Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnline CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 59-10 (2025) 2070–2075

Learning Towards Fair Order Dispatching via Hierarchical Attention-based Reinforcement Learning for Garment Manufacturing

Yanying Wang*, Zhiheng Zhao*, George Q. Huang*

**Department of Industrial and Systems Engineering, The Hong Kong Polytechnic University, Hong Kong, China (email: ywonne.wang@connect.polyu.hk , zhiheng.zhao@polyu.edu.hk, gq.huang@polyu.edu.hk)*

Abstract: Garment production represents a typical form of social manufacturing, where orders are received centrally but processed in a decentralized manner. Factories are equipped with different processing capabilities that cater to highly tailored demands. Despite extensive research on production order allocation, the changeover costs in garment manufacturing and the fairness of earnings among factories remain largely neglected. This paper proposes Hierarchical Attention-based reinforcement learning for order dispatching in garment production. In this paper, we propose Hierarchical Attention-based reinforcement learning for order dispatching in garment production. Specifically, a novel Hierarchical Attention Network is introduced to model the complex relationships between factories and orders, as well as the long-term income fairness of factories. Finally, the proposed method is deployed on the Cyber-Physical Internet.

Keywords: Order dispatching problem, Deep reinforcement learning (DRL), Fairness, Self-attention.

1. INTRODUCTION

Apparel manufacturing is a highly customized and highly flexible production task. With the rapid development of O2O (Online-to-Offline) platforms, a new wave of fast fashion companies characterized by high delivery frequencies and low lead times has emerged in recent years. Leading this trend, companies like SHEIN have established flexible supply chains by forming deep partnerships with thousands of small and medium sized garment factories within regional networks. It enables a small-batch, quick-return production model.

In 2022, SHEIN achieved a Gross Merchandise Value (GMV) of $29 billion, surpassing traditional fast fashion brands such as Zara. By 2023, SHEIN's GMV had grown rapidly to nearly $45 billion, placing the company among the most highly valued unicorns globally. This demonstrates that the flexible manufacturing approach pioneered by SHEIN has become an inevitable and dominant trend in the apparel industry.

Community manufacturing serves as a critical framework for achieving highly flexible production. Within a five-kilometer radius of SHEIN's headquarters, there are thousands of associated suppliers and manufacturers. When an order arises, every production step—from printing to final cutting—can be undertaken by multiple factories. In this process, the company's task of allocating orders to factories represents a classic NP-hard Order Dispatch Problem (ODP). Specifically, factories upload available orders to the platform, which then assigns these orders to various factories. Finally, the factories complete the orders and deliver them back to the company.

One of the key factors in enhancing the overall efficiency of apparel manufacturing lies in addressing the platform's Order Dispatch Problem (ODP). The ODP involves the platform allocating appropriate order steps to factories within a short time frame, ensuring that orders are processed in the specified sequence. The goal is to optimize practical production objectives, such as minimizing completion time, delays, and production costs.However, the current order dispatching algorithms in apparel manufacturing face several typical challenges:

1) *Neglect of Factory Changeover Costs*: Current scheduling methods fail to account for the costs associated with production line changeovers, where workers require time and effort to adapt to new processes. This adaptation challenge reduces product quality and production efficiency. In traditional apparel manufacturing models, workers could focus on producing the same clothing style over an extended period, allowing their proficiency to improve. However, with the trend of small-batch orders and rapid style changes, workers often face the need to quickly adapt to new requirements just after mastering the previous process. Many workers, unable to cope with these frequent adjustments, choose to resign.

2) *Fairness in Factory Revenue*: Beyond considering production constraints, order dispatching must also account for the fairness of factory revenue to ensure the long-term sustainability of the entire system. These order dispatching platforms are often internally developed by companies, where efficiency—such as maximizing production profit—has been the primary metric for performance evaluation. However, prioritizing efficiency inevitably compromises the interests of factories. Over time, this imbalance can undermine the sustainability of the order dispatching system and even jeopardize the company's long-term viability.

To address the above challenges, this study proposes a Hierarchical Attention-based Reinforcement Learning Scheduling Framework to achieve fair order dispatching. Specifically, the contributions are as follows:

1) A novel Hierarchical Attention-based network is proposed to model complex relationships between orders and factories, including changeover costs and factory competition intensity, thereby enhancing decision-making quality.

