eq. (10)), and fibers with prolonged storage durations (Eq. (11)). Finally, Eq. (12) ensures that these four selection principles are considered concurrently during the fiber allocation process.

To enhance understanding of the number of tubes and fibers, an illustration is provided in the form of a cross-sectional diagram of a specific optical cable product model, as depicted in Fig. 2.

In Fig. 2, small colored rings represent tubes, with the number of rings corresponding to the tube count. Within each tube, different colored dots symbolize fibers, with the number of these dots representing the fiber count. Specifically, the optical cable illustrated consists of 8 tubes, each containing 6 fibers.

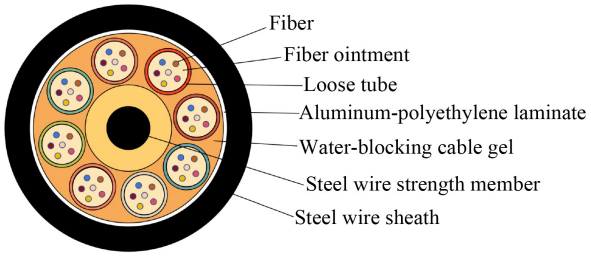

Fig. 2 Cross-sectional diagram of the optical cable.

3.2 Network design and algorithm implementation

This section presents the D3QNTF framework, which enhances the traditional D3QN methodology. For the fiber resource allocation problem in optical cable production, this algorithm aims to efficiently allocate fiber resources while minimizing the number of segmented fibers and fiber allocation operations. The proposed D3QNTF framework is illustrated in Fig. 3. The subsequent sections will outline the basic components of D3QNTF: the state S_t, the action A_t, and the reward R_t.

3.2.1 State vector

The state variable captures inventory information and the length of the optical cable to be produced. Thus, a state vector composed of the most relevant influencing factors in the environment is defined as follows:

$$S_t = \{\mu, \xi\}, \tag{13}$$

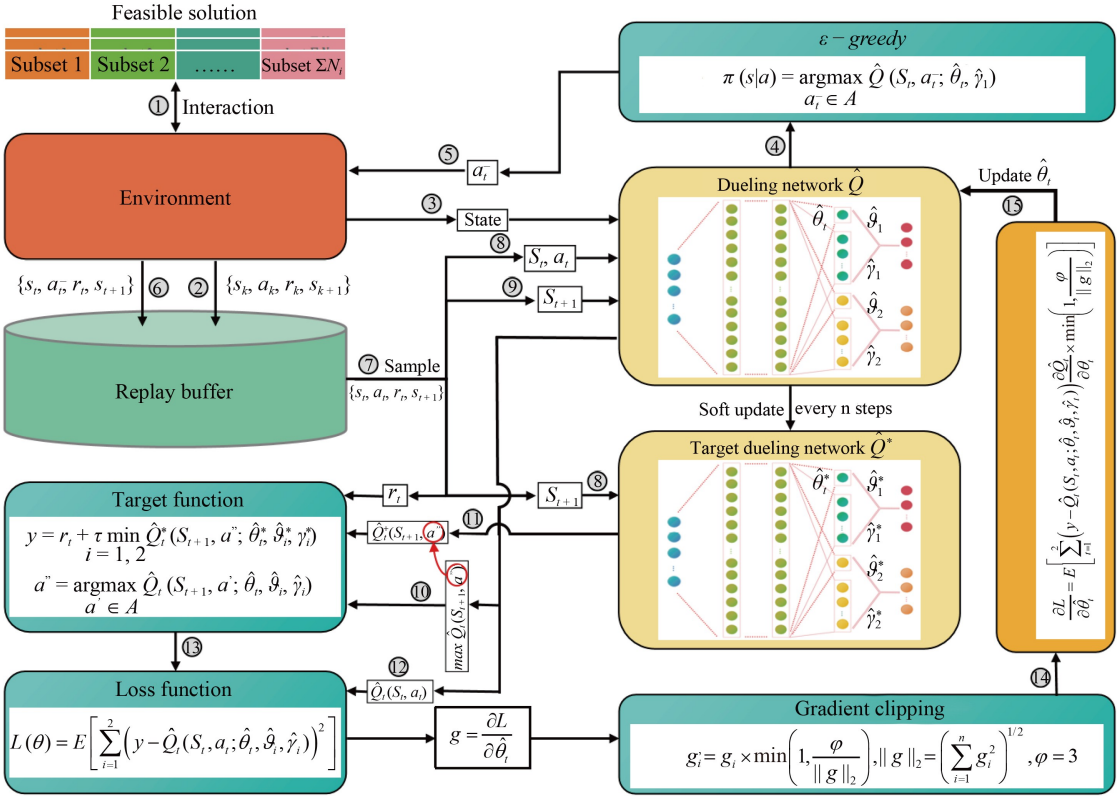

Fig. 3 Graphical illustration of the D3QNTF training process.

where μ represents the number of fiber groups meeting the length constraints, and ξ represents the remaining order demand.

3.2.2 Action vector

The goal of optical fiber allocation is to meet customer order demands while efficiently utilizing optical fibers of various lengths from inventory. Therefore, an action vector A_t is defined in the algorithm framework.

$$A_t = \{\mathbb{L}\}, \tag{14}$$

where $\mathbb{L}$ represents the length combinations that meet the length constraints.

3.2.3 Reward function

The reward function is pivotal in DRL as it steers the agent toward making appropriate decisions and avoiding detrimental actions. In this framework, the reward function aims to minimize both the number of segmented fibers and the number of fiber allocations, while simultaneously maximizing the overall inventory score. These metrics are essential for optical cable production. Consequently, the agent incurs a substantial penalty when fiber segmentation is excessive or allocation operations occur frequently. In contrast, when there is a significant improvement in the inventory score for the selected fiber lengths, the agent is rewarded.

