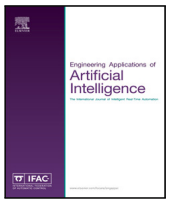
Engineering Applications of Artificial Intelligence 156 (2025) 111126

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

journal homepage: www.elsevier.com/locate/engappai



Research paper

# Evolutionary multi-objective multi-agent deep reinforcement learning for sustainable maintenance scheduling

Marcelo Luis Ruiz-Rodríguez [a,*], Sylvain Kubler [a], Jérémy Robert [b], Alexandre Voisin [c], Yves Le Traon [a]

[a] SnT, University of Luxembourg, 6 Rue Richard Coudenhove-Kalergi, L-1359 Luxembourg, Luxembourg
[b] Cebi International S.A., 30 rue J.F. Kennedy, L-7327 Steinsel, Luxembourg
[c] Université de Lorraine, CNRS, CRAN, F-54000 Nancy, France

ARTICLE INFO

*Keywords:*
Sustainable manufacturing
Maintenance
Scheduling
Reinforcement Learning
Evolutionary algorithms

ABSTRACT

In recent years, sustainability has emerged as a major priority for businesses across various industries, and the manufacturing sector is no exception. Production and maintenance processes now need to be economically profitable while also adopting practices that adhere to the principles of environmental integrity and social responsibility. This article explores an innovative approach aimed at optimizing maintenance scheduling from an economic perspective (considering maintenance, breakdown, downtime costs), an environmental perspective (considering the carbon footprint produced during production) and a social perspective (considering the fatigue experienced by technicians during maintenance activities). To the best of our knowledge, this is the first study to propose a manufacturing scheduling approach that considers all three pillars of sustainability. Another significant contribution of this research is the innovative way in which the optimization problem is addressed. We propose an evolutionary multi-objective multi-agent Deep Q-network-based approach, where multiple agents explore the preference space to maximize the hypervolume of these sustainable objectives. Our methodology uses industrially representative data that incorporate realistic machine degradation signals, carbon intensity indicators, and technician constraints. The results demonstrate the trade-offs between these objectives when compared to traditional maintenance policies such as corrective and condition-based maintenance, as well as different Deep Q-network policies trained with various preferences. Our approach demonstrates superior performance compared to both baselines. Specifically, we observe an 11.6% improvement in hypervolume over Deep Q-network and an 18.9% improvement over Proximal Policy Optimization, resulting in significantly increased profitability within the system.

## 1. Introduction

The development towards Industry 4.0 has a substantial influence on the manufacturing industry. Not only facilitates the creation of smart products and services, it also enables new and disruptive business models. Industry 4.0 technologies include, but are not limited to, standardized communication protocols, the Internet of Things, artificial intelligence, big data and analytics, blockchain, cloud computing, and simulation (Muhuri et al., 2019). Although these technologies are often used, in the first place, to improve production operations (aka Overall Equipment Effectiveness), their implication in the Sustainable Development Goals (United Nations, 2015) requires more attention and evaluation. In fact, traditional production systems are notorious for their poor ecological (and social) imbalances. It becomes a necessity for companies to address this problem for their future, as they face growing pressure from governments and customers to deliver sustainable products, aligned with European Green Deal-like initiatives (European Commission, 2019).

Predictive maintenance (PdM) moves away from traditional preventive maintenance (PM), where tasks are scheduled at regular intervals, or corrective maintenance (CM), where tasks are scheduled when a failure occurs, to optimal maintenance timing. This approach aims to maximize the overall profit of the manufacturing system (Ran et al., 2019). This move comes with a dynamic maintenance task scheduling optimization problem (Aissani et al., 2009; Ruiz-Rodríguez et al., 2024), consisting of assigning a set of maintenance tasks to a set of technicians, while minimizing the overall system downtime and cost, and

* Corresponding author.
*E-mail addresses:* marcelo.ruiz@uni.lu (M.L. Ruiz-Rodríguez), sylvain.kubler@uni.lu (S. Kubler), jeremy.robert@cebi.com (J. Robert), alexandre.voisin@univ-lorraine.fr (A. Voisin), yves.letraon@uni.lu (Y. Le Traon).