10.1016/j.ifacol.2025.09.348

2) By integrating factory revenue considerations during the ranking phase, this framework accounts for the production preferences of factories. This approach helps to balance long-term revenue disparities among factories, ensuring the long-term sustainability of the order dispatching system.

2. RELATED WORKS

2.1 Order Dispatching and Flexible Job-Shop Scheduling Problem

The Order Dispatching Problem (ODP) typically refers to the process of reasonably dispatching a large number of orders to riders or drivers in real-time within a very limited decision time, as seen in ride-sharing services or on-demand food delivery services (Chen et al., 2024). The primary goals of dispatching are generally to minimize delivery time (X. Wang et al., 2023), improve delivery efficiency and customer satisfaction(Joshi et al., 2021), or maximize gross merchandise volume (GMV) (Shi et al., 2021). In these ODP scenarios, the dispatched entities are usually individuals with preferences and choices, such as riders or drivers. Consequently, research on such problems not only focuses on improving service efficiency but also ensures fairness in income distribution among individuals, thereby benefiting the platform's long-term performance (Sun et al., 2024). This bears certain similarities to order allocation in production factories, as factories, being revenue-generating production entities, also rely on fair income distribution to sustain retention on the platform.

However, the order allocation process in manufacturing factories differs from the aforementioned ODP scenarios. In food delivery and ride-hailing platforms, the order allocation process is completed in a single step, where orders are simply matched with service providers. In contrast, in garment manufacturing, a single order often consists of multiple steps that must follow a predefined sequence, and each step must be performed in a designated factory. This process closely resembles the Flexible Job-Shop Scheduling Problem (FJSP) (Lei et al., 2022).

FJSP is a task allocation problem commonly encountered in manufacturing, where, given a set of jobs and machines, the goal is to allocate operations of each job to machines with diverse capabilities and determine their processing sequence while meeting technological constraints (Song et al., 2023). The objective is typically to optimize realistic performance metrics such as minimizing the makespan. As a classic NP-hard scheduling problem, FJSP has been tackled using various methods, including exact algorithms, heuristics, and meta-heuristics (Lei et al., 2022).

With the development of Deep Reinforcement Learning (DRL), end-to-end DRL methods that automatically generate Priority Dispatching Rules (PDRs) have gained increasing attention. These methods utilize neural networks to capture the global state of scheduling problems, guiding decision-making actions toward maximizing rewards. This approach achieves a balance between accuracy and efficiency, making it a promising direction for solving complex scheduling challenges (Wang et al., 2024).

Based on the above analysis, the order dispatching problem in garment manufacturing not only requires consideration of income distribution fairness, as in typical ODP problems, but also must account for manufacturing sequence constraints, as in typical FJSP problems. Therefore, this paper proposes an end-to-end deep reinforcement learning method to model and address both aspects of these constraints.

2.2 Cypher-Physical Internet (CPI) and CPI Router

Cyber-physical internet (CPI) in manufacturing is to establish a new paradigm for manufacturing cooperation, similar to how sending and receiving instant messages over the internet is a solution (Qu et al., 2024).

The router is a primary component of a computer network that connects networks, responsible for routing and forwarding datagrams. Datagrams are typically forwarded from one router to another across the network until they reach their destination node (He et al., 2024).

Similarly, in this paper, the CPI Router serves as a critical information transmission tool for inter-factory production scheduling. By forwarding the order data package through the distributed CPI Routers in each factory, the system facilitates the dispatching of company orders, aggregates order information, and synchronizes the production status of factories and the processing status of orders in a timely manner.

Furthermore, the Internet Protocol Address (IP address) concept is introduced in the CPI. Using IP addresses to represent physical locations—compared to traditional latitude and longitude—enables the system to achieve hierarchical and logical addressing for regions, and also provides a vast address space and enhances user privacy protection.