The reward function is thus defined as the weighted sum of the number of segmented fibers, the number of fiber allocations, and the inventory score, which can be mathematically expressed as follows:

$$R_t = \ell(-\partial_1 z + \partial_2(\Psi_1 - \Psi_2) - \partial_3 \nu) - (1-\ell)\times 500, \tag{15}$$

where z represents the number of segmented fiber (i.e., the amount of fiber that needs to be cut during optical cable production), Ψ_1 denotes the inventory score after fiber allocation, Ψ_2 represents the inventory score before fiber allocation (i.e., a metric used to assess the overall quality of inventory fibers), and ν refers to the iteration number of fiber allocation (i.e., the total number of production runs based on orders). Additionally, ∂_1, ∂_2, and ∂_3 are the weight coefficients employed to compute the reward function score, while ℓ indicates the validity of the current action (1 for valid, 0 for invalid).

After defining all elements, both the evaluation network and the target network are constructed with identical architectures. Each network processes the input quadruplet $\{s_t, a_t, r_{t+1}, s_{t+1}\}$, which comprises the current state, action, reward, and next state, and outputs the predicted action value $Q(s,a)$ and the target value Y_t, respectively. Both networks utilize fully connected layers. This framework draws inspiration from the dual-network concept inherent in the twin delayed deep deterministic policy gradient (Fujimoto et al., 2018), which generates two functions: action-advantage and state-value, thereby mitigating the overestimation error of the predicted action value $Q(s,a)$. To enhance training efficiency, the common layers of the networks are merged, with the final layer employing a softmax function to convert predicted probabilities into the range [0, 1).

The detailed training process of the D3QNTF network is outlined in Algorithm 1. It is important to note that the testing process varies slightly from the training process. The specific steps of the testing process are detailed in steps 17 to 29 of Algorithm 1, with the remaining steps not being executed during the testing phase.

Algorithm 2 provides a comprehensive description of the initialization method for the replay buffer utilized by D3QNTF. Initially, several feasible solutions are generated through a random initialization method. These solutions subsequently interact with the environment, resulting in the generation of a Markov decision chain, which is stored in the replay buffer.

4 Experiments and discussion

4.1 Experimental design

4.1.1 Data description

This study utilizes real inventory data from an optical cable manufacturing company, which includes details such as fiber ID, fiber color, fiber length, a flag indicating return-to-inventory fibers, and the fiber coloring sequence. To accurately simulate real-world application scenarios and thoroughly evaluate the performance and stability of D3QNTF, the study establishes the number of tubes and fibers at 4 and 12, respectively. The outer diameter of each tube is set at 0.255 cm, with a width of 63 m, a major diameter of 100 m, a minor diameter of 50 m, and the maximum fiber length in inventory capped at 66 km.

The specific parameters are defined as follows: the fiber length selection interval is set to 1.02, meaning that fibers within the length range are standardized to the nearest minimum value within this range (i.e., the left boundary length). The redundancy length is set at 0.15, representing the additional length of the fiber relative to the tube length. The length factor is established at 1.025, utilized to estimate the relationship between the final product length and the original length. The experiment concentrates on the three most commonly used optical cable production configurations in the factory: 3*20, 3*20 + 5*20, and 3*20 + 5*20 + 7*20, where 3, 5, and 7 denote the order lengths, and 20 represents the number of bundles.

Algorithm 1 Dueling double DQN with twin state-value and action-advantage functions

Initialize dueling network $\hat{Q}$ network parameters $\hat{\theta}_t, \hat{\vartheta}_1, \hat{\gamma}_1, \hat{\vartheta}_2, \hat{\gamma}_2$;

Copy completely the dueling network $\hat{Q}$ as the target dueling network $\hat{Q}^*$; Target dueling network parameters $\hat{\vartheta}_1^*, \hat{\gamma}_1^*, \hat{\vartheta}_2^*, \hat{\gamma}_2^*$;

Initialize training step T, replay buffer P, batch K, soft update η, per G step to evaluate dueling network $\hat{Q}$, exploration step H, update time U, and descent clipping φ other parameters required for training;

The samples generated by Algorithm 2 are used to populate the replay buffer P;

$B \leftarrow 0$, $store \leftarrow [\,]$;

while True **do**

  The dueling network $\hat{Q}$ selects actions based on ε-greedy policy to interact with the environment for H steps, and the generated samples are subsequently stored in the replay buffer;

  for $i \leftarrow 1$ to U **do**

    Sample randomly batch *K* samples from the replay buffer P;

    Calculate the loss error $\mathrm{a}'' = \mathrm{argmax}_{\mathrm{a}' \in A} \widehat{\mathrm{Q}}_t(\mathrm{S}_{t+1}, \mathrm{a}'; \hat{\theta}_t, \hat{\vartheta}_1, \hat{\gamma}_1)$. $y = r_t + \tau \min_{i=1,2} \widehat{\mathrm{Q}}_t^*(\mathrm{S}_{t+1}, \mathrm{a}''; \hat{\theta}_t^*, \hat{\vartheta}_i^*, \hat{\gamma}_i^*)$;

    $\mathrm{L}(\hat{\theta}_t) = \mathbb{E}\left[\sum_{i=1}^{2} \left(\mathrm{y} - \widehat{\mathrm{Q}}_t(\mathrm{S}_t, \mathrm{a}_t; \hat{\theta}_t, \hat{\vartheta}_i, \hat{\gamma}_i)\right)^2\right]$;