https://doi.org/10.1016/j.engappai.2025.111126
Received 30 September 2024; Received in revised form 31 March 2025; Accepted 7 May 2025
Available online 26 May 2025

taking into account several constraints related to production, safety, technician skills, job duration, etc. (Lee, 1996). As illustrated in Fig. 1, traditional scheduling optimization strategies such as CM, PM, or PdM focus primarily on economic indicators and rarely, if ever, consider the two other pillars of the Triple Bottom Line (TBL): Environment and Social. However, there is room to jointly optimize these three dimensions. In our framework, energy consumption refers to the total amount of energy used by the system, while grid carbon intensity measures the amount of carbon dioxide emitted per unit of energy produced. Thus, even if energy consumption remains constant, variations in the grid's energy mix can lead to higher or lower carbon footprints. For example, performing inspection and maintenance on energy-intensive machines during periods of high grid carbon intensity, *when fossil fuel sources dominate*, can significantly reduce carbon emissions. Similarly, an employee-focused maintenance policy can deliver economic benefits by reducing productivity losses due to accidents or fatigue, while also positively impacting social factors such as employee morale.

Multi-objective problems are often addressed by combining multiple objectives into a single objective (scalarization). However, in certain scenarios, such as the sustainability-focused maintenance scheduling problem we present, this approach is not always feasible (Hayes et al., 2022; Roijers et al., 2013). In particular, the manager's or company's preferences are unknown at the time of policy learning and become available only during operational decision making. Therefore, it is necessary to compute a coverage set of policies to respond quickly as additional information becomes available, such as the company's evolving priorities, equipment failures, current carbon intensity, or technician status.

Developing policies that take into account diverse preferences and adapt to the manager's changing needs can be a complex task, especially in high-dimensional, uncertain environments with interdependent objectives. To address this challenge, we use Reinforcement Learning (RL) and evolutionary computation to evolve a set of policies that collectively maximize hypervolume and ensure broader coverage of the preference space. In particular, evolutionary computation allows us to refine existing solutions without retraining from scratch, while RL uses direct feedback from the environment to improve policy decisions. As a result, decision makers gain the flexibility to quickly select or adapt policies in response to current company needs.

This article advances the state-of-the-art in two significant ways:

- It is the first to consider both grid carbon intensity and technician fatigue in the maintenance scheduling optimization process, in addition to the economic criterion;
- It introduces an evolutionary multi-objective multi-agent deep Q-Learning (EvoDQN) maintenance scheduling approach, enabling companies to adjust, whenever needed, the importance and preference of each TBL criterion without incurring any re-optimization cost/time.

Section 2 discusses the pressing need for organizations to adopt sustainable practices, providing an overview of the evolution of regulations and certifications over the years, and presents a realistic scenario in the manufacturing industry that requires the company to adapt its process due to customer requirements. Section 3 reviews existing maintenance scheduling strategies, highlighting their limitations when it comes to taking into account environmental and social aspects. Section 4 introduces in detail our framework for sustainable maintenance scheduling called EvoDQN. Section 5 evaluates and compares EvoDQN against traditional maintenance scheduling policies and baselines. The discussion and conclusion follow. All acronyms and variables used throughout the article are summarized in Table A.8.