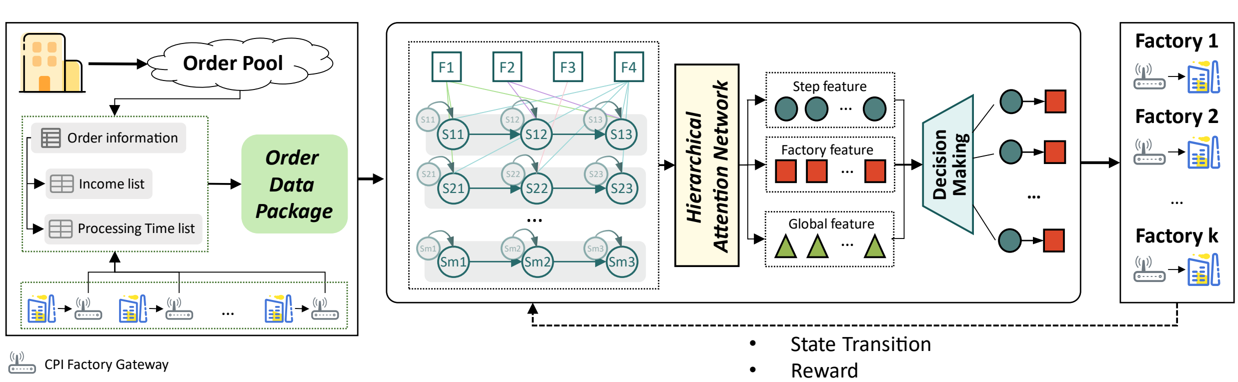


Figure 1. Overview of the proposed method.

3. PROPOSED METHOD

The overall framework of the proposed method is shown in Figure 1. First, after company orders are released, they are added to the order pool and incorporated into the order information by retrieving the processing status of factories through the CPI Router.

Next, in the order dispatch phase, scheduling is performed for the orders in the order pool, including feature extraction based on the Hierarchical Attention Network and decision-making based on reinforcement learning.

Finally, according to the decision-making results, orders are assigned to each factory, forming a task list for each factory, which is then sent to the factory's CPI Router. At the same time, rewards are generated based on the scheduling results, and the states are updated accordingly.

3.1 Order Data Design

In this section, we design the data package for orders in the system, including the order information generated by the company and the factory information aggregated through CPI router, to provide scheduling information for order dispatching.

The format of the order information issued by the company is shown in Table 1. Here: 'Step_sequence' indicates the processing sequence of the order. The order needs to go through three steps for completion. 'Requirement' specifies the requirements of the order, such as the process specifications required at each step, presented in natural language. 'Next_hop' indicates where the order will be sent after the current step is completed, represented using an IP address.

Different steps of an order need to be processed in different factories, and their corresponding relationships are shown in Table 3 and Table 2. This information is transmitted by each factory's CPI Router. As shown in Table 3, the production cost of a specific step for a single product in a certain factory is listed. Table 2 indicates the processing time required for a single product at a specific step in a factory. If a factory cannot produce the step, it is replaced with a very large value. Together, these tables form the order data package, which serves as the data foundation for the order dispatching stage.

3.2 Order Dispatching

This section first models the order scheduling process as a Markov Decision Process (MDP), including the state, action, transition, and reward. Then, it introduces the proposed Hierarchical Attention Network-based reinforcement learning (HAN-RL), which models and extracts features between steps and factories through HAN. By leveraging deep reinforcement learning, it selects steps and factories, ultimately outputting feasible step-factory (S-F) pairs. It should be noted that, as discussed in Section 1, the apparel community manufacturing scenario in this paper places factories within a few kilometers of each other. Therefore, we do not account for transportation costs between factories.

Table 2. Time (/pic in each factory) (Day)

Step	F1	F2	F3
S1	0.1	1e10	1e10
S2	0.1	0.1	1e10
S3	1e10	1e10	0.3

Table 3. Price (/pic in each factory) (USD)

Step	F1	F2	F3
S1	2	0	0
S2	2	1.5	0
S3	0	0	5

Table 1. Order Information table

Order_ID	0001
Step_sequence	S1-S3-S2
Order_quantity	100
Order_place_time	2023/10/22
Order_due_time	2023/11/09
Requirement	XXX
Next_hop	192.168.0.238

3.2.1 MDP Formulation

In the system, let there be a factory $F_k \in \mathcal{F}$ and an order set $O_i \in \mathcal{O}$. The time horizon is discretized into equal-length time slots $\mathcal{T} = \{1, 2, ...T\}$ .Each order consists of j steps that must be processed in a specific sequence, forming the step set $S_i = \{S_{i1}, S_{i2}, ...S_{ij}\}$ for each order. The set of all steps across all orders is denoted as $\mathcal{S} = \bigcup_i S_i$.