    Perform gradient descent and clipping on the parameters $\hat{\theta}_t$ of the dueling Network $\hat{Q}$

$$\mathrm{g}_i' = \mathrm{g}_i \times \min\left(1, \frac{\varphi}{||g||_2}\right), \quad ||\mathrm{g}||_2 = \left(\sum_{i=1}^{\mathrm{n}} \mathrm{g}_i^2\right)^{1/2}$$

$$\hat{\theta}_{t+1} = \hat{\theta}_t + \alpha \cdot \mathbb{E}\left[\sum_{i=1}^{2} \left(\mathrm{y} - \widehat{\mathrm{Q}}_t(\mathrm{s}_t, \mathrm{a}_t; \hat{\theta}_t, \hat{\vartheta}_i, \hat{\gamma}_i)\right) \frac{\partial \widehat{\mathrm{Q}}_t(\mathrm{s}_t, \mathrm{a}_t; \hat{\theta}_t, \hat{\vartheta}_i, \hat{\gamma}_i)}{\partial \hat{\theta}_t} \cdot \min\left(1, \frac{\varphi}{||g||_2}\right)\right];$$

    Every step, soft update $\hat{\theta}_t^* = \eta \hat{\theta}_t + (1 - \eta)\hat{\theta}_t^*$;

  end for

  $E \leftarrow 0$;

  $E \leftarrow E + H$;

  if $E \geq G$ **then**

    Reset the environment;

    $returns \leftarrow 0$;

    while True **do**

      $action \leftarrow \mathrm{argmax}_{\mathrm{a} \in A} \widehat{\mathrm{Q}}_t(\mathrm{state}; \hat{\theta}_t, \hat{\vartheta}_1, \hat{\gamma}_1)$;

      Input *action* into the environment for interaction, and return *next state, reward* and *done;*

      $returns \leftarrow returns + reward$;

      if done = 1 **then**

        break;

      end if

      $state \leftarrow next\ state$;

    end while

  end if

  $B \leftarrow B + E$;

  Append *B* and *returns* to the *store*;

  if $B \geq T$ **then**

    break;

  end if

end while

return dueling network $\hat{Q}$, store;

Algorithm 2 The principle of replay buffer

Require: Number of feasible solutions Size S, Order $\sum L_i * N_i$, *redundant_length, limit_length*

Ensure: *replay_buffer*

number_combinations ← the Cartesian product of all possible combinations of the ranges defined by each integer in *N*;

Skip the first combination in *number_combinations;*

valid_$mapping$ ← [ ];

for *combination* in *number_combinations* **do**

  total_length ← sum of products of *L* and corresponding *combination* elements;

  if (*total_length* + *redundant_length*) ≤ *limit_length* **then**

    Add *combination* to *valid_mapping*;

  end if

end for

pop ← [ ];

valid_mapping_set ← set(*valid_mapping*);

for $i = 1$ **to** S **do**

  res_N ← copy of N;

  Solution ← [ ];

  while True **do**

    for *value* in *res_N* **do**

      selected_values ← random *values* from range(*value* + 1)

    end for

    if tuple(*selected_values*) in *valid_mapping_set* **then**

      Append *selected_values* to *Solution*;

      res_N ← element-wise subtraction of *selected_values* from *res_N*;

      if all elements in *res_N* are 0 **then**

        break;

      end if

    end if

  end while

  Append *Solution* to *Solutions*;

end for

Convert the *Solutions* into *actions*;

Create *replay_buffer*;

Reset the environment and assign the initial state to the variable *initial_state*;

state ← *initial_state*;

for *k* in the length of the actions *A* **do**

  Input the *state* and *action*[k] into the environment for interaction, and return *next_state, reward* and *done;*

  if done = 1 **then**

    Reset the environment and assign the initial state to the variable *state*;

  end if

  Pass *state*, *action*, *reward*, *done* into *replay_buffer*;

  state ← *next_state*;

end for

return *replay_buffer*;

4.1.2 Evaluation criteria

In line with the specific requirements of the optical cable manufacturing plant, we have established four evaluation metrics: inventory score, number of fiber allocations, number of segmented fibers, and an equally weighted sum of these three metrics. Among these, the number of fiber allocations and the number of segmented fibers serve as critical indicators in optical cable production. A reduction in these values can significantly enhance production efficiency while improving the quality of inventory fibers. Conversely, elevated values in these metrics may lead to decreased production efficiency and a decline in the quality of inventory fibers.

4.1.3 Benchmark models

To comprehensively assess the learning performance of D3QNTF in discrete action spaces, we compared it against four categories of algorithms: greedy algorithms, heuristic algorithms (including GA and PSO), multi-objective optimization algorithms (such as NSGA-II and MOPSO), and fundamental reinforcement learning algorithms (including DQN, DDQN, and D3QN). These algorithms are widely utilized across various problem-solving contexts for specific reasons: greedy algorithms offer quick local optima; heuristic algorithms are proficient in resolving complex problems by identifying near-global optima; multi-objective optimization algorithms adeptly balance multiple conflicting objectives; and foundational reinforcement learning algorithms (like DQN, DDQN, and D3QN) serve as direct performance benchmarks for D3QNTF. This comparative methodology allows for a more comprehensive evaluation of D3QNTF's decision-making performance and its applicability.

The hyperparameters utilized in the DRL experiments are outlined in Table 2. These parameters were chosen based on preliminary experimental findings and best practices from relevant literature, aiming to optimize model performance and ensure the reliability of the experimental results. Key parameters include the learning rate, batch size, and the number of iterations, all of which significantly influence the training effectiveness of the model. For the length combination case of 3*20 + 5*20, the network architecture of D3QNTF is detailed in Table 3. In comparison, D3QN omits the $V_2(s)$ and $A_2(s,a)$ layers, while DQN and DDQN do not include the $V(s)$ and $A(s,a)$ layers. Although DQN and DDQN share the same network architecture, they differ in their strategies for target value estimation.