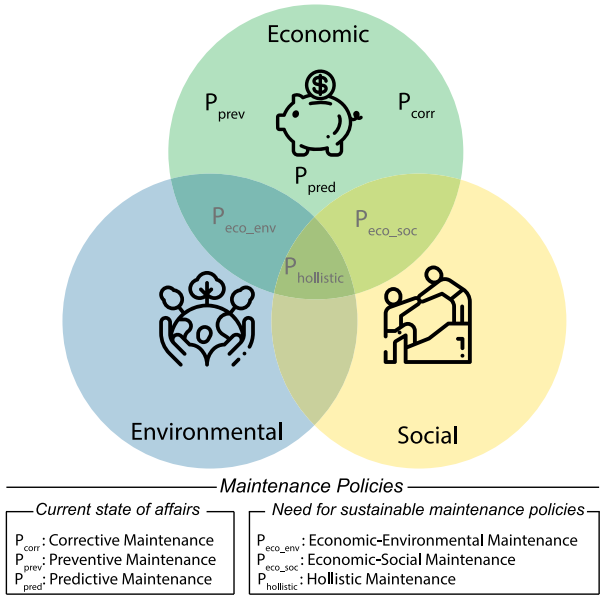



**Fig. 1.** Maintenance policies from a TBL perspective.

## 2. Sustainable manufacturing: a pressing need

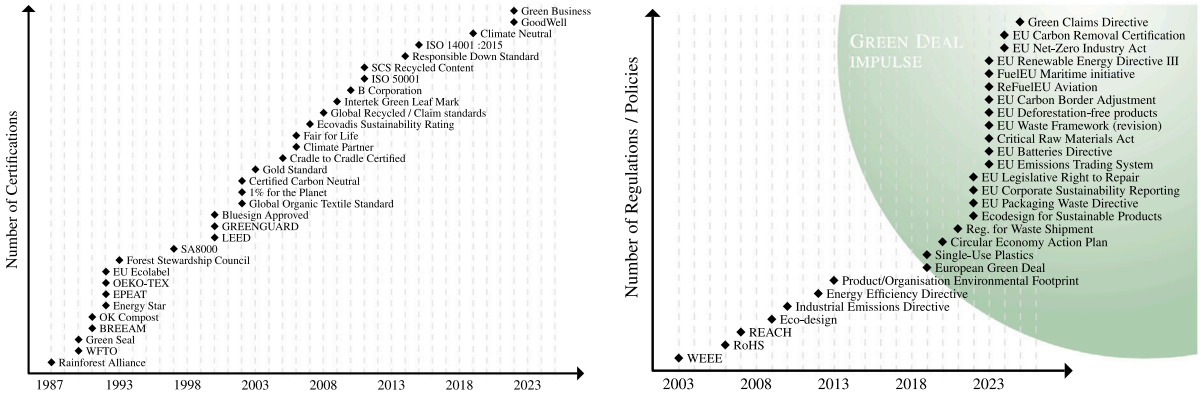
Section 2.1 discusses the evolution of regulations and certification schemes over the years. Section 2.2 presents a scenario that illustrates the need for manufacturing processes to be flexible enough to accommodate real-time adjustments to the TBL criteria, resulting in diverse production and maintenance schedules.

### *2.1. Evolution of regulations & certifications for sustainable product development*

The manufacturing sector is under increasing pressure to adopt more sustainable practices, focusing on all aspects of the TBL (Perera et al., 2023). Recent literature and global initiatives show evidence of a willingness among companies and governments to improve the situation. The evolution of key certifications and regulations for sustainable development, depicted in Fig. 2(a) and Fig. 2(b) respectively, highlights a significant increase in regulations and certification schemes over the past two to three decades. Notably, Fig. 2(b) shows an exponential increase in regulation starting from 2019, largely due to the European strategy (*through the "European Green Deal"*) to make the EU's economy sustainable. Under the emerging pressure for the industry to market products as "sustainable" or "green", it is expected that companies will increasingly subscribe to green certification schemes (issued by independent third-party organizations) to ensure their customers of the products' credibility. For example, adopting cornerstone certification schemes like ISO 14001 (Environmental Management System) is anticipated to provide a competitive advantage and increase firm value, as discussed in Widiastuti et al. (2022). While most regulations and certifications focus on environmental sustainability, a few also emphasize the social pillar, such as SA8000 (Social Accountability International), which certifies an organization's capability to meet standards for worker safety and well-being.

### *2.2. Motivation scenario for adjusting production and maintenance schedules to meet customers' sustainable development requirements*

With the increasing pressure on companies to adopt sustainable development practices, they are constantly looking for ways to improve their products and processes. The 6R methodology (Jawahir et al.,