State*:* At a given time t, the decision-making process depends on: a) The set of all steps $S^t \in \mathcal{S}$, excluding those already completed and that do not affect subsequent scheduling. b) The set of all machines $F_k \in \mathcal{F}$, excluding those incapable of processing any unscheduled steps.

Let the compatible step-machine pairs between S^t and F^t be denoted as A^t. At time t, the feature vectors for each step S, factory F, and S-F pair are defined in the appendix. The state s_t at time t is defined as the combination of S^t, F^t, and A^t.

Action*:* The action $a_t \in \mathcal{A}^t$ is defined as the set of all feasible S-F pair at time t. A pair is considered feasible if all preceding steps of S_{ij} have been completed and the factory F_k.

Transition*:* After taking action a_t at time t, the state of all steps, factories, and S-F pairs is updated, resulting in a new s_{t+1}.

Reward*:* The reward should guide the agent to select actions that reduce the makespan while balancing system income fairness and efficiency. Thus, at time t, the reward consists of two components. For the makespan component, inspired by(Song et al., 2023), the lower bound of the completion time of S_{ij} at time t is denoted as $\underline{C}(S_{ij}, s_t)$. Assuming $\underline{C}(S_{i0}, s_t) = 0$, the lower bound of the completion time for unscheduled steps can be estimated as $\underline{C}(S_{ij}, s_t) = \underline{C}(S_{i(j-1)}, s_t) + \min_{k \in F_{ij}} p_{ij}^k$, where p_{ij}^k is the processing time of S_{ij} on factory F_k. Consequently, the makespan at time t can be expressed as $\max_{S_{ij} \in \mathcal{S}} \underline{C}(S_{ij}, s_t)$. The first component of the reward, $r1_t$ is then defined as the difference in makespan between time t and $t+1$:

$$r1_t = \max_{S_{ij} \in \mathcal{S}} \underline{C}(S_{ij}, s_t) - \max_{S_{ij} \in \mathcal{S}} \underline{C}(S_{ij}, s_{t+1}) \tag{1}$$

The cumulative $r1$ over the entire process is:

$$\sum_{t=0} r1_t = \max_{S_{ij} \in \mathcal{S}} \underline{C}(S_{ij}, s_0) - C_{\max} \tag{2}$$

Where $C_{\max}$ represents the makespan of all steps in the entire process, and $\max_{S_{ij} \in \mathcal{S}} \underline{C}(S_{ij}, s_0)$ is a constant. Therefore, minimizing the makespan is equivalent to maximizing $\sum_{t=0} r1_t$.

In terms of system revenue, to reduce long-term income disparities among factories and enhance the overall sustainability of the system, the factory accumulative income (FAI) and the Gross Merchandise Volume (GMV) in the system are defined as follows:

FAI: At the end of a certain time slot t, let the set of accumulated order steps received by factory F_k be $S_{ij}^t \in \mathcal{S}$. The factory accumulative income (FAI) of factory F_k at time t is then defined as $u_k^t = \sum_{S_{ij}^t \in \mathcal{S}} \upsilon_{ij}^k$, where υ_{ij}^k represents the processing price of step S_{ij} at factory F_k.

GMV: The total FAI of all factories at the end of the time horizon is defined as the Gross Merchandise Volume (GMV) of the platform, denoted as $G = \sum_{F_k \in \mathcal{F}} u_k^T$.

This section adopts GMV to measure the platform's long-term efficiency and uses Max-Min Fairness (Sun et al., 2024) to evaluate income fairness among individual factories. Accordingly, the second part of the reward is defined as:

$$r2_t = \max\{(1-\beta)\min_{F_k \in \mathcal{F}} u_k^t + \frac{\beta}{N}\sum_{F_k \in \mathcal{F}} u_k^t\} \tag{3}$$

Where, β is the coefficient that balances platform efficiency and fairness.
The final reward of the MDP process at time t is defined as:

$$r_t = \alpha \sum_{t=0} r1_t + (1-\alpha) r2_t \tag{4}$$

In this context, α is the coefficient that balances the platform's revenue with production efficiency.
Policy: The policy $\pi(a_t \mid s_t)$ is defined as the probability distribution over the action set A^t given the state s_t. This distribution will be derived using the proposed Deep Reinforcement Learning (DRL) algorithm in this paper.