For different order combinations 3*20, 3*20 + 5*20, and 3*20 + 5*20 + 7*20, the dimensions of the state vector S_t for DQN, DDQN, D3QN, and D3QNTF are set to 14, 64, and 165, respectively, while the dimensions of the action vector A_t are set to 13, 62, and 162, respectively.

Table 2 Hyper-parameters of DRL

Hyper-parameters	Value
Discount factor	0.98
Learning rate	1e−4
Batch size	64
Buffer size	5e4
Total learning steps	3e5
Evaluate model every multiple step	1e−3
Evaluation frequency	1.0
Soft update	5e−3
Exploration rate	0.25
Exploration length	1e−3
Number of initial solutions	50
Clip grad norm	3.0

4.2 Experimental results

D3QNTF is evaluated using the common length combination case of 3*20 + 5*20. Fig. 4 illustrates the training process of the D3QNTF algorithm for this scenario. In Fig. 4(a), it is evident that during the initial stage of training, the returns and rewards obtained by the agent from interactions with the environment exhibit considerable fluctuations, primarily due to the random initialization of the network parameters. As training progresses, these fluctuations gradually diminish and ultimately converge. In Fig. 4(b), as the number of training steps increases, the estimated Q value stabilizes, and the loss curve becomes steady, converging toward zero. This trend indicates that the model's prediction accuracy and stability improve as training advances. The minor oscillations observed during convergence can primarily be attributed to the low probability of random action selection.

To provide a clearer comparison of fiber allocation effectiveness across various algorithms, we smoothed the learning curves for each algorithm, as illustrated in Fig. 5. The results reveal that DQN exhibits the lowest return and average reward values. Although its return curve

Table 3 The network structure of D3QNTF

Layer	Input	Number of neurons	Activation function
1	64	16 × 64	ReLU
2	16 × 64	16 × 62	ReLU
$3V_1(s)$	16 × 62	1	ReLU
$3V_2(s)$	16 × 62	1	ReLU
$3A_1(s,a)$	16 × 62	62	ReLU
$3A_2(s,a)$	16 × 62	62	ReLU
4	62	62	Softmax

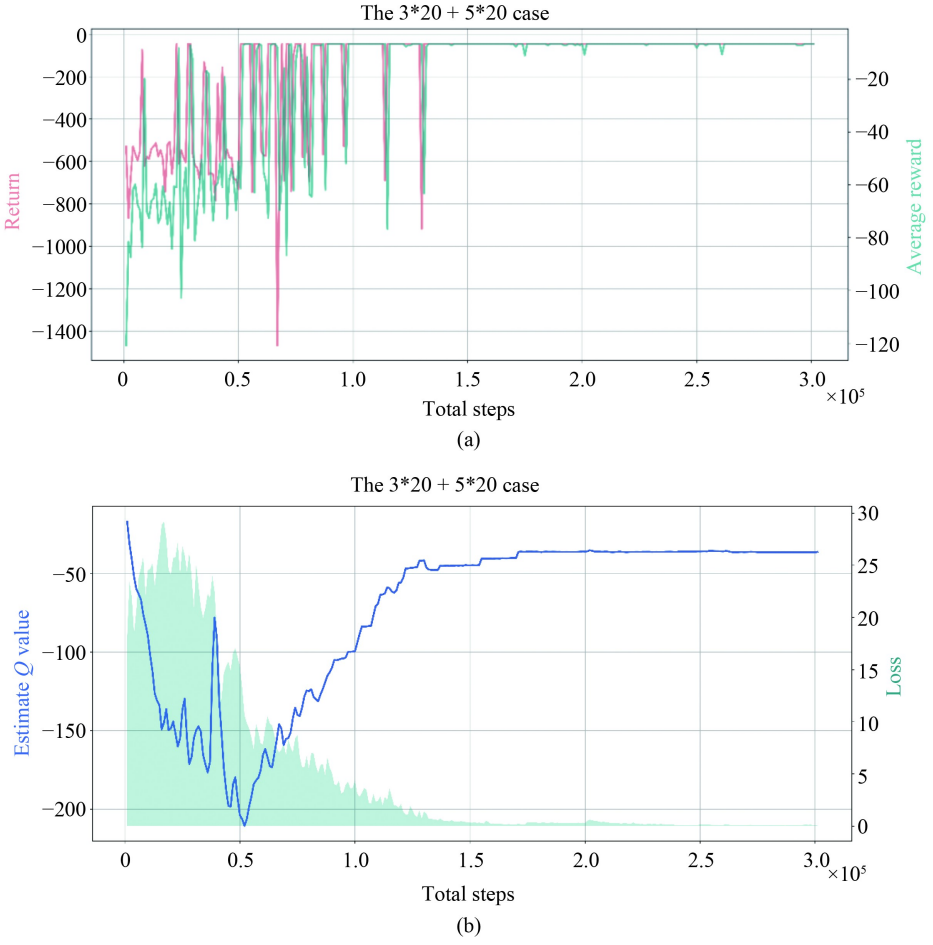

Fig. 4 Performance of D3QNTF during training. (a) Return and average reward; (b) Estimate Q value and loss.

