

قضايا بحثية

البحث: The Impact of Overall Optimization on
Warehouse Automation

التكليف الثالث لمادة قضايا بحثية
محمد عبدالله الزاير
242000178



العنوان

متغير مستقل

متغير تابع

The Impact of **Overall Optimization** on **Warehouse Automation**

عنوان الدراسة مثير
للاهتمام خاصة
بالنسبة لي
كمختص بمجال
المستودعات
واحتوى على متغير
مستقل ومتغير تابع،
ولكن يمكن أن أوجه
له انتقادا سلبيا
للباحثين لأنهما لما
يحدد النطاق
الجغرافي للدراسة
في العنوان



المستخلص والمقدمة

المستخلص: قدم المستخلص تعريفا مختصرا بين لي كقارئ ما يمكن أن أتوقعه من الدراسة البحثية، ويعتبر طول المستخلص مناسباً، ولكن أرى به بعض جوانب القصور:

- لم يتم الإشارة بشكل مختصر إلى مجتمع البحث ولا وجود لأي إشارة عن طرق جمع البيانات.
- لم يتم ذكر الكلمات المفتاحية الخاصة بالدراسة البحثية في المستخلص.

المقدمة:

- طول المقدمة مناسب.
- تم تدعيم المقدمة ببعض المعلومات مع ذكر مصادرها وهذا أمر جيد.
- حدد الباحثان الأسباب التي دفعتهما للقيام بالدراسة البحثية وما هو متوقع منها تقديمها للحقل العلمي.
- حدد الباحثان ثلاثة أسئلة يهدف البحث لتقديم إجابات عليها.
- لم يتم الإشارة إلى لمحات من النتائج التي توصلت إليها الدراسة ولا التوصيات وهذا أمر سلبي.

مفاهيم الدراسة والإطار النظري

مفاهيم الدراسة:

كوني متخصص في مجال المستودعات تحديدا فإني ملم بالمصطلحات المرتبطة بهذا المجال، ولكن كان ينبغي أن يقدم الباحثان شرحا مبسطا للمصطلحات المستخدمة في الدراسة البحثية كونها دراسة تخصصية في مجال معين، وقد تكون بعض المصطلحات غير مألوفة حتى للعاملين بالمجال اللوجستي بأقسامه الأخرى غير المستودعات، خاصة أن بعض الباحثين ذكروا بعض الاختصارات دون شرح معناها وهي وإن كانت بسيطة لنا نحن المتخصصين إلا أنها لن تكون كذلك للغير مثل MHE وهي الاختصار لـ Material Handling Equipment.

الإطار النظري:

- قدم الباحثين إطارا نظريا يوضح بعض المفاهيم، إلا أنني أراه غير كافٍ لإعطاء شرح واضح حول مفاهيم البحث، وكان ينبغي أن يتم التفاصيل فيه بشكل أوسع.
- وثق الباحثين الجوانب النظرية التي تم شرحها وأسندوها لمصادرهما وهذا أمر مهم وجيد.



منهجية الدراسة

هذه الدراسة مبنية على التجربة العملية، وقد استخدم الباحثان المحاكاة، وهذا النوع من الدراسات يختلف اختلافا جذريا عن الدراسات النظرية.

شرح الباحثان مشكلة البحث وحددا تحديان رئيسيان لتطبيق مفهوم MARL بالمستودعات.

تم إجراء محاكاة لعمل المستودعات المؤتمتة بالكامل في عملية تحضير الطلبات داخل المستودعات، وذلك للتمكن من الإجابة التي يهدف البحث للإجابة عليها.



فرضيات الدراسة

لم يتم تحديد فرضيات لهذه الدراسة وهذا أمر سلبي، فالدراسة حتى وإن كانت تطبيقية إلا أنه من الممكن تحديد فرضيات معينة مبنية على أسس منطقية ويتوقع الباحثين حدوثها، ومن ثم اختبار تلك الفرضيات وتوضيح إذا ما كانت الفرضيات صحيحة أو إثبات خلافها.



مناقشة النتائج والتوصيات

أوضح الباحثان النتائج التي تم توصل إليها في الدراسة، وكانت كما هي باقي الدراسة تتسم بالتعقيد نوعا ما ولم يتم شرحها بطريقة سهلة الفهم للقارئ خاصة غير المتخصص، بل أنها صعبة أيضا حتى للمتخصصين ممن لا يألّفون التعامل مع المستودعات المؤتمتة.

لم يقدم الباحثان أي توصيات بخصوص الدراسة وهذا أمر سلبي أيضا.

دراسة من هذا النوع تحتاج إلى تفصيل أكبر وأن تكون أكثر عمقا ولكن أسهل للفهم في الوقت ذاته.

The Impact of Overall Optimization on Warehouse Automation

Hiroshi Yoshitake¹ and Pieter Abbeel²

Abstract—In this study, we propose a novel approach for investigating optimization performance by flexible robot coordination in automated warehouses with multi-agent reinforcement learning (MARL)-based control. Automated systems using robots are expected to achieve efficient operations compared with manual systems in terms of overall optimization performance. However, the impact of overall optimization on performance remains unclear in most automated systems due to a lack of suitable control methods. Thus, we proposed a centralized training-and-decentralized execution MARL framework as a practical overall optimization control method. In the proposed framework, we also proposed a single shared critic, trained with global states and rewards, applicable to a case in which heterogeneous agents make decisions asynchronously. Our proposed MARL framework was applied to the task selection of material handling equipment through automated order picking simulation, and its performance was evaluated to determine how far overall optimization outperforms partial optimization by comparing it with other MARL frameworks and rule-based control methods.

I. INTRODUCTION

In recent years, industrial automation has rapidly advanced with the active installation of robots. Many industrial sites, such as logistics warehouses and production lines, are experiencing labor shortages. They also require highly efficient and long-hour operations to deal with bulk orders promoted by E-commerce [1]. To address these issues, manual operations are automated by replacing human laborers with industrial robots. The installation of industrial robots increases at an annual rate of $\sim 10\%$, and robotic automation helps compensate for the workforce shortage [2]. More efficient automation techniques are also required to improve the effectiveness of robot installation.

