Knowledge-aware and Learning-focused Multi-Objective Multi-agent Reinforcement Learning for Maintenance Technician Assignment

Adrien Bolling[a,*,1] **and Sylvain Kubler**[a]

[a]SnT, University of Luxembourg
ORCID (Adrien Bolling): https://orcid.org/0009-0008-6766-4350, ORCID (Sylvain Kubler): https://orcid.org/0000-0001-7672-7837

Abstract. The allocation of tasks to operators with different skill levels is crucial in the manufacturing industry, which is known as Human Resources Assignment Problem (HRAP). In the literature, HRAP is usually solved through linear programming and meta-heuristics methods. However these methods face limitations in their ability to take into account the human factor in a more complex manner (e.g., uncertainty of efficiency at a given time, uncertainty of availability at a given time, personal preferences, ability to learn and retain information, and to be able to use it when needed). To efficiently handle this complex nature of the problem, this paper introduces an innovative approach that leverages Multi-Objective Multi-Agent Reinforcement Learning (MOMARL) to optimize HRAP in the context of maintenance activities/tasks on production lines. To the best of our knowledge, this is the first paper focusing on modeling a technician's knowledge as part of a MOMARL framework, along with its impact on HRAP.

1 Introduction

As time goes by, digital technologies are becoming more widespread in the workplace. However, at the shop-floor level, this presents challenges not only due to technological advancements (upgrading the factory with new technologies), but also due to organizational difficulties (altering the way operators perform their daily tasks). Although it is widely recognized that the recent technological advancements (incl., Cloud, Internet of Things, Artificial Intelligence, etc.) have driven efficiency and productivity, recent studies suggest that they also contribute to a growing divide among workers due to various side effects[13][12][14][7].

At the forefront of these issues is the rise of routine tasks and the lack of emphasis on training low-level workers. The contrast in regards to skill is becoming more evident, as most current digital assistance solutions are only training employees that are already skilled enough to be up-skilled, at the cost of the others. This growing rift highlights a critical challenge in the Human Resources Assignment Problem (HRAP), where the optimal allocation of human resources to various tasks must be balanced against human development concerns, such as mental health or continuous progression. Traditional HRAP solutions, primarily utilizing operations research techniques like linear programming and meta-heuristics, often require simplifications that fail to capture the dynamic and stochastic nature of real-world environments, particularly those involving complex human factors.

To address this trend and reduce the skills gap within the HRAP field, we propose a novel approach using Multi-Objective Multi-Agent Reinforcement Learning (MOMARL). Our approach aims to optimize maintenance technician assignment in a manufacturing context by focusing not only on task allocation but also modeling the learning processes through a novel representation of knowledge and its propagation among technicians and tasks. This ensures a more equitable and realistic assignment process, where all employees have the opportunity to develop and use their skills.

This paper is structured as follows : we first provide a background on Multi-Objective Multi-Agent Reinforcement Learning, detailing its relevance and application to HRAP. Following this, we describe our modeling of technicians and tasks, including the introduction of knowledge grids based on the task's representation, and the corresponding propagation mechanisms. We then present our experimental setup and results, demonstrating the effectiveness of our MOMARL-based framework. Finally, we discuss related work and conclude with insights and future research directions.

By integrating MOMARL into HRAP, our aim is to close the gap between traditional optimization methods and the complex and dynamic requirements of modern human resource management, ultimately contributing to more efficient, humane, and intelligent human resource allocation in manufacturing settings.

2 Background

2.1 Multi-Objective Multi-agent Reinforcement Learning

A Multi-Objective Multi-Agent context to an optimization problem such as this one can be modeled as a Multi-Objective Stochastic Game (MOSG) [10].
A MOSG is formally defined as a tuple $M = (S, \mathcal{A}, T, \mathcal{R})$, with $n_T \geq 2$ agents and $d \geq 2$ objectives, where :

- S : state space

* Corresponding Author. Email: adrien.bolling@uni.lu.
[1] Code publicly available at https://github.com/AdrienBolling/technician-assignment

- $\mathcal{A} = A_1 \times ... \times A_n$ a set of joint actions, A_i is the action set of agent i
- $\mathcal{T} : S \times \mathcal{A} \times S \to [0, 1]$ the probabilistic transition function
- $\mathcal{R} = R_1 \times ... \times R_n$ reward functions, $R_j : S \times \mathcal{A} \times S \to \mathbb{R}^d$ the vectorial reward function of agent j for each of the d objectives.