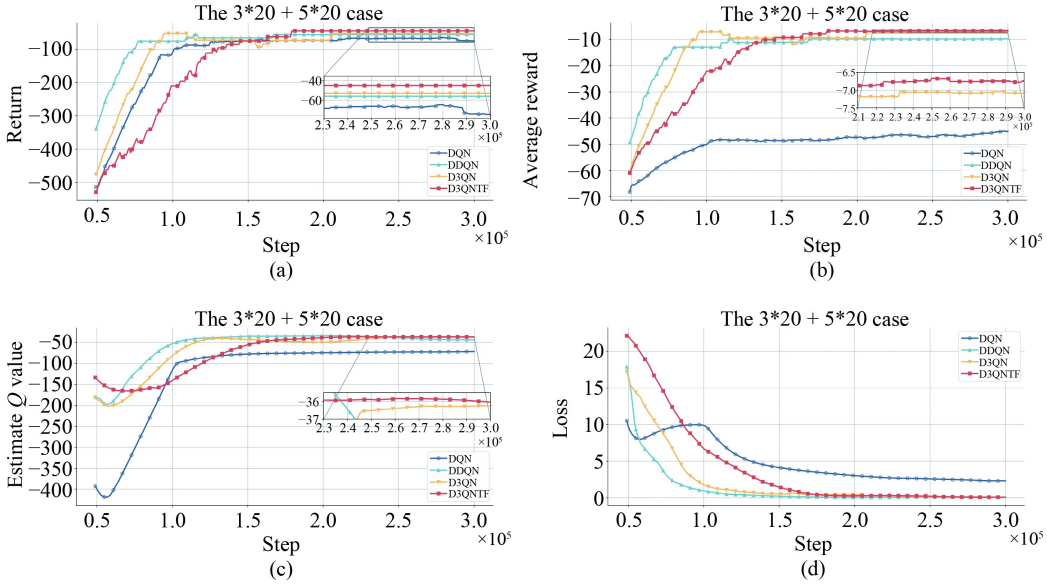

Fig. 5 Performance of different algorithms during training. (a) Return; (b) Average reward; (c) Estimate Q value; (d) Loss.

shows a tendency to stabilize, it continues to experience oscillations and even declines, primarily due to DQN's failure to achieve full convergence within a finite number of steps and its substantial Q-value estimation error. In contrast, D3QN and DDQN demonstrate notable improvements, with their loss functions converging to

near zero, reduced Q-value estimation errors, and higher rewards and returns. Notably, D3QNTF outperforms both D3QN and DDQN, benefitting from its sample initialization operation, which enhances the model's decision-making capabilities. As shown in Fig. 5(a), 5(b), and 5(d), although D3QNTF converges more slowly due to the necessity of fitting a larger number of network parameters, its return and average reward curves remain more stable throughout the learning process. This stability can be attributed to D3QNTF's implementation of dual action-advantage and state-value functions, which effectively mitigate the issue of action-value overestimation and enhance the algorithm's overall learning stability. In comparison, D3QN displays significant oscillations in its curve between 110,000 and 200,000 steps. The Q-value estimation curves for the various algorithms are presented in Fig. 5(c).

4.3 Comparison results

The experimental findings indicate that D3QNTF consistently outperforms other algorithms across different test sets. As shown in Table 4, for the 3*20 case, all algorithms successfully identify the optimal fiber allocation combination, increasing the inventory score to 0.6207 while reducing the number of segmented fibers to 0 and fiber allocations to 4. The final return of −9.9958 suggests that the greedy algorithm's search rules, alongside the heuristic and multi-objective optimization algorithms' mathematical models, are well-aligned to this scenario (Zhang et al., 2021). However, as the case expands to 3*20 + 5*20, the performance of the greedy algorithm substantially lags behind that of the other algorithms, highlighting its inefficiency in addressing large-scale solution space problems. In contrast, the GA algorithm identifies high-quality feasible solutions within the integer programming model

Table 4 Performance comparison of different algorithms

Test case	Methods	Improvement in inventory score	Number of segmented fiber	Number of fiber allocation	Return
3*20	Greedy	0.6207	000	04	−009.9958
	GA	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	PSO	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	NSGA-II	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	MOPSO	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	DQN	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	DDQN	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	D3QN	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
	D3QNTF	0.6207 ↑ 000.00%	000 ↑ 000.00%	04 ↑ 000.00%	−009.9958 ↑ 000.00%
3*20+5*20	Greedy	**0.6301**	032	12	−109.9864
	GA	0.6202 ↓ 001.57%	**000 ↑ 100.00%**	10 ↑ 016.67%	−054.9963 ↑ 050.00%
	PSO	0.6191 ↓ 001.75%	017 ↑ 046.88%	09 ↑ 025.00%	−061.9974 ↑ 043.63%
	NSGA-II	0.6197 ↓ 001.65%	020 ↑ 037.50%	**08 ↑ 033.33%**	−055.9968 ↑ 049.088%
	MOPSO	0.6214 ↓ 001.38%	020 ↑ 037.50%	**08 ↑ 033.33%**	−055.9951 ↑ 049.089%
	DQN	0.6164 ↓ 002.17%	017 ↑ 046.88%	09 ↑ 025.00%	−062.0001 ↑ 043.63%
	DDQN	0.6217 ↓ 001.33%	020 ↑ 037.50%	**08 ↑ 033.33%**	−055.9948 ↑ 049.09%
	D3QN	0.6191 ↓ 001.75%	017 ↑ 046.88%	**08 ↑ 033.33%**	−052.9974 ↑ 051.81%
	D3QNTF	0.6198 ↓ 001.63%	**000 ↑ 100.00%**	09 ↑ 025.00%	**−044.9967 ↑ 059.09%**
3*20+5*20+7*20	Greedy	0.6226	**060**	18	−230.9939
	GA	0.6170 ↓ 000.90%	124 ↓ 106.67%	13 ↑ 027.78%	−214.9995 ↑ 006.92%
	PSO	0.6197 ↓ 000.47%	127 ↓ 111.67%	13 ↑ 027.78%	−217.9968 ↑ 005.63%
	NSGA-II	0.6213 ↓ 000.21%	118 ↓ 096.67%	13 ↑ 027.78%	−208.9952 ↑ 009.523%
	MOPSO	0.6178 ↓ 000.77%	118 ↓ 096.67%	13 ↑ 027.78%	−208.9987 ↑ 009.521%
	DQN	0.6174 ↓ 000.84%	146 ↓ 143.33%	**12 ↑ 033.33%**	−223.9991 ↑ 003.03%
	DDQN	0.6166 ↓ 000.96%	129 ↓ 115.00%	**12 ↑ 033.33%**	−206.9999 ↑ 010.39%
	D3QN	0.6183 ↓ 000.69%	114 ↓ 090.00%	13 ↑ 027.78%	−204.9982 ↑ 011.25%
	D3QNTF	**0.6253 ↑ 000.43%**	076 ↓ 026.67%	15 ↑ 016.67%	**−195.9912 ↑ 015.15%**