3.2.2 Hierarchical Attention Network

The structure of the Hierarchical Attention Network is shown in Figure 3. This network is divided into two modules, which are responsible for feature extraction of step features and factory features, respectively. The step attention module includes two attention mechanisms, while the factory attention module includes one attention mechanism. All three attention mechanisms are based on the self-attention mechanism. The extracted step and factory features are then combined to form the global features.

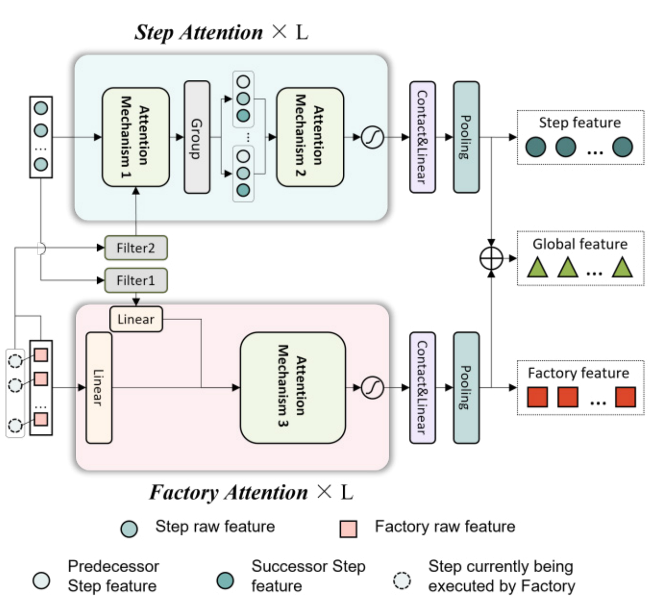


Figure 3. Hierarchical Attention Network.

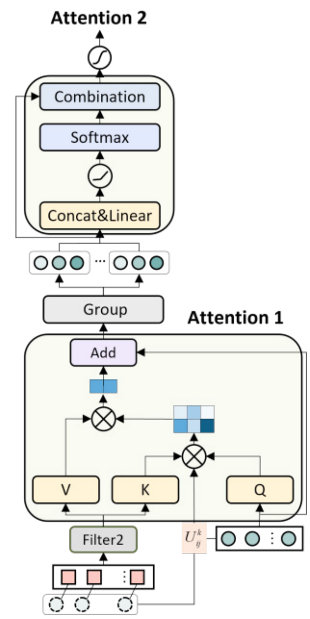


Figure 2. Attention mechanism in Step attention.

Step attention

In the Step Attention module, it uses attention mechanism to make the step focusing on the factory with the lowest changeover cost.

Specifically, in Attention Mechanism 1, the competing factory set that is idle and capable of executing step S_{ij} is denoted as M_{ij} (obtained through Filter 2). Attention Mechanism 1 calculates the attention coefficient e_{ijk} of S_{ij} with respect to factory $F_{ij}^k \in M_{ij}$. Let the most recent step executed by factory $F_{ij}^k \in M_{ij}$ be denoted as S^q. Hence, U_{ij}^k represents the setup time required to switch from S^k to S_{ij} in factory F_{ij}^k, and this setup time is predefined according to product and process requirements. Therefore, the attention coefficient e_{ijk} is calculated as follows:

$$e_{ijk} = \text{softmax}(\frac{1}{\sqrt{d}} Q_{ij} K_k \text{-} \lambda U_{ij}^k) \tag{5}$$

$$h_{ijk} = \sum_{k=1}^{M} e_{ijk} V_k \tag{6}$$

$$h_{S_{ij}}^M = h_{S_{ij}} + h_{ijk} \tag{7}$$

Where $Q_{ij} = h_{S_{ij}} W_Q$, $K_k = h_{F_{ij}^k} W_K$, $V_k = h_{F_{ij}^k} W_V$, $h_{S_{ij}} \in \mathbb{R}^{d_S}$, $h_{F_{ij}^q} \in \mathbb{R}^{d_F}$, $W_Q \in \mathbb{R}^{d_S \times d}$, $W_K \in \mathbb{R}^{d_F \times d}$, $W_V \in \mathbb{R}^{d_F \times d}$, λ is a coefficient that regulates how much the model considers setup time.
As shown in (5), this formula explicitly incorporates the switching cost. A higher probability corresponds to a factory with lower switching cost and matching features. By applying

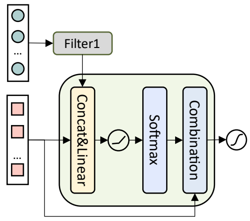


Figure 4. Attention mechanism in Factory attention.

residual connections, the final step features are thus augmented to reflect the factory's switching cost.