One of the reasons why the installation of automated systems is expected to increase efficiency is the possibility of further efficient operations based on overall optimization. In conventional manual operations, detailed control and modeling are challenging due to the several uncertainties associated with human actions and decision making. If automation progresses using industrial robots, their predictable and deterministic behaviors will increase the control accuracy in operations. By constructing such highly accurate control schemes in various processes, there is a prospect of optimizing the entire system. However, most of the previous studies have only addressed partial optimization for limited cases, such as a part of an automated system or material handling equipment (MHE) for a certain process [3]. It is unclear

how much the overall optimization of multiple processes will affect the automated system.

Multi-agent reinforcement learning (MARL) can provide practical overall optimization for controlling industrial automated systems. A control method of robot coordination for the overall optimization has not yet been established. However, centralized training with decentralized execution (CTDE), one of the major MARL frameworks in which agents make decisions in a decentralized manner and learn the coordination using centralized estimators [4], would be a solution for it: cyber-physical modeling enables the training of reinforcement learning (RL) agents centrally [5]. Furthermore, a decentralized RL policy would be reasonable for practical implementation: a centralized controller (e.g., warehouse management system) would not control thousands of robots one by one in terms of system load [6]. In this case, it is more natural that the robot fleet is controlled by a decentralized sub-system operating the corresponding process. Furthermore, there is currently no unified controller that can handle various types of MHE and robots released from different vendors.

In this study, we investigate the impact of overall optimization on automated systems, particularly warehouse automation using different MARL frameworks. There are two main challenges for the MARL-based approach in robot coordination, which is essential for the overall optimization of industrial automation. The first one is a long-horizon task under extremely sparse reward settings: evaluation indices of system performance such as task completion time (makespan) and productivity can be estimated only at the terminal state when all tasks are completed. The second one is the asynchronous decision making of heterogeneous agents: when various types of robots execute tasks, they do not make decisions under the synchronous settings that major MARL algorithms assume. We, therefore, examine an effective MARL framework that maximizes the performance of automated systems under these challenges by introducing a simplified automated warehouse simulator.

The purpose of this study is to answer the following research questions (RQs).

- **RQ1:** How can MARL agents acquire coordinated behaviors in industrial automated environments?
- **RQ2:** Which coordination condition is important for overall optimization?
- **RQ3:** What advantages does overall optimization bring to automated systems compared with partial optimization?

The contributions of this study are as follows.

¹H. Yoshitake is with R&D Group, Hitachi Ltd., Kokubunji-shi, Tokyo, Japan hiroshi.yoshitake.nt@hitachi.com, ²P. Abbeel is with Department of EECS, University of California, Berkeley, CA, USA pabbeel@cs.berkeley.edu

- Proposal of a CTDE-based practical MARL algorithm applicable to industrial automated environments.
- Building a simulation environment of an automated order picking system¹.
- Clarifying effective coordination conditions and the impact of overall optimization when using the proposed MARL algorithm and simulation environment.

The remainder of this paper is organized as follows. In §II, we explain related works. We provide a detailed description of the proposed MARL frameworks and algorithms in §III. In §IV, we evaluate the proposed methods. We conclude the study and discuss future work in §V.

II. RELATED WORKS

A. Optimization of Logistics Warehouse Operation

The operational purpose of logistics warehouses is to collect and ship inventory items according to orders received from customers [7]. To achieve this order fulfillment, warehouse operations consist of several processes: receiving, inventory control, order picking, inspection, packing, and shipping. Each process also includes the workforce such as human laborers and MHE, as well as their tasks to handle the ordered items. Optimizing these processes helps maintain efficient warehouse operations, and thus, minimizing the makespan and maximizing the system throughput are key research topics for warehouse optimization [8]. Because warehouse automation has made operation control more accurate by replacing human laborers with robots, optimization has become even more crucial for efficiency [9].

Maximizing the efficiency of warehouse operations requires control technology that comprehensively optimizes all related processes. Despite the widely recognized need for such technology, previous studies have been limited to partial optimization within a certain process such as optimal routing and order batching for manual/robotized order picking and optimal storage assignment based on order frequency for inventory control [3]. Therefore, an overall optimization method that considers the operation of multiple processes has not yet been established. In this study, we introduce a state-of-the-art (SOTA) MARL algorithm as an overall optimization method for warehouse operations.

B. Multi-Agent Reinforcement Learning

MARL, an extension of the RL framework, where several agents execute different tasks, is extensively studied for acquiring the optimal policies of agents in a multi-agent system (MAS). Agents need to learn by considering not only information about their local environments but also other learning agents. Many MARL frameworks have been proposed depending on various conditions such as operational settings, where agents are completely controlled by a central unit or operate autonomously in decentralized settings, and situations in which the tasks of agents are stationary or non-stationary [4].

MARL-based control with a decentralized policy would be reasonable for automated systems due to its scalability. The simplest framework of such control is independent learning (IL), where each agent is trained in the same manner as in single-agent RL: an agent independently learns the policy from its local observations and behaviors [10] even though it suffers from instability of the training environment caused by the policy updates of other agents. Recent MARL studies have focused on the CTDE framework where decentralized policies are trained by centralized critics that estimate the contributions of all agents. One of the most widely referenced CTDE frameworks is the multi-agent deep deterministic policy gradient (MADDPG), which uses joint critics of the states and actions of all agents [11]. We apply both IL and CTDE frameworks to automated warehouse operations and clarify how much they can contribute to the optimization.

C. Applications of MARL in Industrial Automation

Applications of MARL are frequently studied in many industrial fields such as manufacturing, logistics, networking, and automotive [12], [13]. Most industrial systems require intelligent controllers that create operation schedules or send instructions to the control object to maximize performance. RL provides reasonable solutions to these problems compared with other optimization techniques [14] and has the advantage of flexibility: RL agents are trained by experiencing various situations in a system, and the trained policy can output optimal actions for any states. Because unexpected variable factors such as noises and disturbances tend to occur in industrial systems, the control of agents, resource allocation, and task planning is expected to be flexible in response. The goal of MARL research in the industrial field is to incorporate the above RL features into systems consisting of several autonomous agents.