(a) Evolution of certifications (b) Evolution of regulations/policies

**Fig. 2.** Overview of the evolution over time of key regulations and certification for sustainable development.

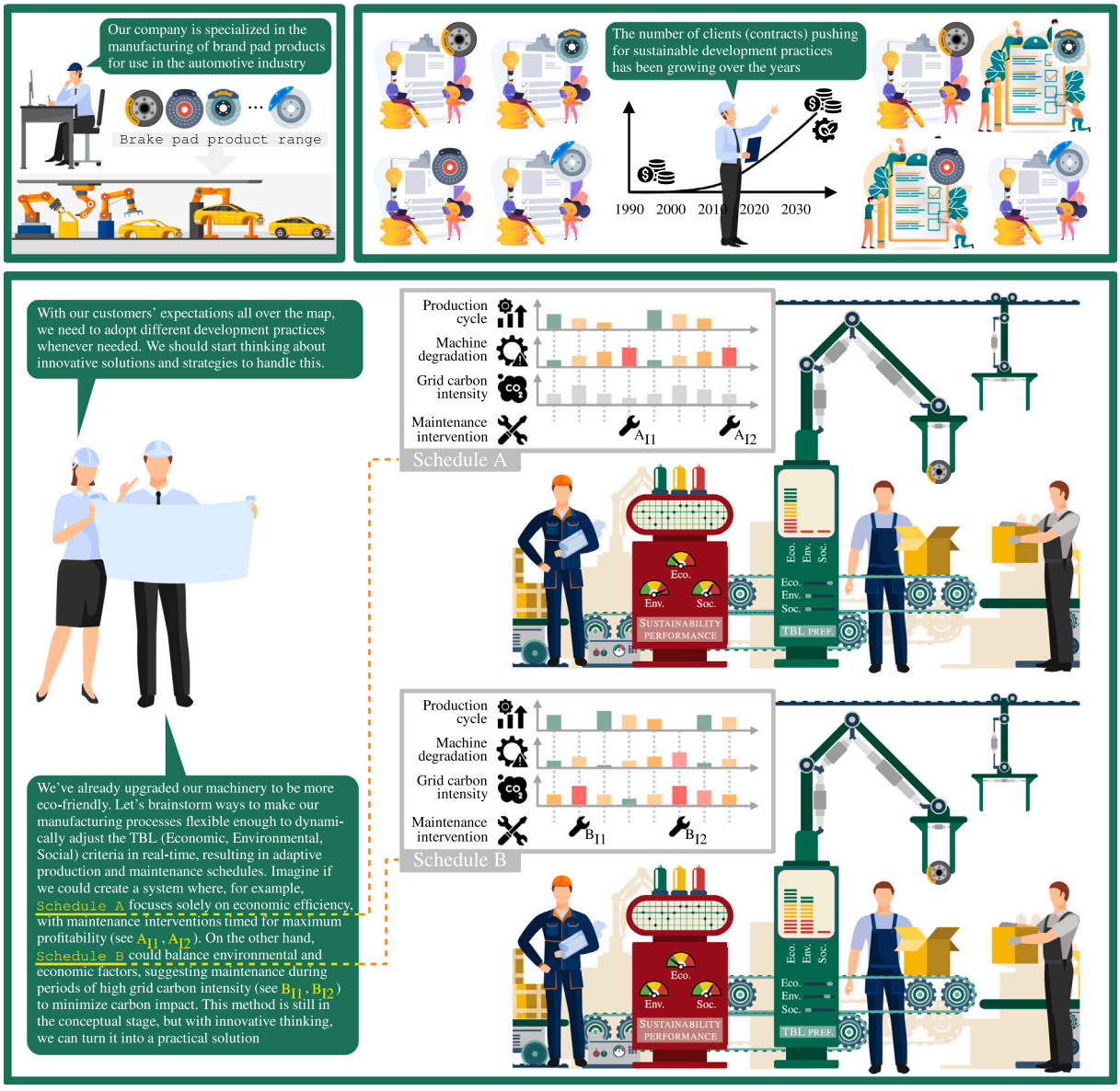



**Fig. 3.** Illustration of dynamic adjustment of TBL criteria in manufacturing processes.

2006) offers six key directions for improvement: (1) Reduce, (2) Reuse, (3) Remanufacture, (4) Recycle, (5) Reclaim or Recover, and (6) Redesign. Focusing on the first direction (Reduce), several strategies can be employed, such as using fewer materials and resources, reducing the number of components, minimizing pollution, or decreasing consumer returns. One approach to reducing pollution is to optimize production and maintenance schedules taking into account both environmental and social criteria. Fig. 3 illustrates, in a comic strip format, a scenario where the manufacturing process is made flexible enough to dynamically adjust the TBL criteria (Economic, Environmental, Social) in real-time, effectively meeting each customer's TBL needs. This illustration focuses on adjusting only the Economic and Environmental criteria to reduce the complexity of the scenario, but of course the social dimension (e.g., operator fatigue) is important to be considered too. This innovative strategy is the central proposal of the present article.