In our specific case, a MOSG will be sufficient, although a more realistic approach could consider using a multi-objective partially observable stochastic game (MOPOSG) where agents do not have access to the full state of the environment.

Each agent will learn a policy $\pi_i : S \times A_i \to [0, 1]$, maximizing the expected discounted long-term reward.

Here we chose an ESR approach. Despite the long-term nature of our environment, which allows us to evaluate the agent's utility over multiple executions, which could have motivated the choice of SER [10], promoting simplicity of computation in the beginning, and promoting the agent's cooperation through simpler rewards seemed more important.

2.2 *Human resources allocation problem*

The original Allocation Problem (AP) is one of the first studied combinatorial problems. It consists of assigning limited resources to a set of tasks with the aim of optimizing one or more objectives, under the resources constraints. The relevance of APs comes from its many applications, such as, but not limited to, the dispatching of orders or supplies along a production line [5], but also outside of manufacturing, for example, in healthcare, project management or review systems. The Human Resources Allocation Problem (HRAP) is simply an extension of the AP where the resources subject to constraints and allocation are people [1]. However, it is crucial to properly consider the complexity added by the human factor [3].

Traditionally, HRAP has been addressed using operations research techniques, such as linear programming, Hungarian methods, genetic algorithms, and ant colonies to list a few [1]. However, these methods often require simplifications and assumptions that may not hold in dynamic, complex, and stochastic environments. Such difficulties being a given considering the addition of the human nature and considerations to the well-known AP problem, it is only natural for more sophisticated approaches to emerge with the advent of machine learning. Multi-agent systems, among them multi-agent reinforcement learning (MARL), seem to offer a promising framework for HR allocation, where each employee can be modeled as an agent with its own characteristics (preferences, skills, goals, limitations).

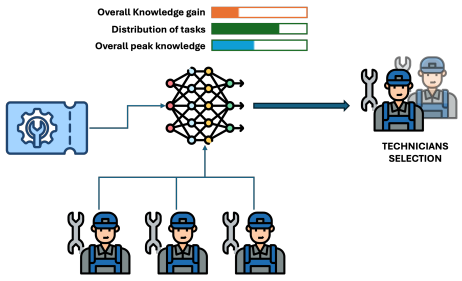

Figure 1. Schema of the model

Table 1. Some mathematical notations that will be used

Notation	Description
T	Technician set
Tk	Ticket set
T_j	Technician j
G_j	Knowledge grid of T_j
h_j^l	Tickets treated over horizon l by T_j
lr	Learning rate of a technician
tk_i	Maintenance ticket i
e_i^{tk}	Embedding of ticket i
e_j^T	Embedding of technician j
k_{T_j,tk_i}	Knowledge of T_j for tk_i
σ_p	Propagation parameter
τ	Transmission parameter
n	Dimension of a ticket embedding
n_T	number of technicians
$hte^l_{T_j}$	Embedding of past l tk treated by T_j

3 Modelisation

3.1 *Environment*

3.1.1 *Technicians*

The goal of an algorithm aiming to solve an HRAP instance, is to allocate humans, here some maintenance technicians. Let $T_j \in T$ be the j-th technician in our system. it is defined by the following attributes : a learning rate lr_j representing his ability to learn, a knowledge grid G_j representing his global knowledge about machines and maintenance activities, and a history h_j^l a value representing the proportion of tickets T_j has treated over the horizon l.

The technicians need to be embedded to be effectively used in a neural network. In order to do that, we create an embedded vector in two parts :

$$e_j^T = features\_embedding + hte^l_{T_j} \quad (1)$$

where $hte^l_{T_j}$ is the output of a Gated Recurrent Unit (GRU) cell, which aims to capture the history of the tickets previously treated by the technician. This approach has previously been used successfully to model tasks previously completed by an agent [16].