One of the most popular MARL applications is the control of MHE used for automating order picking in logistics warehouses and factories [15]. Order picking is an operation, where ordered items are collected for shipping destinations from warehouse storage. The transfer of items during order picking has been recently automated by MHE consisting of several automated guided vehicles (AGVs) [3]. The task allocation or path planning of AGVs has been successfully executed by introducing recent CDTE algorithms [15], [16]. However, such MARL-based control has only been applied to the process of item transfer by homogeneous robots, whereas order picking includes other processes involving heterogeneous robots such as picking and placing items by picking robots and subsequent transfer of items by conveyors. Few previous studies have addressed more complicated cases that allow training heterogeneous agents with asynchronous decision making [17]–[19]; however, they used the MADDPG (i.e., off-policy MARL algorithm), making it difficult to apply them to the sparse reward setting, one of the typical features in industrial automation. In this study, we develop an MARL framework that can be applied to several processes with heterogeneous agents.

¹The source code is available at <https://github.com/16444take/aope-sim.git>

III. METHODOLOGY

A. Problem Settings

Although the operational status of automated systems can be disturbed by noise and stochastic factors, it depends on the decision making for controlling MHE and robots that perform the tasks in each process. By regarding the controller of MHE and robots as an autonomous agent, state changes in an automated system can be described as a Markov decision process: $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}_T, \mathcal{R} \rangle$, where $s \in \mathcal{S}$ denotes a set of system states, $a \in \mathcal{A}$ denotes a set of agents' actions, $\mathcal{P}_T : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0, 1]$ denotes the state transition probability organized by the system operations, and $r \in \mathcal{R} : \mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow \mathbb{R}$ denotes the reward function, respectively. Assuming a situation, where the controller makes decisions solely based on the states of its corresponding process, we further study the decentralized partially observable MDP as $\langle \mathcal{S}, \{\mathcal{A}^i\}, \mathcal{O}, \mathcal{P}_T, \mathcal{R}, N \rangle$, where $o^i \in \mathcal{O}$ denotes the local observation for agent $i \in N$ at global state s , and $a^i \in \mathcal{A}^i$ denotes an action set of each agent [20]. In a finite horizon setting with length T , agent i is trained with its policy π^i to maximize a discounted accumulated reward at time-step t computed as $J_\pi^i = \mathbb{E}_\pi \left[\sum_{\tau=0}^{T-t} \gamma^\tau r_{\tau+t}^i \right]$, where γ denotes a discount factor for accumulating the rewards.

There are two major challenges when we apply the MARL framework to the flexible coordination of agents in an industrial automated environment.

- Long trajectory and extremely sparse reward (LTESR): an automated system is expected to maximize performance based on evaluation indices such as makespan and productivity estimated at the end of operations. Agents are, therefore, rewarded only at the last state transition even though they experience many state transitions to complete all tasks.
- Asynchronized multi-agent (MA) decision making (AMADM): each agent, which corresponds to an automated process or robot, performs its tasks asynchronously. Most CTDE frameworks assume synchronized behaviors among agents for estimating joint state(-action) values, and few previous studies on asynchronous settings described in §II.C can only be applied to off-policy RL algorithms with dense reward settings.

B. Proposal of MARL Framework for Industrial Automation

1) *Foundational Framework*: to answer **RQ1**, we propose a CTDE framework with a shared critic, termed CDSC: we adopt an actor-critic (AC) architecture for CDSC, where an actor policy π and a critic $\hat{V}$ are modeled by different neural networks. Although widely referenced CTDE frameworks such as the MADDPG introduce individual joint critics that can be trained cooperatively with global state $s_t = \bigcup_{i=1}^N o_t^i$ and global reward r_t^g , they can only train networks based on state transitions caused by themselves under the AMADM settings. Thus, CDSC can provide more globalized training conditions for agents by aggregating all transition histories into a single critic.

2) *Algorithm Design*: to address the LTESR issue, we used an on-policy AC algorithm in the proposed MARL framework: The LTESR makes state values, estimated as TD errors frequently used in off-policy RL algorithms, uncertain because of many non-rewarded state transitions. We, therefore, used the on-policy algorithm in our proposed MARL frameworks to estimate state values as reward-to-go using the Monte Carlo (MC) method. As a SOTA RL algorithm, proximal policy optimization (PPO), which can train the policy of an agent conservatively, was implemented in our framework [21]. PPO also achieves good performance in the MA domain when implemented for CTDE (multi-agent PPO, MAPPO) [22]. Hence, we applied MAPPO to CDSC, whereas the effectiveness of the PPO-based MARL framework is not well verified for our unique problem settings explained in §III.A.

Algorithm 1: Episodic Training in CDSC

```

1: Initialize policies  $\{\pi^i\}_{i=1}^N$  parametrized by  $\{\theta^i\}$ , and
   shared critic  $\hat{V}$  parameterized by  $\phi$ 
2: for episode = 1 to  $N_{\text{ep}}$  do
3:   Set rollout buffer  $B \leftarrow \emptyset$ 
4:   for rollout = 1 to  $N_\ell$  do
5:     Set rollout  $R \leftarrow \emptyset$ , and time  $t = 0$ 
6:     while  $flag = \text{True}$  do
7:       for Agent  $i = 1$  to  $N$  do
8:         if Agent  $i$  needs to take action then
9:           Make decision  $a_t^i = \pi_\theta^i(a_t^i | o_t^i)$ 
10:        end if
11:      end for
12:      if  $\{a_t^i\} \neq \emptyset$  then
13:        Execute actions  $\{a_t^i\}$ 
14:      end if
15:      Count up  $t += 1$ 
16:      for Agent  $i = 1$  to  $N$  do
17:        if Agent  $i$  took action at  $t - 1$  then
18:          Observe  $o_t^i, s_t$ , and  $r_t^i$ 
19:           $R += [o_t^i, s_t, a_t^i, i, r_t^i, o_{t+1}^i, s_{t+1}]$ 
20:        end if
21:      end for
22:      if All tasks have been completed then
23:         $flag = \text{False}$ 
24:      end if
25:    end while
26:    Compute  $\hat{V}_\tau(s_\tau)$ ,  $\{C_\tau^i\}$ , and  $\{A_\tau^i\}$  by MC or
      GAE from  $\tau = 0$  to  $t$  and add to  $R$ 
27:     $B += R$ 
28:  end for
29: end for
30: for epoch = 1 to  $N_k$  do
31:   for mini-batch  $b = 1$  to  $B$  do
32:     Make mini-batch  $b$  sampled from  $B$ 
33:     Update  $\{\theta^i\}, \phi$  with data  $b$ 
34:   end for
35: end for