## 3. Maintenance scheduling: current state of affairs

This section provides an overview of the state-of-the-art approaches used for sustainable maintenance policies in manufacturing. For this purpose, keywords and search terms were identified to establish a search query. Web of Science was used for the literature search (Gusenbauer and Haddaway, 2020). The query used was the following:

Query used for literature review

```
(TS=(sustainable)  OR  TS=(sustainability)
OR  TS=(social)  OR  TS=(environmental)  OR
TS=(environment)  OR  TS=(''carbon footprint'')
OR  TS=(CO2))  AND  (TS=(maintenance)  AND
TS=(scheduling))  AND  (TS=(industry)  OR
TS=(manufacturing))
```

A total of 413 articles were identified from 2019 to July 2024. After screening titles, abstracts, and keywords, a total of 46 articles were preserved and analyzed in Table 1 based on six criteria, each one taking two or more values, as detailed hereinafter:

- **Maintenance Policy:** Corrective Maintenance (CM), Preventive Maintenance (PM), Predictive Maintenance (PdM)
- **Type of objective:** Single (Si), Multiple (Mu)
- **Solution method:** Metaheuristic (Me), Machine Learning (ML), Mathematical Programming (MP), Others (Ot).
- **Sustainability Pilars:** Economic (Eco), Environmental (Env), Social (Soc)
- **Level:** Machine (Mac), Production Line (PL), Factory (Fac)
- **Evaluation:** Real Dataset (RD), Artificial Dataset (AD)

To ease the analysis, Fig. 4 offers an overview of the distribution of the reviewed articles by criterion. Examining the "Pillars" covered by the 47 articles (see Fig. 4(a)), the majority focus on the Economic (Eco) as they are primarily business-oriented. Furthermore, it can be observed that most maintenance policies developed are preventive (PM in Fig. 4(b)). Although PdM exhibits a higher level of maturity compared to CM, the last is still more commonly used. As evidenced in Fig. 4(c), most articles address production lines (PL) with multiple interdependent machines. This is followed by articles focusing on independent machines (Mac) and, lastly, larger systems of factories (Fac) or plants with multiple production lines at the same or different remote sites. Regarding optimization of maintenance scheduling for economically driven policies, metaheuristic methods represent the standard approach to solving (Boufellouh and Belkaid, 2019; Cacereño et al., 2021; Seidgar et al., 2020) (see Fig. 4(d)). Single-objective problems (Si) (Al-Hourani, 2020; Ben Abdellafou et al., 2021; Chang, 2023) are slightly more addressed than multi-objective (Mu) (Baykasoğlu and Madenoğlu, 2021; Bencheikh et al., 2022) (see Fig. 4(e)). Interestingly, most articles use artificial datasets (AD) (Rokhforoz and Fink, 2022; Shen and Zhu, 2019) for evaluation purposes, as evidenced in Fig. 4(f).