Note: The highest value is highlighted in bold.

of fiber allocation, achieving a 100% reduction in segmented fibers and a 16.67% reduction in allocations. Due to the requirement to balance multiple metrics and GA's superior performance in one metric, the multi-objective optimization algorithms NSGA-II and MOPSO achieve slightly lower scores than GA. Nevertheless, in comparison to D3QN and D3QNTF, the solutions provided by GA remain uncompetitive, further underscoring the overall superiority of reinforcement learning algorithms for this particular problem.

In the case of 3*20 + 5*20 + 7*20, all algorithms experience a significant increase in the number of segmented fibers and allocations. The greedy algorithm, while attempting to minimize fiber segments, results in a drastic increase in the number of fiber allocations, leading to its overall poorest performance. As illustrated in Fig. 6 and Fig. 7, a visual comparison of the metrics for the 3*20 + 5*20 and 3*20 + 5*20 + 7*20 cases demonstrates that D3QNTF effectively balances the number of segmented fibers and allocations, maximizes the inventory score, and ultimately achieves the highest return. For instance, D3QNTF increases the number of fiber allocations to reduce fiber segmentation, thereby improving the final score. Furthermore, D3QNTF's architecture, which incorporates dual action-advantage and state-value functions and populates the replay buffer with randomly initialized feasible solutions, enhances its learning capabilities compared to traditional D3QN. This architecture not only improves the stability of the algorithm but also significantly enhances its optimization performance in complex environments. Overall, D3QNTF demonstrates exceptional capability in addressing large-scale and highly complex fiber allocation problems.

5 Conclusions

In optical cable production, fiber allocation faces challenges like optimizing production efficiency, improving quality, and lowering inventory costs. Traditional heuristic algorithms often struggle, especially in complex and dynamic settings. To address these issues, this paper presents an advanced D3QNTF model, which extends the DRL framework for a more adaptive solution. The D3QNTF model incorporates dual action-advantage and state-value functions and introduces a novel random initialization method. This replaces the traditional method of populating the experience pool through environmental

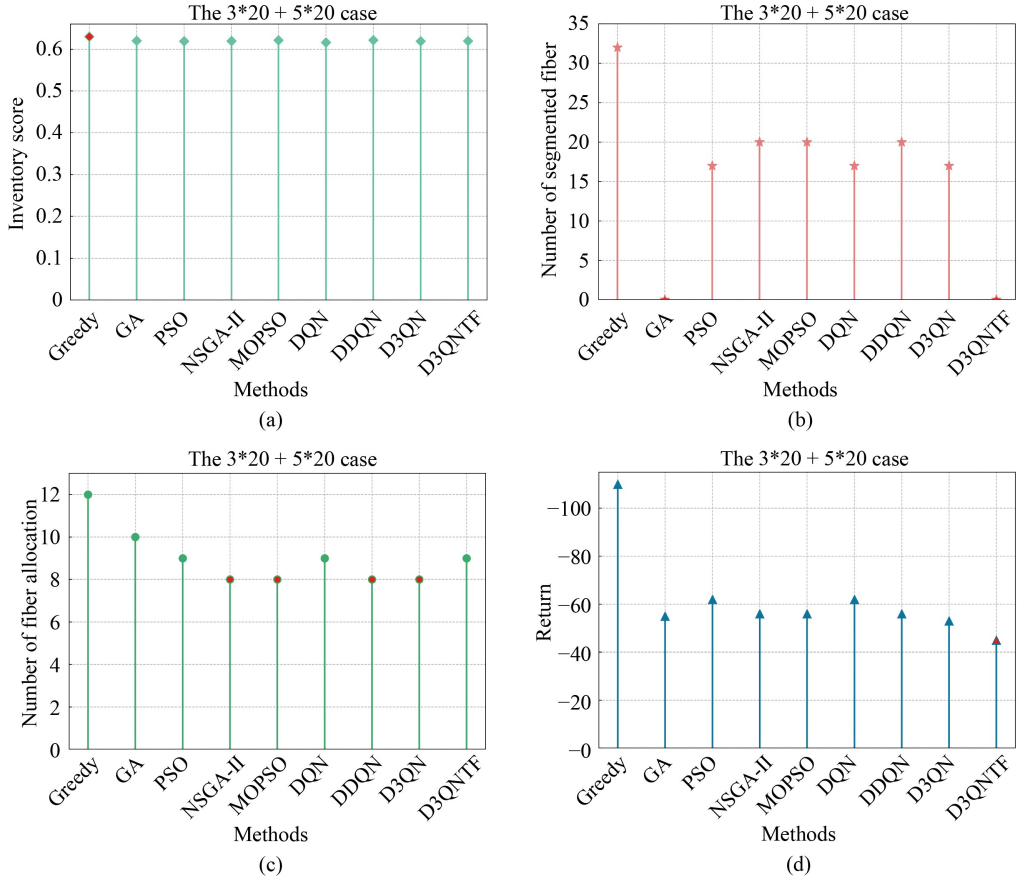

Fig. 6 Metrics of different algorithms for the 3*20 + 5*20 case. (a) Inventory score; (b) Number of segmented fiber; (c) Number of fiber allocation; (d) Return. Note: The red markers indicate the highest values achieved by the algorithm for the respective evaluation metrics, which are highlighted in bold in Table 4.