```

A pseudocode for training agents using the proposed MARL framework is shown in Algorithm 1. Because industrial operations frequently involve batch processes (for instance, in logistics warehouses, shipping orders received from customers are processed hourly in batches), we assume an episodic training scheme for the proposed MARL framework. In this scheme, agent-environment interactions continue until all agents have completed all tasks (Algorithm 1, lines 22–24). To address the AMADM issue, an agent makes a decision and stores the state transition in the rollout buffer only when an action is required (lines 7–11 and 16–21). The PPO-based policy is trained conservatively by maximizing the following clipped objective function:

$$L^i(\theta^i) = \mathbb{E}_b[\min(\rho_t^i(\theta^i)A_t^i, \text{clip}(\rho_t^i(\theta^i), 1 \pm \epsilon)A_t^i)] , \quad (1)$$

where $\rho_t^i(\theta^i) = \pi_{\theta}^i(a_t^i|o_t^i)/\pi_{\theta_{\text{old}}}^i(a_t^i|o_t^i)$ is a policy ratio with the old policy prior to the update parametrized by θ_{old}^i , A_t^i is an advantage, and ϵ is the clipping range of the policy (line 33). Here, the expectation $\mathbb{E}_b[*]$ indicates the empirical average over a finite batch b of samples. The advantage can be estimated using the MC method as $A_t^i = C_t^i - \hat{V}_{\phi}(s_t, i_t)$, where $C_t = \sum_{\tau=0}^{T-t} \gamma^{\tau} r_{t+\tau}$ denotes a reward-to-go from time t to the horizon T when all agents have completed their tasks. The CDSC critic $\hat{V}$ takes as inputs not only state variables but also the ID of the agent i to identify which agent's action contributes a corresponding state transition (line 26). The critic was trained in the following supervised learning manner as $L(\phi) = \mathbb{E}_b[(\hat{V}_{\phi} - C_t)^2]$.

3) *Generalized Advantage Estimation for Asynchronous Heterogeneous Multi-agent Settings*: the feature of a CDSC framework is that each agent can be trained with a shared critic aggregating all state transitions caused by different agents in a given rollout. Thus, the installation of a shared critic brings alternative methods of advantage estimation based on a TD error. Furthermore, generalized advantage estimation (GAE), which adjusts the bias-variance tradeoff of policy gradient estimates between MC and TD methods, can be used for computing A_t^i in CDSC. Although a truncated version of GAE has already been proposed for episodic training by Schulman *et al.* [21], we need to modify it to address the AMADM issue. Because the decision making of asynchronous agents would have irregular time intervals, GAE for AMADM in a finite episode is calculated as follows:

$$A_t^i = \delta_t + (\gamma\lambda)^{\Delta t} \delta_{t+1} + \dots + (\gamma\lambda)^{\Delta(T-1)} \delta_{T-1} , \quad (2)$$

where λ denotes a bias-variance tradeoff parameter [0,1], T represents the episode length, Δt denotes a time difference between t and $t+1$, and δ_t denotes the following modified TD error:

$$\delta_t = r_t + \gamma^{\Delta t} \hat{V}(s_{t+1}, i_{t+1}) - \hat{V}(s_t, i_t) . \quad (3)$$

Compared with the original one, we can apply the proposed GAE to AMADM by simply extending the discounting coefficients $(\gamma\lambda)$ and γ in Eqs. (2) and (3) to an irregular interval setting. In addition, we can use the proposed GAE

in the case of the AMADM of heterogeneous agents because the input of the value function includes the agent ID i . We use both the MC method and GAE to estimate the advantage in CDSC to maximize performance.

C. Other MARL Frameworks for Comparison

To answer **RQ2** and **RQ3**, we further introduced three different MARL frameworks to compare their coordination performance with that of CDSC. Table I summarizes the features of critics in these frameworks, including CDSC from the viewpoint of information globalization. These frameworks assume the same functional policies $\{\pi^i(a_t^i|o_t^i)\}_{i=1}^N$ for decentralized execution. Detailed training conditions of agents in the proposed MARL frameworks are described as follows:

1) *IL with Local Reward (ILLR)*: an IL-based framework, where individual critic V^i using the local state of an agent o_t^i as input is trained with the local reward r_t^i that can be estimated by itself. Because all information required for agent training is based on local observations, ILLR is the most localized training framework. A policy trained with ILLR corresponds to a short-sighted strategy, where robots are controlled to complete their tasks at hand as quickly as possible such as first-in-first-out and greedy heuristic algorithms [23], [24].

2) *IL with Global Reward (ILGR)*: an alternative IL-based framework with the same learning structure as ILLR, but the critic is trained with the global reward r_t^g shared by all agents. Introduction of the global reward to the MAS is a typical cooperative setting in MARL [25].

3) *CTDE with Individual Critic (CDIC)*: a CTDE-based framework, where the individual critic V^i can access information about the global state s_t . All agents are also trained with r_t^g to promote coordination.

We applied independent PPO (IPPO) to ILLR and ILGR and MAPPO to CDIC, respectively [22], [26]. Compared with these three frameworks, CDSC uses the most global information (data) for agent training. By comparing the training performance of these frameworks, we can determine which global information is effective for achieving coordination in industrial settings.

IV. EVALUATION

To answer **RQs**, we applied the proposed MARL frameworks to the following automated order-picking simulator and evaluated the performance of trained policies.