Attention Mechanism 2 input step and its preceding and succeeding steps within the same order to identify the most significant steps. Specifically, let the preceding node of step S_{ij} be denoted as $S_{i(j-1)}$ and the succeeding node as $S_{i(j+1)}$.The attention coefficients of S_{ij} with respect to its preceding, succeeding nodes, and itself are calculated as shown in Equation (8). After applying softmax normalization, the output step feature vector $h'_{S_{ij}}$ is obtained.

$$e_{ijp} = \mathrm{LeakyReLU}(\vec{\mathrm{b}}^{\mathrm{T}}[\mathrm{W}h^{M}_{S_{ij}} \| \mathrm{W}h_{S_{ip}}]) \tag{8}$$

$$e'_{ijp} = \mathrm{softmax}(e_{jip}) \tag{9}$$

$$h'_{S_{ij}} = \sigma(\sum e'_{ijp}\, \mathrm{W}h_{S_{ip}}) \tag{10}$$

Where $p = j-1, j, j+1$, $\mathrm{W} \in \mathbb{R}^{d_S' \times d_S}$, $\vec{\mathrm{b}}^{\mathrm{T}} \in \mathbb{R}^{2d_S'}$. And $\|$ represents the concatenation of vectors。

Factory attention

In the Factory Attention module, inspired by (Wang et al., 2024), Attention Mechanism 3 introduces the competing steps between factories. This models the competition relationships among factories, enabling the model to focus on the most critical factory.

Specifically, Attention Mechanism 3 introduces c_{km} as a measure of the competition intensity between F_k and F_m (obtained through filter1). Here, C_{km} is defined as the set of competing steps between F_k and F_m, and $\mathcal{N}_k$ represents the set of factories competing with F_k, $F_m \in \mathcal{N}_k$. The attention score of F_k towards $F_m \in \mathcal{N}_k$ is calculated as follows:

$$c_{km} = \sum_{S_{ij} \in C_{km}} h_{S_{ij}} \tag{11}$$

$$u_{km} = \mathrm{LeakyReLU}(\vec{\mathrm{d}}^{\mathrm{T}}[\mathrm{Z}^1 h_{F_k} \| \mathrm{Z}^1 h_{F_m} \| \mathrm{Z}^2 c_{km}]) \tag{12}$$

$$u'_{km} = \mathrm{softmax}(u_{km}) \tag{13}$$

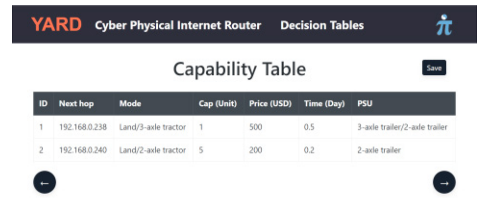
$$h'_{M_k} = \sigma(u'_{kk}\, \mathrm{Z}^1 h_{F_k} + \sum_{F_m \in \mathcal{N}_k} u'_{km}\, \mathrm{Z}^1 h_{F_m}) \tag{14}$$

Where $h_{S_{ij}} \in \mathbb{R}^{d_S}$, $h_{F_k}, h_{F_m} \in \mathbb{R}^{d_F}$, $Z^1 \in \mathbb{R}^{d_F' \times d_F}$, $Z^2 \in \mathbb{R}^{d_F' \times d_S}$, $\vec{\mathrm{d}}^{\mathrm{T}} \in \mathbb{R}^{3d_F}$.

Multi-Head Attention and Pooling

To capture more global information and avoid the model excessively focusing on its own position. Thus, this paper adopts a multi-head attention mechanism to enhance the model's learning ability. As shown in Figure 3, the HAN employs H layers of attention. The outputs from each attention layer are concatenated and transformed into the target dimension through a linear layer. After passing through H layers of HAN, the step features are denoted as $h^{(H)}_{S_{ij}}$, and the factory features are denoted as $h^{(H)}_{F_k}$. Both are subjected to average pooling to obtain the global feature.