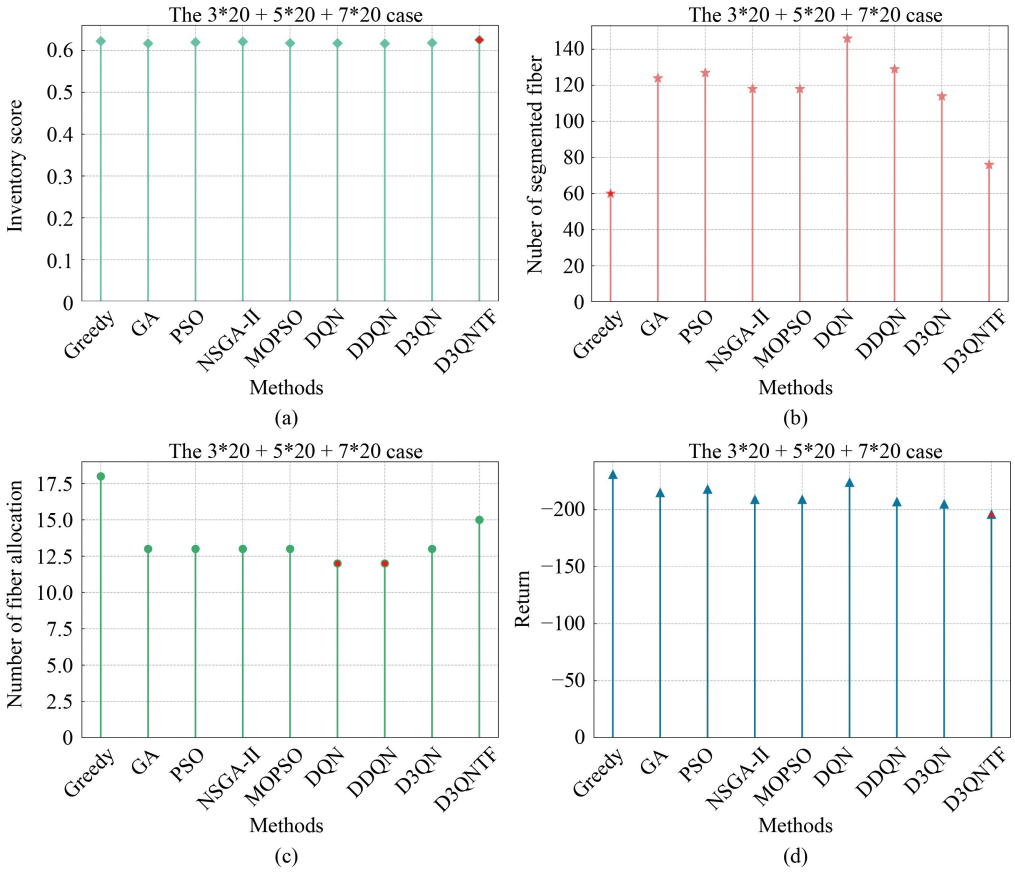

Fig. 7 Metrics of different algorithms for the 3*20 + 5*20 + 7*20 case. (a) Inventory score; (b) Number of segmented fiber; (c) Number of fiber allocation; (d) Return. Note: The red markers indicate the highest values achieved by the algorithm for the respective evaluation metrics, which are highlighted in bold in Table 4.

exploration. The approach helps reduce action-value overestimation, improves network learning stability, and enhances decision-making capabilities.

The agent interacts with the environment through continuous decisions, feedback, and adjustments to obtain the optimal strategy. Simulations confirm the model's effectiveness. Compared to traditional reinforcement learning, the D3QNTF model significantly improves stability and learning capacity. It handles high-dimensional challenges more efficiently and uses an innovative initialization strategy to enhance learning. As a result, decision-making quality is greatly improved, boosting performance in fiber allocation tasks. The D3QNTF model achieves better inventory scores, reduces segmented fibers, and optimizes fiber allocations, yielding the highest return values in all tests. These results demonstrate strong decision-making and robustness. However, the challenge of managing diverse customer orders in production scheduling remains unresolved. Future research should explore managing multiple orders within a single model.

Competing Interests The authors declare that they have no competing interests.

References

Fang J, Wang Z, Liu W, Chen L, Liu X (2024). A new particle-swarm-optimization- assisted deep transfer learning framework with applications to outlier detection in additive manufacturing. Engineering Applications of Artificial Intelligence, 131: 107700

Fujimoto S, Hoof H, Meger D (2018). Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning: 1587–1596

Gök M (2024). Dynamic path planning via Dueling Double Deep Q-network (D3QN) with prioritized experience replay. Applied Soft Computing, 158: 111503

Gui Y, Tang D, Zhu H, Zhang Y, Zhang Z (2023). Dynamic scheduling for flexible job shop using a deep reinforcement learning approach. Computers & Industrial Engineering, 180: 109255

Hausknecht M, Stone P (2015). Deep recurrent Q-learning for partially observable MDPs. In: 2015 AAAI Fall Symposium Series

Huang M, Hao Y, Wang Y, Hu X, Li L (2023). Split-order consolidation optimization for online supermarkets: Process analysis and optimization models. Frontiers of Engineering Management, 10(3): 499–516

Kiran B R, Sobh I, Talpaert V, Mannion P, Sallab A A A, Yogamani S, Perez P (2022). Deep reinforcement learning for autonomous driving: A survey. IEEE Transactions on Intelligent Transportation