### 3.1. Economic scheduling

Although a large share of the literature focuses on economically drive maintenance policies, as illustrated in Fig. 4(a), many of these approaches do not incorporate environmental or social factors. For instance, Seidgar et al. (2020) addresses the problem of two-stage scheduling in an assembly flow shop with random machine breakdowns. In this work, the objective is to minimize the expected weighted sum of the makespan and the mean completion time. Maintenance is performed as part of the completion time for each processing time of each job on the machines. Four metaheuristic algorithms are proposed to solve this optimization problem: Genetic Algorithm (GA), Imperialist Competitive Algorithm, cloud theory-based Simulated Annealing (SA), and New Self-adapted Differential Evolutionary (NSDE). The NSDE algorithm is statistically superior to the other proposed meta-heuristic algorithms in terms of solution quality and computational time. Similarly, in their study, Wang et al. (2023) addressed the challenge of joint optimization of integrated mixed maintenance and distributed two-stage hybrid flow-shop production for multi-site maintenance requirements. Their objective was to minimize the total weighted earliness/tardiness penalty and the number of lost orders. To achieve this, they proposed an Improved Non-dominated Sorting Genetic Algorithm-II (INSGA-II), which demonstrated superior performance compared to other algorithms utilized in the study. Furthermore, Boufellouh and Belkaid (2019) addressed the problem for a joint production and maintenance policy under non-renewable resource constraints (e.g. raw materials or fuel). Two objectives were defined, to minimize the expected makespan and the total maintenance cost for preventive and corrective actions. Non Dominated Sorting Genetic Algorithm (NSGA-II) was adopted to solve the optimization problem. A numerical examples were generated to evaluate the proposed solution against LINGO 10. The results presented different trade-offs of the solutions for the multi-objective problem. In addition, Cacereño et al. (2021) worked on a multi-objective optimization for preventing maintenance policy, where different experiments were conducted on the Non-dominated Sorting Genetic Algorithm II (NSGA-II) to optimize system availability (maximize) and operating cost (minimized). The approach was evaluated in a case of study in which optimization of the design and PM strategy needed to be implemented for an industrial fluid injection system. Other metaheuristic methods have been implemented to solve the maintenance problem as presented in Ben Abdellafou et al. (2021), where the objective is to minimize the makespan for task scheduling in identical and parallel machines. For this problem, a heuristic approach was implemented, together with an implementation of Tabu Search (TS). The results presented show insights about the different performance of the methods evaluated, which are dependent on the shape of the intree that is formed by the constraints of the ordered tasks. Other works, as investigated by Yazdani et al. (2022), address the problem of joint maintenance and production policy with the objective of minimizing the Total Absolute Deviation of Completion times (TADC) on a parallel machine under periodic maintenance. A metaherusitic algorithm called Lion Optimization Algorithm (LOA) was proposed. The model was evaluated with random instances and compared against several metaheuristics that outperformed the other algorithms for problems in different size ranges. The authors of Qamhan et al. (2020) addressed the problem of a joint production and maintenance policy for a single machine environment with the objective of minimizing the number of tardy jobs. In this work a Mixed-Integer Linear Programming (MILP) model and an Ant Colony Optimization (ACO) model were proposed for optimizing small and large sizes instances, respectively. In addition, Moore's algorithm was used to evaluate ACO solutions, showing how ACO outperform Moore's algorithm for all the instances tested. Baykasoğlu and Madenoğlu (2021) worked on the problem of joint production and maintenance policy with the aim to optimize mean tardiness, schedule instability, makespan and mean flow time. To solve this, a