Figure 5. CPI-Router Deployed on Raspberry Pi



ID	Next hop	Mode	Cap (Unit)	Price (USD)	Time (Day)	PSU
1	192.168.0.238	Land/3-axle tractor	1	500	0.5	3-axle trailer/2-axle trailer
2	192.168.0.240	Land/2-axle tractor	5	200	0.2	2-axle trailer

Figure 6. Sample Data Table of the CPI-Router.

$$h^{(H)}_G = [\frac{1}{|\mathcal{S}|}\sum_{S_{ij} \in \mathcal{S}} h^{(H)}_{S_{ij}} \| \frac{1}{|\mathcal{F}|}\sum_{F_k \in \mathcal{F}} h^{(H)}_{F_k}] \tag{15}$$

3.2.3 Decision Making

This section employs an actor-critic RL-based network for decision-making. The objective of the actor network is to generate the probability of distribution of actions, i.e., the policy network $\pi(a_t \mid s_t)$. In this part, the learned step features, factory features, global features, and raw S-F features are contact and used as inputs and are passed through MLP layers：

$$P(a_t, s_t) = \mathrm{MLP}_{\omega}[h^{(H)}_{S_{ij}} \| h^{(H)}_{F_k} \| h^{(H)}_G \| h_{(S_{ij}, F_k)}] \tag{16}$$

Then, the outputs are passed through a softmax layer to generate the action probability distribution:

$$\pi_{\omega}(a_t \mid s_t) = \frac{\exp(P(a_t, s_t))}{\sum_{a'_t \in A_t} \exp(P(a'_t, s_t))} \tag{17}$$

Similarly, the critic network processes the global feature through another MLP layer to estimate the value of the state:

$$v(s_t) = \mathrm{MLP}_{\phi}(h^{(H)}_G) \tag{18}$$

4. CASE STUDY

This section designs a router for each factory based on the CPI router proposed in Section 3, deploying it on a Raspberry Pi. The detailed system setup is shown in Figure 5.

Each factory's corresponding router stores the factory's current production information and the details of the orders being processed, as shown in Figure 6. Communication between multiple CPI routers enables the synchronization of production information within the Cyber-Physical Internet.

5. CONCLUSIONS

To address the challenges in garment manufacturing order allocation, including (1) the neglect of changeover costs, which reduces production efficiency, and (2) the disregard for long-term factory income fairness, which undermines platform sustainability, this paper proposes a Hierarchical Attention-based Reinforcement Learning Scheduling Framework. The Hierarchical Attention Network incorporates two attention modules to model (i) the changeover costs between consecutive steps, (ii) the competitive relationships and long-term income levels among factories, and (iii) the complex interactions between steps and factories. Furthermore, a CPI Router is proposed and deployed in factories to enable real-time synchronization of scheduling information across the network. Ultimately, this framework achieves fair order dispatching in garment manufacturing.

ACKNOWLEDGEMENTS

The authors would like to express thanks to the financial support from the National Natural Science Foundation of China (No. 52305557), Guangdong Basic and Applied Basic Research Foundation (No. 2024A1515011930), Innovation and Technology Fund (PRP/015/24TI), Hong Kong RGC TRS Project(T32-707/22-N), Collaborative Research Fund (C7076-22GF), and Research Impact Fund (R7036-22).

APPENDIX

At time t, the feature vector is defined as follows:

Feature of Step S_{ij}

1) Minimum processing time among all factories.
2) Average processing time among all factories.
3) Span of processing time among all factories.
4) Proportion of factories that S_{ij} can be processed.
5) A binary value indicates whether S_{ij} has been scheduled or not.
6) Estimated complete time $\underline{C}(O_{ij}, s_t) = \underline{C}(O_{i(j-1)}, s_t) + \min_{k \in M_{ij}} p_{ij}^k$.
7) The number of unscheduled steps in O_i.
8) The sum of average processing time of unscheduled steps in O_i.
9) Waiting time that the time from the ready time until T_s.
10) Remaining Processing Time that the time from T_s until the completion time.