3.1.2 *Tickets*

The tickets here function similarly to tasks in HRAP. Each ticket $tk_i \in Tk$ does not need precise design specifications because they will be processed by a ticket feature extractor. This extractor retrieves relevant information from the tickets and returns them as an embedding e_i^{tk} of dimension n, which will then be the only thing used in the rest of the loop. This allows us to free ourselves from some constraints of the current literature when it comes to knowledge representation in manufacturing, oftentimes limited to a small set of categories.

In the context of this paper, we are generating some synthetic random 2-d embedding from normal distributions. This decision has been taken in an attempt to simplify a first implementation of this idea. It is also based on some principal components analysis done on real manufacturing maintenance data from a partner company, which was then embedded through an S-BERT model.

3.1.3 *Knowledge Grids*

A knowledge grid is a n-dimensional representation of the knowledge of a technician. It is modeled as a n-d array, where each dimension is a dimension in the embedding e_i^{tk} of the maintenance ticket tk_i. Thus $G_k(e_i^{tk})$ represents the knowledge of T_j for a ticket tk_i, which will be called k_{T_j,tk_i}.

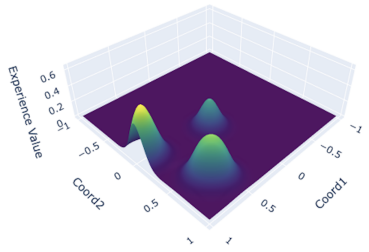

Figure 2. Example of an initial experience grid

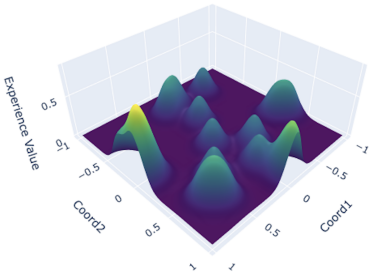

Figure 3. Same experience grid after 50 steps of the environment with random actions taken

The experience grids displayed in Figure (2) and Figure (3) represent the state of the knowledge of an agent of this environment. Figure (2) is this state at initialisation time, and Figure (3) after 50 steps.
The axis *Coord1* and *Coord2* correspond to the two dimensions of the tickets embeddings in this environment. The third dimension *Experience Value* represents the knowledge of a technician at a certain point in the embedding space.

3.1.4 *Knowledge*

The knowledge k_{T_k,tk_i} of a technician at a given location of the embedding space is logarithmic with respect to the number of experience the technician has for this type of ticket. However there are 3 mechanisms that are used to increase this knowledge which are applied in order :

1. **New experience** : we assume that the previous knowledge can be expressed as [6]

$$k^{prev} = \log(1 + \text{previous number of experiences}) \quad (2)$$

Thus we compute the following :

$$\Delta k = \log(1 + \frac{1}{\exp k^{prev}}) \times lr_j \quad (3)$$

2. **Knowledge transfer** : considering maintenance expert's knowledge, we assume that a technician T_a may transfer part of his knowledge to another technician T_b through supervision, a mechanism also mentioned by [6]. We represent it as a weighted average between the two knowledge, the one assigned and the one supervising. This average is weighted by a parameter τ representing how much of the information is transferred.

$$\Delta k_{T_b} = (k_{T_b}^{prev} + \Delta k_{T_b})(1-\tau) + k_{T_a}^{prev} \times \tau - k_{T_b}^{prev} \quad (4)$$

3. **Knowledge propagation** : having discussed it with maintenance experts, we assume that knowledge about a certain maintenance operation can be transferred to neighboring operations. This could be translated in several ways : knowledge about a machine, about the brand, about the type of failure. To represent this behavior, each increase in a technician's knowledge grid will be propagated in a neighborhood using a Gaussian kernel K with the following

$$\text{Let } G^{\Delta}(x) = \begin{cases} \Delta k & \text{if } x = e_i^{tk} \text{, the ticket treated} \\ 0 & \text{else} \end{cases} \quad (5)$$