TABLE I
MARL FRAMEWORKS WITH DIFFERENT CRITICS

Critic feature	MARL framework			
	ILLR	ILGC	CDIC	CDSC
State	Local		Global	
Reward	Local		Global	
Architecture	Individual (N)			Shared (1)
State value	Localized		$\longleftrightarrow$	Globalized

A. Simulation Environment

1) *Fully Automated Order Picking*: the simulation of an automated order picking environment (AOPE) in a logistics warehouse was performed for quantitative evaluation of the proposed method. The overview of the AOPE is illustrated on the left side of Fig. 1. Ordered items are arranged from left to right in the figure. The AOPE consists of four types of MHE whose controller makes decisions for task selection, as shown on the right side of Fig. 1. The first one is a flow rack (FR) in which sorting boxes are allocated. Each sorting box corresponds to a shipping destination. When all ordered items are sorted into a box, the box is transported to the next working area, and FR replaces it with a new one. The FR controller selects a shipping box to start order picking tasks for its required items. The second one is a set of parallel conveyors (PC) that transports items from the inventory area to allocated shipping boxes in FR. There are three conveyors in parallel, and items are loaded from the inventory area by type. Each conveyor has six loading ports. The PC controller selects an item type to be loaded on the conveyor from the inventory area. The third one is a picking robot (PR1) that picks up items from the PC and places them on a carousel conveyor one by one. PR1 can also move among the conveyors to pick items from each conveyor. The PR1 controller selects a conveyor to pick items: conveyor selection occurs whenever the picking of the same item type is completed. The last one is another picking robot (PR2) that picks up items from a carousel conveyor and sorts them into shipping boxes one by one. The PR2 controller selects an allocated shipping box to sort items from the carousel conveyor. To prevent the simulation from being fixed to one work scenario, the item loading and replacement times of PCs and the picking time of both PR1 and PR2 are given normal distribution variations. Similar configurations were designed as a type of parts-to-picker systems [27], [28], whereas the AOPE was simplified to complete the simulation quickly as a training environment.

2) *Picking Order Data*: we evaluated the performance of the proposed MARL frameworks with two different picking order datasets in warehouse operations. The characteristics of these datasets are summarized in Table II. The “Orders” in the table shows the number of unique sets of item types (“Types”) and shipping boxes (“Shippings”). In logistics warehouses, items are typically stored by type [7]. Thus, the more item types in a picking order, the more item transfers

TABLE II
PICKING ORDER DATA

Picking order	Orders	Items	Types	Shippings
Low Mixed (LM)	179	201	16	42
High Mixed (HM)	186	200	95	41

there are in the upstream process, and the more difficult it is to optimize operations. To investigate the performance of our proposed algorithm for different picking orders, the item types are different, but items and shipping boxes are nearly the same. These picking orders have reasonable characteristics compared with previous studies in terms of the number of orders per type [27], [29].

B. RL Settings

1) *Training Condition*: the proposed MARL frameworks were applied to train the four process controllers in an AOPE. Each process controller was regarded as an RL agent and trained in the advantage AC manner [30]. Table III summarizes the hyperparameters used in this evaluation.

2) *Agent States*: the states of all agents are summarized in Table IV. The number in brackets represents the number of states multiplied by the number of components. As described in §III, all agents were assumed to make decisions in a decentralized manner.

3) *Action Mask*: we introduced an action mask to eliminate invalid actions at every decision making [32]. The masked stochastic policy $\hat{\pi}_{\theta}^i$ computes the probability of a valid action $a_t^{i,k}$ as follows:

$$\hat{\pi}_{\theta}^i(a_t^{i,k}|o_t^i) = \pi_{\theta}^i(a_t^{i,k}|o_t^i) / \sum_{a_t^{i,j} \in \mathcal{A}^i} m(a_t^{i,j}) \pi_{\theta}^i(a_t^{i,j}|o_t^i), \quad (4)$$

where $m(a_t^{i,j})$ denotes the action mask of an agent i that outputs 0/1 when the j -th action is valid/invalid at t . The action mask reduces $\rho_t^i(\theta^i)$ and stabilize gradient updates.

4) *Reward Design*: the objective of agents in AOPE was to minimize the makespan T_c of all order picking tasks. Thus, the local reward r_t^i for ILLR was set to the negative value of the elapsed time until an agent selects the next task: the RL

TABLE III
HYPERPARAMETERS IN MARL SETTINGS

Setting	Hyperparameter	Value
IPPO, MAPPO	Clipping range ϵ	0.2
	Discount factor γ	0.99
	Scaling factor ζ	800
	Rollouts N_{ℓ}	64
	Epochs N_k	5
	Episodes N_{ep}	5000
Network	Minibatch size	64
	Network	MLP
	Hidden layers	2
	Hidden units	128 /layer
Optimizer	Activation	Tanh()
	Optimizer	Adam [31]
	Learning rates	0.001(θ), 0.0003(ϕ)
	Decay rate	0.8/250 episodes
Reward	Initialization	Orthogonal
	t_{ofs}	6.6(LM), 6.4(HM)

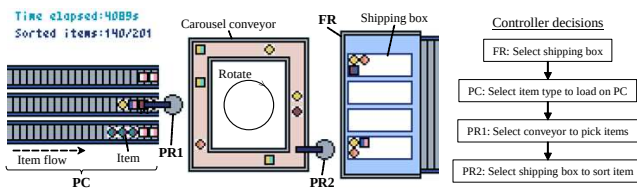


Fig. 1. Simulation environment. Left side: overview of AOPE composed of four automated processes (FR, PC, PR1, and PR2). Items with different shapes (square and circle) and colors denote different item types. Right side: decisions taken by the controller of each process.

TABLE IV
STATES OF AGENTS

Agent	States
FR	Numbers of unsorted items and types in allocated shipping boxes (2), numbers of items and types waiting to be loaded on PC in allocated shipping boxes (2), number of unallocated shipping boxes (1), and numbers of items and types in unallocated shipping boxes (2)
PC	ID of conveyor onto which items are loaded (1), numbers of loaded items (1×3), locations of the first and last loaded items (2×3), numbers of items and types waiting to be loaded at loading ports (2×6), and numbers of items loading onto conveyors and location of work-in-progress loading ports (2×3)
PR1	Location and direction of PR1 (2), numbers of items on carousel conveyor (1), numbers of items on PC (1×3), locations of the first items on PC and the numbers of items whose types are the same as the first ones (2×3), location of the last item on PC (1×3), numbers of items waiting to be loaded at loading ports (1×6), and numbers of items loading into conveyors and location of work-in-progress loading ports (2×3)
PR2	Location and direction of PR2 (2), number of sorted items (1), array of items on carousel conveyor whether they can or cannot be sorted into shipping boxes (28 × 4), and number of unsorted items of shipping boxes (1×4)
Common	Time-step (1) and action mask (number of actions)