Feature of Factory F_k

1) The most recently executed step S^k of factory F_k.
2) The estimated cumulative income u_k^t of factory F_k up to t.
3) Minimum processing time among all steps.
4) Average processing time of steps that F_k can process.
5) Free time that the moment when F_k is free.
6) Waiting time that the time from the free time until T_s.
7) Working tag, if F_k is working then the value is 1.
8) Remaining processing time, the time from T_s until free time.

Feature of Step-Factory pair (S_{ij}, F_k)

1) The processing price υ_{ij}^k of step S_{ij} at factory F_k.
2) The processing time p_{ij}^k of step S_{ij} at factory F_k.
3) Ratio of p_{ij}^k to the maximum processing time of S_{ij}.
4) Ratio of p_{ij}^k to the maximum processing time of candidates that F_k can process.
5) Ratio of p_{ij}^k to the maximum processing time of unscheduled steps.
6) Ratio of p_{ij}^k to the maximum processing time of unscheduled steps that F_k can process.
7) Ratio of p_{ij}^k to the maximum processing time of compatible pairs Fair Order Dispatching.
8) Ratio of p_{ij}^k to remaining workload of O_i.
9) Summation of waiting time of S_{ij} and F_k.

REFERENCES

Chen, J.-F., Wang, L., Liang, Y., Yu, Y., Feng, J., Zhao, J., & Ding, X. (2024). Order Dispatching Via GNN-Based Optimization Algorithm for On-Demand Food Delivery. *IEEE Transactions on Intelligent Transportation Systems*, *25*(10), 13147–13162. https://doi.org/10.1109/TITS.2024.3389090

He, S., Zhang, M., Wang, S., & Huang, G. Q. (2024). Pricing and revenue allocation mechanisms for regional logistics networks in Cyber-Physical Internet. *Transportation Research Part C: Emerging Technologies*, *165*, 104714. https://doi.org/10.1016/j.trc.2024.104714

Joshi, M., Singh, A., Ranu, S., Bagchi, A., Karia, P., & Kala, P. (2021). Batching and Matching for Food Delivery in Dynamic Road Networks. *2021 IEEE 37th International Conference on Data Engineering (ICDE)*, 2099–2104. https://doi.org/10.1109/ICDE51399.2021.00207

Lei, K., Guo, P., Zhao, W., Wang, Y., Qian, L., Meng, X., & Tang, L. (2022). A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem. *Expert Systems with Applications*, *205*, 117796. https://doi.org/10.1016/j.eswa.2022.117796

Qu, X., Li, M., Ouyang, Z., Ng, C., & Huang, G. Q. (2024). Routing protocols for B2B e-commerce logistics in cyber-physical internet (CPI). *Computers & Industrial Engineering*, *193*, 110293. https://doi.org/10.1016/j.cie.2024.110293

Shi, D., Tong, Y., Zhou, Z., Song, B., Lv, W., & Yang, Q. (2021). Learning to Assign: Towards Fair Task Assignment in Large-Scale Ride Hailing. *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 3549–3557. https://doi.org/10.1145/3447548.3467085

Song, W., Chen, X., Li, Q., & Cao, Z. (2023). Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Transactions on Industrial Informatics*, *19*(2), 1600–1610. https://doi.org/10.1109/TII.2022.3189725

Sun, J., Jin, H., Yang, Z., & Su, L. (2024). Optimizing Long-Term Efficiency and Fairness in Ride-Hailing Under Budget Constraint via Joint Order Dispatching and Driver Repositioning. *IEEE Transactions on Knowledge and Data Engineering*, *36*(7), 3348–3362. https://doi.org/10.1109/TKDE.2023.3348491

Wang, R., Wang, G., Sun, J., Deng, F., & Chen, J. (2024). Flexible Job Shop Scheduling via Dual Attention Network-Based Reinforcement Learning. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(3), 3091–3102. https://doi.org/10.1109/TNNLS.2023.3306421

Wang, X., Wang, L., Wang, S., Pan, J., Ren, H., & Zheng, J. (2023). Recommending-and-Grabbing: A Crowdsourcing-Based Order Allocation Pattern for On-Demand Food Delivery. *IEEE Transactions on Intelligent Transportation Systems*, *24*(1), 838–853. https://doi.org/10.1109/TITS.2022.3209722