and

$$\text{Let } s = \frac{k^{incr}}{convolve(G^{\Delta}, K)} \quad (6)$$

in

$$G^{new} = G^{old} + convolve(G^{\Delta}, K) \times s \quad (7)$$

3.1.5 *Environment loop*

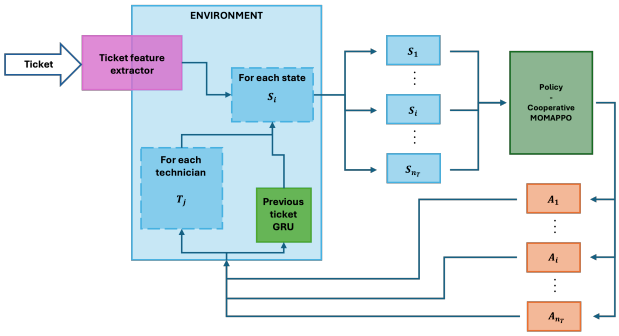

Figure 4. Schema of the environment loop

The Figure (4) describes the behavior and interactions of our environment for easier understanding.

3.2 *State*

The state space S is defined as a collection of individual spaces S_j, one per agent. However, we are still in a fully observable setting, as these spaces contain the same information, just in a different order.

$$S_j = (e_j^T, e_i^{tk}, e_1^T, ..., e_{n_T}^T) \quad (8)$$

In particular, each technician will have its own features placed at the beginning of the vector.

3.3 Actions

There are three types of actions an agent can take, with some constraints. Each agent will simultaneously pick an action among those three.

$$\mathcal{A} = A_1 \times ... \times A_{n_T} \text{ , with } A_j \subseteq \{assign, supervise, nothing\}$$

- **Assign** : an agent that chooses this action will choose to treat the current ticket, exactly one agent has to take this action.
- **Supervise** : an agent that chooses this action will choose to supervise the agent treating the current ticket, at most one agent can take this action
- **Nothing** : an agent that chooses this action will do nothing, all the remaining agents should pick this action

3.4 Rewards

In the following, let $\mathcal{D}_G = \mathcal{D}_{e_1} \times ... \times \mathcal{D}_{e_n}$ where D_{e_i} is the set for the i-th dimension of a ticket embedding. To promote collaboration between our agents, especially regarding the supervision action, the rewards are computed as a team. That is to say, each reward is an aggregate (here an average) of what can be considered the "individual" rewards.
Let's consider the following **individual** rewards :

- **Total knowledge** : representing the objective of having technicians that become as knowledgeable as possible, it is defined as the hypervolume of the knowledge grid and expressed as such

$$r_{T_j}^{tot} = \int_{\mathcal{D}_G} G_{T_j}(x)dx \tag{9}$$

  in practice, this integral translates to nested sums as each dimension of our knowledge grids is discrete for ease of computations purposes.
- **Proportion of tickets treated** : representing the objective of not focusing solely on one technician, and having an appropriate distribution of tasks. With $h_j^l = [0, 1, ..., 1]$ a vector of size l where $h_j^l(t) = 0$ means the $(l-t)$-th last ticket has not been treated by T_j (and 1 means that it has), and

$$p_j = \frac{1}{l}\sum_i h_j^l(i) \tag{10}$$

  we finally have

$$r_{T_j}^p = \exp^{\frac{(p_j - p_{ideal})^2}{2\sigma_p^2}} \text{ , with } p_{ideal} = \frac{1}{n_T} \tag{11}$$

  with the goal of pushing the agents to handle each the same number of tickets, by punishing both under and over performance.
- **Highest knowledge** : representing the objective to get some degree of specialization in an agent. although it may seem counter-intuitive at first, this objective is actually in conflict with $r_{T_j}^{tot}$

$$r_{T_j}^{hk} = \max_{x \in \mathcal{D}_G} G_{T_j}(x) \tag{12}$$

In the following, let's assume

$$R_1 = r_{T_j}^{tot} \tag{13}$$

4 Experiments and Results

The experiments have been run using our own implementation of a technician assignment environment, and a slightly modified version of the Cooperative MOMAPPO delivered with the MOMALand library, both publicly available on GitHub, with the following key hyperparameters.