policy was expected to learn to complete the current task as quickly as possible. Agents in ILLR received r_t^i as an immediate reward for every decision they made. In contrast, agents in the other three MARL frameworks were trained in the LTESR setting: the global reward was only given at the terminal state transition. The terminal global reward r_{tml}^g was estimated as follows:

$$r_{tml}^g = \begin{cases} 2 \cdot [(t_c^{ks} - t_{ofs})^4 - 1] & \text{if } t_c \geq t_{ofs} \\ -2 \cdot [(t_c^{ks} - t_{ofs})^4 + 1] & \text{otherwise} \end{cases}, \quad (5)$$

where t_c^{ks} denotes the ksec-unit T_c when the last item is sorted into the last shipping box, and t_{ofs} denotes an offset depending on the picking order dataset. Because t_c represents a ksec-order value and the AOPE was computed at 10 fps, agents exhibit a long trajectory through the simulation. Therefore, if a well-known $\gamma \sim 0.99$ is used in the training, the impact of r_{tml}^g on C_t will be significantly decayed because of the long trajectory. To propagate r_{tml}^g through the trajectory, we introduced a scaling factor ζ listed in Table III into the discounting coefficients in Eqs. (2) and (3) as $(\gamma\lambda)^{\Delta t/\zeta}$ and $\gamma^{\Delta t/\zeta}$.

C. Rule-based Control

To compare the performance of MARL-based control, we established a set of control rules for each process controller. Control features in AOPE are summarized as follows:

- Decentralized control decisions with hierarchical structure (FR→PC→PR1→PR2).
- Task execution can be parallelized in a process (processes in FR and PC).
- Each process has different task granularities (shipping destinations for FR, item types for PC, and items for PR1 and PR2).

TABLE V
CONTROL RULES OF FOUR PROCESSES

Process	Control rules for task selection
FR	A shipping box with the most/fewest items (2) A shipping box with the most/fewest item types (2) A shipping box selected by seed algorithm (4)
PC	An item type with the most/fewest items, or farthest/closest to PR1 (8)
PR1	A conveyor loading a head item type with the most/fewest items, or farthest/closest to PR1 (8)
PR2	A shipping box with the most/fewest unsorted items, or farthest/closest to PR2 (8)

- Each process contains stochastic uncertainties (e.g., picking speeds of PR1 and PR2).

We have found that there is no prior approach using mathematical optimizations, metaheuristics, or artificial intelligence that can comprehensively handle the above four features even though recent studies have covered some of them [28], [33]–[35]. Thus, the rule-based control, still actively researched for warehouse optimization, was used for our evaluation [36]. The sets of control rules of all processes are summarized in Table V, where a number in parentheses indicates the number of control rules for each process. We applied a seed algorithm to four of the eight rules of FR [37]. The seed algorithm can select a shipping box with the best score obtained by comparing items in candidate and allocated boxes. In this evaluation, we estimated the score of shipping boxes as the similarity of allocated boxes. The similarity was calculated by accumulating weights (w_s, w_d), where w_s is added to the score if one item type in the candidate box is required for the allocated boxes, whereas w_d is added if the item type is excluded from the allocated boxes. To expand the choices of shipping boxes, we established seed algorithm-based rules by changing weights, as $(w_s, w_d) = (1, 0), (0, 1), (1, -1),$ and $(-1, 1)$.

D. Results and Discussions

1) *Improving Training Performance with GAE in CDSC:* Figure 2 (a) shows training curves of CDSC with different advantage estimations in AOPE with LM and HM picking orders. Each curve shows averaged performance with a standard deviation over three random seeds. TD(0) has the worst performance among all methods, suggesting the difficulty of applying the TD method in LTESR as explained in §III.B.2. Compared with the MC method, GAE accelerated the training and achieved significantly better performance in the early phase. This performance superiority was maintained until the end of training for $\lambda = 0.5 \sim 0.95$. Hence, the proposed GAE is an effective advantage estimation for CDSC in AOPE.

2) *Comparison of Different Control Methods:* we summarize the training results of four MARL frameworks in Fig. 2 (b) and simulation results of AOPE obtained from different control methods in Table VI. Each value shows an average and standard deviation of T_c sampled from 192 rollouts. As the baseline of each simulation, we added the results, where all controllers selected their tasks at random,

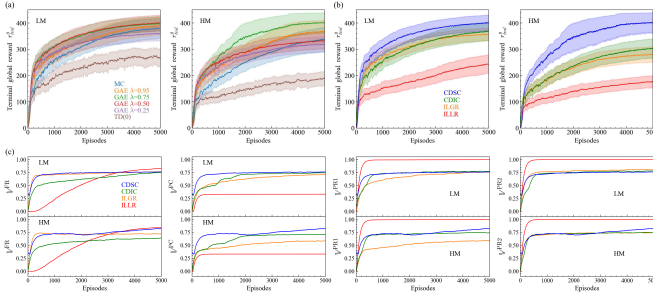


Fig. 2. Training results of proposed MARL frameworks in APOE with LM and HM picking orders. (a) learning curves of CDSC with different advantage estimations. (b) Comparison of learning curves among four MARL frameworks. (c) Comparison of explained variances of each APOE agent among four MARL frameworks.

as “Random choice.” The performance of rule-based control represents the result of the best combination rules (4096 in total) listed in Table V. ILLR, the most localized MARL framework, achieved comparative performance to the rule-based control for the LM picking order, outperforming the HM one. Because these methods control robots with a short-sighted plan to complete their tasks as hand as quickly as possible, the advantage of ILLR over rule-based control for the HM picking order can be attributed to the flexibility in task selection. The ILLR-trained policy can flexibly select a task depending on the input o_t^i reflecting the operation status compared with the rule-based policy whose selection is solely based on the implemented rule. Such flexibility is more effective for the HM picking order whose task selection in PCs is more frequent due to the large variety of item types. Thus, the MARL-based control can provide more efficient operations in industrial automated systems than the rule-based control, even with localized training.

We answer the research questions as follows.

RQ1: as shown in Table VI, CDSC-based control with GAE ($\lambda = 0.5, 0.75$ for LM, HM picking orders) achieved the shortest makespans among all methods. Thus, the MARL agents achieve coordination by introducing both the globalization of training information and the unification of state transitions to the shared critic.