Systems, 23(6): 4909–4926

Lee J, Perkins D (2021). A simulated annealing algorithm with a dual perturbation method for clustering. Pattern Recognition, 112: 107713

Li H, Liu H, Lan C, Yin Y, Wu P, Yan C, Zeng N (2023). SMWO/D: A decomposition-based switching multi-objective whale optimiser for structural optimisation of turbine disk in aero-engines. International Journal of Systems Science, 54(8): 1713–1728

Li H, Wang Z, Lan C, Wu P, Zeng N (2024). A novel dynamic multi-objective optimization algorithm with hierarchical response system. IEEE Transactions on Computational Social Systems, 11(2): 2494–2512

Liu J, Yuan S, Luo B, Biondi B, Noh H Y (2023). Turning telecommunication fiber-optic cables into distributed acoustic sensors for vibration-based bridge health monitoring. Structural Control and Health Monitoring, 2023: 1–14

Luo S, Zhang L, Fan Y (2021). Dynamic multi-objective scheduling for flexible job shop by deep reinforcement learning. Computers & Industrial Engineering, 159: 107489

Ma G, Wang Z, Liu W, Fang J, Zhang Y, Ding H, Yuan Y (2023). Estimating the state of health for lithium-ion batteries: A particle swarm optimization-assisted deep domain adaptation approach. IEEE/CAA Journal of Automatica Sinica, 10(7): 1530–1543

Martin E, Cervantes A, Saez Y, Isasi P (2020). IACS-HCSP: Improved ant colony optimization for large-scale home care scheduling problems. Expert Systems with Applications, 142: 112994

Mazyavkina N, Sviridov S, Ivanov S, Burnaev E (2021). Reinforcement learning for combinatorial optimization: A survey. Computers & Operations Research, 134: 105400

Ming F, Gong W, Li D, Wang L, Gao L (2023). A competitive and cooperative swarm optimizer for constrained multiobjective optimization problems. IEEE Transactions on Evolutionary Computation, 27(5): 1313–1326

Moerland T M, Broekens J, Plaat A, Jonker C M (2023). Model-based reinforcement learning: A survey. Foundations and Trends® in Machine Learning, 16(1): 1–118

Qiu C, Hu Y, Chen Y, Zeng B (2019). Deep deterministic policy gradient (DDPG)-based energy harvesting wireless communications. IEEE Internet of Things Journal, 6(5): 8577–8588

Silva Y L T, Subramanian A, Pessoa A A (2018). Exact and heuristic algorithms for order acceptance and scheduling with sequence-dependent setup times. Computers & Operations Research, 90: 142–160

Tan F, Yuan Z, Zhang Y, Tang S, Guo F, Zhang S (2024). Improved genetic algorithm based on rule optimization strategy for fibre allocation. Systems Science & Control Engineering, 12(1): 2347887

Tokic M (2010). Adaptive ε-greedy exploration in reinforcement learning based on value differences. In: Annual Conference on Artificial Intelligence. Berlin, Heidelberg: Springer, 203–210

Van Hasselt H, Guez A, Silver D (2016). Deep reinforcement learning with double Q-learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, 30(1): 2094–2100

Wang L, Hu X, Wang Y, Xu S, Ma S, Yang K, Liu Z, Wang W (2021b). Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. Computer Networks, 190: 107969

Wang X, Wang S, Liang X, Zhao D, Huang J, Xu X, Dai B, Miao Q (2024). Deep reinforcement learning: A survey. IEEE Transactions on Neural Networks and Learning Systems, 35(4): 5064–5078

Wang Y, Han Y, Gong D, Li H (2023a). A review of intelligent optimization for group scheduling problems in cellular manufacturing. Frontiers of Engineering Management, 10(3): 406–426

Wang Y, Liu W, Wang C, Fadzil F, Lauria S, Liu X (2023b). A novel multi-objective optimization approach with flexible operation planning strategy for truck scheduling. International Journal of Network Dynamics and Intelligence, 100002

Wang Y, Wu Z, Guan G, Li K, Chai S (2021a). Research on intelligent design method of ship multi-deck compartment layout based on improved taboo search genetic algorithm. Ocean Engineering, 225: 108823

Wang Z, Schaul T, Hessel M, Hasselt H, Lanctot M, Freitas N (2016). Dueling network architectures for deep reinforcement learning. In: International Conference on Machine Learning, 1995–2003

Xue Z, Zhang Y, Cheng C, Ma G (2020). Remaining useful life prediction of lithium-ion batteries with adaptive unscented kalman filter and optimized support vector regression. Neurocomputing, 376: 95–102

Yao F, Du Y, Li L, Xing L, Chen Y (2023). General modeling and optimization technique for real-world earth observation satellite scheduling. Frontiers of Engineering Management, 10(4): 695–709

Zhang Y, Chen L, Li Y, Zheng X, Chen J, Jin J (2021). A hybrid approach for remaining useful life prediction of lithium-ion battery with adaptive levy flight optimized particle filter and long short-term memory network. Journal of Energy Storage, 44: 103245

Zhao Y, Wang Y, Tan Y, Zhang J, Yu H (2021). Dynamic jobshop scheduling algorithm based on deep Q network. IEEE Access: Practical Innovations, Open Solutions, 9: 122995–123011

Zheng T, Zhou Y, Hu M, Zhang J (2023). Dynamic scheduling for large-scale flexible job shop based on noisy DDQN. International Journal of Network Dynamics and Intelligence, 100015

Zhong L, Zeng Z, Huang Z, Shi X, Bie Y (2024). Joint optimization of electric bus charging and energy storage system scheduling. Frontiers of Engineering Management, 11(4): 676–696