Table 2. Training hyperparameters - MOMAPPO and Environment

Hyperparameter	Value
num_weights	500
weights_generation	uniform
num_steps_per_epoch	120
actor_net_arch	[256,256]
critic_net_arch	[256,256]
learning_rate	0.001
gru_hidden_size	32
num_technicians	3
technicians_history_horizon	20
num_experience_initial_seeds	5
experience_propagation_var_scale	0.05
grid_size	100
ticket_embedding_shape	2
transmission_factor	0.1
proportion_reward_sigma	0.1

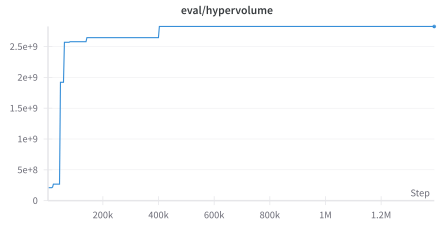

Figure 5. Training hypervolume

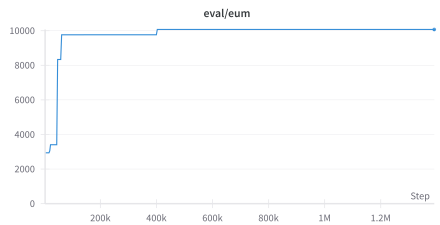

Figure 6. Training expected utility

The Figure (5) and Figure (6) suggest that our agent quickly reached a plateau in his performances. This could be due to a number of reasons, but the most likely scenario here is that by using rewards designed on such different scales (one reward between 0 and 1 and another usually in the thousands), our agent learns to focus only on the more "meaningful" reward.
This conclusion seems to be in line with Figure (7) which depicts very high sparsity, which is usually not what we are looking for. Our solutions are very much focused on maximizing one objective above all others : the Total knowledge.

We tried implementing a vector-based method : GPI-PD, however the weight selection algorithm introduced the same bias in the re-

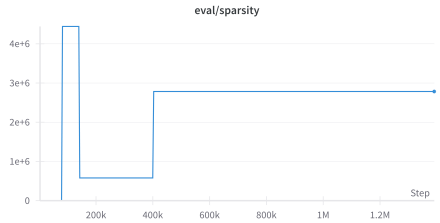

Figure 7. Training sparsity

sults, a reward re-design seems to be a more promising choice for the future.

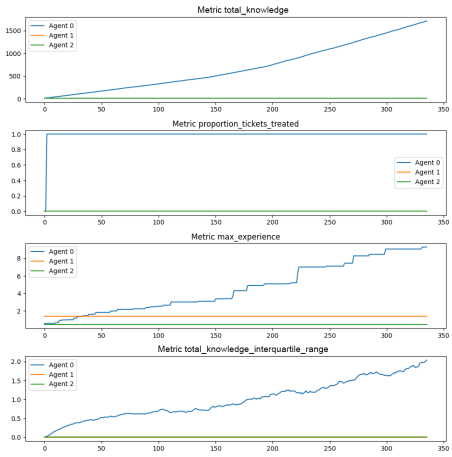

Figure 8. Baseline of literature : the policy of always choosing the most knowledgeable technician

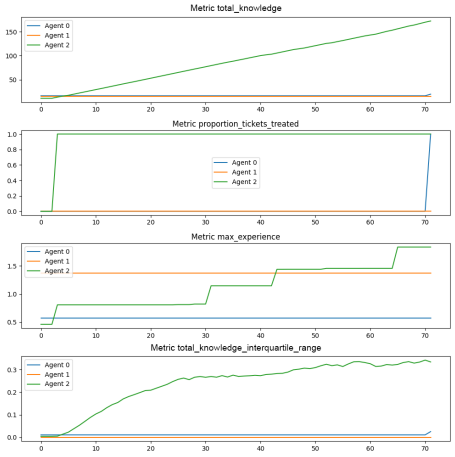

Figure 9. Baseline of literature : the policy of always choosing the most knowledgeable technician

Figure (8) and Figure (9) show a benchmark of the behaviour of two policies.
The first is a baseline policy, typical in the literature, represented by a simple heuristic : always choose the most knowledgeable technician for the task.
Its shortcomings are obvious : only the first agent to be picked has treated any tickets.
The second benchmark is one made using a trained policy from our approach. As we can see, the influence from the first reward is too strong, it has led the behavior of this policy to tend towards always increasing the same prefered agent. However we notice some occasionnal changes in behaviour that let us think that this policy is capable of better results after some reward re-design.