RQ2: the most effective globalization from ILLR is the reward setting caused by ILGR, where agents share the global reward. Figure 2 (c) shows smoothed explained variance of state values averaged over different trials computed as

TABLE VI

COMPARISON OF MAKESPANS AMONG DIFFERENT CONTROL METHODS

Control method		Picking order	
		LM	HM
Random choice		5385.0 $\pm$ 188.9 s	5227.8 $\pm$ 180.6 s
Rule-based		3273.4 $\pm$ 110.6 s	3599.5 $\pm$ 120.1 s
MARL	ILLR	3278.2 $\pm$ 120.2 s	3334.2 $\pm$ 103.9 s
	ILGR	2941.9 $\pm$ 62.9 s	2975.9 $\pm$ 90.8 s
	CDIC	2914.9 $\pm$ 87.5 s	2891.3 $\pm$ 102.8 s
	CDSC (MC)	2884.7 $\pm$ 94.9 s	2793.1 $\pm$ 123.2 s
	CDSC (GAE)	2834.3$\pm$69.4 s	2638.2$\pm$101.3 s

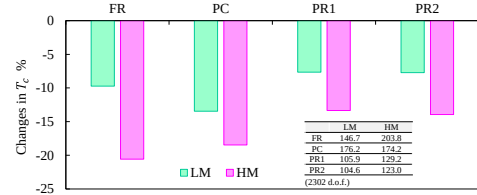


Fig. 3. Percentage change in makespan when switching from CDSC-trained to ILLR-trained policies for each agent. Statistics of Welch’s t -test between CDSC and switched results are listed in the inserted table: all corresponding p -values satisfy $p < 0.001$ (95% confidence interval).

$\nabla^i = 1 - \text{Var}(V_{\text{exp}}^i - V_{\phi}^i) / \text{Var}(V_{\phi}^i)$, where V_{exp}^i denotes a true state value obtained from experiments, and V_{ϕ}^i denotes the prediction by the critic. ∇^{PR_1} and ∇^{PR_2} of ILLR rapidly converged to 1 and almost fully predicted the actual state values compared with the other three frameworks, whereas ∇^{FR} and ∇^{PC} of ILLR yield poor prediction. This result suggests that ILLR agents were trained so locally that the agents in downstream processes could easily infer individual environmental changes, whereas ones in upstream processes, significantly affected by downstream performance, could not achieve reasonable prediction. This localized training tendency was significantly eliminated by introducing the global rewards: ∇^{FR} and ∇^{PC} were significantly improved by ILGR in exchange for a slight decrease in ∇^{PR_1} and ∇^{PR_2} . The relatively marginal contribution of state globalization via CDIC may be attributed to the difficulty in critic training due to the increase in the number of input states, as listed in Table IV. In support of this consideration, CDSC with the MC method and the same advantage estimation outperformed CDIC. Although the representation ability of CDIC individual critics was lost, this drawback may have been resolved by a CDSC single critic. Furthermore, the performance of CDSC was improved by the proposed GAE, and thus, it achieved the shortest T_c among all control methods.

RQ3: we evaluated performance degradation caused by changing the policy for each process from CDSC to ILLR. Figure 3 shows the percentage change in the makespan when switching from CDSC-trained to ILLR-trained policies for each agent. The performance of CDSC was degraded even if its downstream policies (namely, PR1 and PR2) were replaced with ILLR-trained policies. The performance degradation in downstream processes suggests the coordination throughout the overall processes, and such overall coordination may be the first benefit of overall optimization via CDSC. Furthermore, the ILLR-trained policy of FR (and PC) results in significant performance degradation. This trend can be seen more prominently with the HM picking order, where PCs make decisions more frequently due to many item types. Such results are consistent with the comparison results of explained variances, where ILLR causes poor state value predictions in upstream processes, as described above. The upstream processes make decisions more sparsely because their task granularity is bulkier, such as shipping box (FR) and item type (PC). This sparse decision making is more susceptible to environmental changes caused by the

downstream processes; thus, the ILLR framework fails to optimize the control of operations. Hence, the second benefit of overall optimization in warehouse automation via CDSC may be the improvement in efficiency in upstream processes.

V. CONCLUSION

To clarify the impact of overall optimization on industrial automation, we explored an efficient MARL framework that enables practical robot coordination. In the proposed framework, agents were trained in CDSC, a CTDE manner using both globalized rewards and single shared critic. Furthermore, we proposed the modified GAE for policy update to improve the performance of CDSC to address the two major issues in typical industrial settings: LTESR and AMADM. The evaluation results show that the CDSC-based control applied to task selections of MHE in AOE can achieve the shortest makespan compared with other MARL frameworks and rule-based controls. The results also suggest that the overall optimization has the following advantages for warehouse automation: bottom-up efficiency through process coordination and further efficiency of upstream process control. Although the obtained knowledge is limited to our experimental design, we believe that we can qualitatively and quantitatively clarify the impact of overall optimization on various types of automated systems with different layouts and configurations by introducing our proposed MARL frameworks for future work.