5 Related Work

This is the first instance of using MOMARL to address the HRAP within Industry 4.0 [3]. Existing literature on RL in maintenance planning primarily focuses on the Job Shop Scheduling Problem (JSSP) and its variants [9][15], often relying on simple heuristics for technician selection. These heuristics can lead to hyper-specialization, resulting in monotonous tasks and potential risks to workers' mental and physical health [2][12]. Our HRAP solution uniquely prioritizes continuous technician learning in Industry 4.0 [3]. While some approaches categorize knowledge (e.g., mechanical vs. electrical), this limits RL systems' ability to adapt to new maintenance challenges. Research highlights that learning-focused policies enhance job satisfaction and operational efficiency, underscoring the importance of HRAP systems that support continuous skill development and worker empowerment [11].

Table 3. Related JSSP RL works

Heuristic	Works
Pick technician at random	[15]
Pick earliest technician available	[15]
Pick best technician	[9], [8]
Pick lest exhausted technician	[9]
Discrete skill level	[8], [9]

6 Conclusion and Future works

We introduced a new relationship between MOMARL and HRAP, adding complexity to human behavior considerations that became essential as HRAP policies were applied in real-world shop-floor settings. Integrating MOMARL into HRAP offers a promising framework for optimizing resource allocation while addressing complex human factors such as skill development, partial specialization, and fair task assignment. Our experimental results demonstrate the relevance of this approach, highlighting improvements in task distribution and knowledge propagation. MOMARL also improves existing frameworks with continuous knowledge and skill representation, better aligning with current research, particularly those involving NLP in maintenance documents such as [4]. Although this is a work in progress, future work will focus on redesigning rewards to avoid such learning issues, encapsulating this agent in a more complex and realistic industrial environment, to better highlight its relevancy against challenges such as workers falling ill, the forgetting curve, the impact of failure on learning, adding the concept of relative difficulty, as [16] did with school exercises, and so on, as well as implementing known HRAP heuristics to our case of maintenance scheduling in a JSSP-like scheduling environment. The scalability of this solution also needs to be studied, as well as its adaptability to different scenarios such as different types of labor.

References

[1] S. Bouajaja and N. Dridi. A survey on human resource allocation problem and its applications. *Operational Research*, 17(2):339–369, July

2017. ISSN 1109-2858, 1866-1505. doi: 10.1007/s12351-016-0247-8. URL http://link.springer.com/10.1007/s12351-016-0247-8.

[2] J. A. Diego-Mas, S. Asensio-Cuesta, M. A. Sanchez-Romero, and M. A. Artacho-Ramirez. A multi-criteria genetic algorithm for the generation of job rotation schedules. *International Journal of Industrial Ergonomics*, 39(1):23–33, Jan. 2009. ISSN 0169-8141. doi: 10.1016/j.ergon.2008.07.009. URL https://www.sciencedirect.com/science/article/pii/S0169814108001169.

[3] H. Grillo, M. M. E. Alemany, and E. Caldwell. Human resource allocation problem in the Industry 4.0: A reference framework. *Computers & Industrial Engineering*, 169:108110, July 2022. ISSN 0360-8352. doi: 10.1016/j.cie.2022.108110. URL https://www.sciencedirect.com/science/article/pii/S0360835222001802.

[4] Z. Liu, C. Benge, and S. Jiang. Ticket-BERT: Labeling Incident Management Tickets with Language Models, June 2023. URL http://arxiv.org/abs/2307.00108. arXiv:2307.00108 [cs].

[5] T. Lust and J. Teghem. The multiobjective multidimensional knapsack problem: a survey and a new approach, July 2010. URL http://arxiv.org/abs/1007.4063. arXiv:1007.4063 [cs] version: 1.

[6] A. María, L. Palma, L. Cecilia, S. Bárcena, R. Delgado, R. Del, R. Martínez, M. Angel, and M. Campo. The ideas generation process and the role of the learning curve: simulating the wealth of knowledge in organizations.

[7] H. t. Nguyen. Turn Design as Longitudinal Achievement: Learning on the Shop Floor. In J. Hellermann, S. W. Eskildsen, S. Pekarek Doehler, and A. Piirainen-Marsh, editors, *Conversation Analytic Research on Learning-in-Action: The Complex Ecology of Second Language Interaction 'in the wild'*, pages 77–101. Springer International Publishing, Cham, 2019. ISBN 978-3-030-22165-2. doi: 10.1007/978-3-030-22165-2_4. URL https://doi.org/10.1007/978-3-030-22165-2_4.

[8] K. S. H. Ong, W. Wang, D. Niyato, and T. Friedrichs. Deep-Reinforcement-Learning-Based Predictive Maintenance Model for Effective Resource Management in Industrial IoT. *IEEE Internet of Things Journal*, 9(7):5173–5188, Apr. 2022. ISSN 2327-4662. doi: 10.1109/JIOT.2021.3109955. URL https://ieeexplore.ieee.org/abstract/document/9528837?casa_token=tIDJ7ILPjDcAAAAA:msAcsnlFn2pVPPezIWYrejkymwcQlyOHnLB7KGka7qGMziRD-v9OeOWYw_YioPWoNAQxQoBC-DA. Conference Name: IEEE Internet of Things Journal.

[9] M. L. Ruiz Rodríguez, S. Kubler, A. de Giorgio, M. Cordy, J. Robert, and Y. Le Traon. Multi-agent deep reinforcement learning based Predictive Maintenance on parallel machines. *Robotics and Computer-Integrated Manufacturing*, 78:102406, Dec. 2022. ISSN 07365845. doi: 10.1016/j.rcim.2022.102406. URL https://linkinghub.elsevier.com/retrieve/pii/S0736584522000928.

[10] R. Rădulescu, P. Mannion, D. M. Roijers, and A. Nowé. Multi-Objective Multi-Agent Decision Making: A Utility-based Analysis and Survey. *Autonomous Agents and Multi-Agent Systems*, 34(1):10, Apr. 2020. ISSN 1387-2532, 1573-7454. doi: 10.1007/s10458-019-09433-x. URL http://arxiv.org/abs/1909.02964. arXiv:1909.02964 [cs].

[11] C. S. Silva, A. F. Borges, and J. Magano. Quality Control 4.0: a way to improve the quality performance and engage shop floor operators. *International Journal of Quality & Reliability Management*, 39(6):1471–1487, Jan. 2021. ISSN 0265-671X. doi: 10.1108/IJQRM-05-2021-0138. URL https://doi.org/10.1108/IJQRM-05-2021-0138. Publisher: Emerald Publishing Limited.

[12] A. Szalavetz. Digital Technologies and the Nature and Routine Intensity of Work: Evidence from Hungarian Manufacturing Subsidiaries, Feb. 2021. URL https://papers.ssrn.com/abstract=3792000.

[13] A. Szalavetz. Digital technologies shaping the nature and routine intensity of shopfloor work. *Competition & Change*, 27(2):277–301, Apr. 2023. ISSN 1024-5294. doi: 10.1177/10245294221107489. URL https://doi.org/10.1177/10245294221107489. Publisher: SAGE Publications Ltd.

[14] K. Warnhoff and P. de Paiva Lareiro. Skill Development on the Shop Floor - Heading to a Digital Divide? In *Proceedings of the Weizenbaum Conference 2019 "Challenges of Digital Inequality - Digital Education, Digital Work, Digital Life"*, pages 145–154. Berlin: Weizenbaum Institute for the Networked Society - The German Internet Institute, 2019. ISBN 978-3-96701-000-8. doi: 10.34669/wi.cp/2.23. URL https://www.econstor.eu/handle/10419/213820.

[15] Q. Yan, H. Wang, and F. Wu. Digital twin-enabled dynamic scheduling with preventive maintenance using a double-layer Q-learning algorithm. *Computers & Operations Research*, 144:105823, Aug. 2022. ISSN 0305-0548. doi: 10.1016/j.cor.2022.105823. URL https://www.sciencedirect.com/science/article/pii/S0305054822001046.

[16] X. Zhang, Y. Shang, Y. Ren, and K. Liang. Dynamic multi-objective sequence-wise recommendation framework via deep reinforcement learning. *Complex & Intelligent Systems*, 9(2):1891–1911, Apr. 2023. ISSN 2198-6053. doi: 10.1007/s40747-022-00871-x. URL https://doi.org/10.1007/s40747-022-00871-x.