REFERENCES

- [1] L. Custodio and R. Machado, "Flexible automated warehouse: a literature review and an innovative framework," *Int. J. Adv. Manuf. Technol.*, vol. 106, no. 1, pp. 533–558, 2020.
- [2] "Welcome to the presentation of World Robotics 2021," presented at the World Robotics. International Federation of Robotics, 2021. [Online]. Available: https://ifr.org/downloads/press2018/2021-10-28-WR_PK_Presentation_long_version.pdf
- [3] K. Azadeh, R. De Koster, and D. Roy, "Robotized and automated warehouse systems: Review and recent developments," *Transp. Sci.*, vol. 53, no. 4, pp. 917–945, 2019.
- [4] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, 2020.
- [5] NVIDIA, "Develop on NVIDIA Omniverse," <https://developer.nvidia.com/nvidia-omniverse-platform>, accessed: 2023-02-20.
- [6] P. R. Wurman, R. D'Andrea, and M. Mountz, "Coordinating hundreds of cooperative, autonomous vehicles in warehouses," *AI Mag.*, vol. 29, no. 1, pp. 9–20, Mar. 2008.
- [7] J. J. Bartholdi and S. T. Hackman, *Warehouse & Distribution Science*, Release 0.98.1. Atlanta, GA: Georgia Tech., 2017. [Online]. Available: <https://www.warehouse-science.com/book/editions/wh-sci-0.98.1.pdf>
- [8] Y. Jaghbeer, R. Hanson, and M. I. Johansson, "Automated order picking systems and the links between design and performance: a systematic literature review," *Int. J. Prod. Res.*, vol. 58, no. 15, pp. 4489–4505, 2020.
- [9] L. Zhen and H. Li, "A literature review of smart warehouse operations management," *Front. Eng. Manag.*, pp. 1–25, 2022.
- [10] M. Tan, "Multi-agent reinforcement learning: Independent vs. cooperative agents," in *Proc. of the 10th ICML*, 1993, pp. 330–337.
- [11] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in *Proc. of the 31st Int. Conf. on NeurIPS*, 2017, p. 6382–6393.
- [12] T. Pulikottil, L. A. Estrada-Jimenez, H. U. Rehman, J. Barata, S. Nikghadam-Hojjati, and L. Zarzycki, "Multi-agent based manufacturing: current trends and challenges," in *Proc. of the 26th IEEE Int. Conf. on ETFA*, 2021, pp. 1–7.
- [13] L. Canese, G. C. Cardarilli, L. Di Nunzio, R. Fazzolari, D. Giardino, M. Re, and S. Spanò, "Multi-agent reinforcement learning: A review of challenges and applications," *Appl. Sci.*, vol. 11, no. 11, p. 4948, 2021.
- [14] I. Bello, H. Pham, Q. V. Le, M. Norouzi, and S. Bengio, "Neural combinatorial optimization with reinforcement learning," *arXiv preprint 1611.09940*, 2016.
- [15] G. Shen, R. Ma, Z. Tang, and L. Chang, "A deep reinforcement learning algorithm for warehousing multi-agv path planning," in *Proc. of the Int. Conf. on NetCIT*. IEEE, 2021, pp. 421–429.
- [16] M. Li, B. Guo, J. Zhang, J. Liu, S. Liu, Z. Yu, Z. Li, and L. Xiang, "Decentralized multi-agv task allocation based on multi-agent reinforcement learning with information potential field rewards," in *Proc. of the IEEE 18th Int. Conf. on MASS*, 2021, pp. 482–489.
- [17] Y. Xiao, J. Hoffman, T. Xia, and C. Amato, "Learning multi-robot decentralized macro-action-based policies via a centralized q-net," in *Proc. of the IEEE Int. Conf. on ICRA*, 2020, pp. 10 695–10 701.
- [18] J. Wang and L. Sun, "Reducing bus bunching with asynchronous multi-agent reinforcement learning," *arXiv preprint 2105.00376*, 2021.
- [19] Y. Xiao, W. Tan, and C. Amato, "Asynchronous actor-critic for multi-agent reinforcement learning," *arXiv preprint 2209.10113*, 2022.
- [20] F. A. Oliehoek and C. Amato, *A concise introduction to decentralized POMDPs*. Springer, 2016.
- [21] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint 1707.06347*, 2017.
- [22] C. Yu, A. Velu, E. Vinitzky, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of ppo in cooperative multi-agent games," *arXiv preprint 2105.00376*, 2021.
- [23] N. Ascheuer, M. Grötschel, and A. Abdel-Aziz Abdel-Hamid, "Order picking in an automatic warehouse: Solving online asymmetric tsp," *Math. Oper. Res.*, vol. 49, pp. 501–515, 1999.
- [24] J. M. Framinan and R. Leisten, "Total tardiness minimization in permutation flow shops: a simple approach based on a variable greedy algorithm," *Int. J. Prod. Res.*, vol. 46, no. 22, pp. 6479–6498, 2008.
- [25] A. OroojlooyJadid and D. Hajinezhad, "A review of cooperative multi-agent deep reinforcement learning," *arXiv preprint 1908.03963*, 2019.
- [26] C. S. de Witt, T. Gupta, D. Makoviichuk, V. Makoviychuk, P. H. Torr, M. Sun, and S. Whiteson, "Is independent learning all you need in the starcraft multi-agent challenge?" *arXiv preprint 2011.09533*, 2020.
- [27] L. Xie, N. Thieme, R. Krenzler, and H. Li, "Introducing split orders and optimizing operational policies in robotic mobile fulfillment systems," *Eur. J. Oper. Res.*, vol. 288, no. 1, pp. 80–97, 2021.
- [28] I. Suemitsu, H. K. Bhamgara, K. Utsugi, J. Hashizume, and K. Ito, "Fast simulation-based order sequence optimization assisted by pre-trained bayesian recurrent neural network," *IEEE Robot. Autom. Lett.*, vol. 7, no. 3, pp. 7818–7825, 2022.
- [29] D. Füllner and N. Boysen, "High-performance order processing in picking workstations," *EURO J. Transp. Logist.*, vol. 8, no. 1, pp. 65–90, 2019.
- [30] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *Proc. of the 33rd ICML*, 2016, pp. 1928–1937.
- [31] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. of the 3rd Int. Conf. on Learning Representations*, 2015.
- [32] S. Huang and S. Ontañón, "A closer look at invalid action masking in policy gradient algorithms," *arXiv preprint 2006.14171*, 2020.
- [33] S. A. B. Rasmi, Y. Wang, and H. Charkhgard, "Wave order picking under the mixed-shelves storage strategy: A solution method and advantages," *Comput. Oper. Res.*, vol. 137, p. 105556, 2022.
- [34] M. Kordos, J. Boryczko, M. Blachnik, and S. Golak, "Optimization of warehouse operations with genetic algorithms," *Appl. Sci.*, vol. 10, no. 14, p. 4817, 2020.
- [35] J. Park, J. Chun, S. H. Kim, Y. Kim, and J. Park, "Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning," *Int. J. Prod. Res.*, vol. 59, no. 11, pp. 3360–3377, 2021.
- [36] Y. A. Bozer and C. Eamrungrroj, "Throughput analysis of multi-device trip-based material handling systems operating under the modified-fcfs dispatching rule," *Int. J. Prod. Res.*, vol. 56, no. 4, pp. 1486–1503, 2018.
- [37] E. Elsayed, "Algorithms for optimal material handling in automatic warehousing systems," *Int. J. Prod. Res.*, vol. 19, no. 5, pp. 525–535, 1981.