**Table 1**
Trends in maintenance for sustainable scheduling.

| | Policy | Object. | Method | Pillars | Level | Eval. |
|---|---|---|---|---|---|---|
| Al-Hourani (2020) | PM | Si | Ot | Eco | Mac | RD |
| An et al. (2020) | PM | Mu | Me | Eco, Env | – | – |
| Asghar et al. (2019) | CM | Si | Ot | Eco, Env | Mac | AD |
| Attia et al. (2024) | CM, PM | Si | MP | Eco, Env | Mac | AD |
| Baykasoğlu and Madenoğlu (2021) | PM | Mu | Me, Ot | Eco | PL | AD, RD |
| Ben Abdellafou et al. (2021) | PM | Si | Me | Eco | Mac | AD |
| Bencheikh et al. (2022) | PdM | Mu | Ot | Eco | Mac | AD |
| Boufellouh and Belkaid (2019) | CM, PM | Mi | Me | Eco | PL | – |
| Cacereño et al. (2021) | PM, PdM | Mu | Me | Eco | Mac | RD |
| Chang (2023) | PM | Si | Ot | Eco | Mac | AD |
| Cui and Lu (2020) | PM | Mu | Me | Eco, Env | PL | AD |
| Cui and Lu (2021) | PM | Mu | Me, MP | Eco, Env | PL | AD |
| Cui et al. (2020) | PM | Mu | Me, MP | Eco, Env | Mac | AD |
| Detti et al. (2019) | PM | Mu | MP, Ot | Eco | Mac | AD |
| Diaz Cazanas et al. (2019) | PM | Si | Me, Ot | Eco | PL | RD |
| Djassemi and Seifoddini (2019) | CM, PM | Mu | Ot | Eco | Fac | AD |
| Dong and Ye (2020) | PM | Mu | Me | Eco, Env | Fac | AD |
| Farahani et al. (2019) | PM | Si | MP | Eco | PL | AD |
| Ghaleb et al. (2020) | PdM | Si | Me | Eco, Env | Mac | AD |
| Giner et al. (2021) | PdM | Si | ML | Eco | PL | AD |
| Gupta and Jain (2021) | PM, PdM | Mu | Ot | Eco, Env | Mac, PL | AD |
| Gupta and Jain (2022) | PM | Mu | Ot | Eco, Env | PL | AD |
| Gupta et al. (2023) | PM | Si, Mu | Ot | Eco | PL | AD |
| Hedjazi et al. (2019) | CM | Mu | Ot | Eco | Mac, Fac | AD |
| Hidri et al. (2021) | PM | Si | Me, Ot | Eco | PL | AD |
| Jayasuriya et al. (2021) | PM, PdM | Mu | ML | Eco | Fac | RD |
| Kedy et al. (2024) | PM | Mu | Ot | Eco | Mac | AD |
| Li et al. (2023) | CM, PM | Si | Ot | Eco | Mac | RD |
| Mi et al. (2020) | CM, PM, PdM | Si | Me | Eco, Env | Mac | AD, RD |
| Mirahmadi and Taghipour (2019) | CM, PM | Si | Me | Eco, Env | PL | AD |
| Paraschos et al. (2023) | CM, PM | Si | ML | Eco, Env | PL | AD |
| Qamhan et al. (2020) | PM | Si | Me, MP | Eco | Mac | AD |
| Qin et al. (2022) | CM, PM | Si | Me, Ot | Eco, Soc | Mac, PL, Fac | RD |
| Rokhforoz and Fink (2022) | PdM | Mu | Ot | Eco | PL | AD |
| Seidgar et al. (2020) | CM | Si | Me | Eco | Mac, PL | AD |
| Sharifi and Taghipour (2021) | CM, PM | Si | Me | Eco, Env | Mac | AD |
| Shen and Zhu (2019) | PM | Mu | Ot | Eco | PL | AD |
| Sin and Chung (2020) | PM | Mu | Me | Eco, Env | Mac | AD |
| Sun et al. (2020) | CM | Si | Me | Eco, Env | PL | AD |
| Wang et al. (2020) | PM | Si | Me | Eco, Env | PL | AD |
| Wang et al. (2023) | CM, PM | Mu | Me | Eco | Fac | – |
| Wu et al. (2020) | PM | Si | Ot | Eco | PL | AD |
| Xia et al. (2021) | PM | Si | Ot | Eco, Env | Mac | RD |
| Xia et al. (2022) | PM | Si | MP | Env | PL | AD |
| Yazdani et al. (2022) | PM | Si | Me | Eco | PL | AD |
| Yu and Han (2021) | PM | Si | Ot | Eco | Mac, PL | AD |

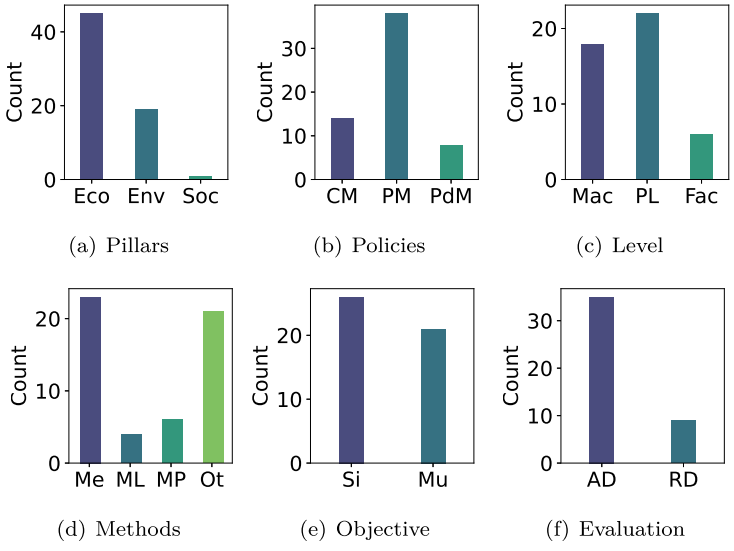



**Fig. 4.** Distribution of research articles per category.

Greedy Randomized Adaptive Search Procedure (GRASP) algorithm is tested. The approach was evaluated against a set of dispatching rules and a SA algorithm, showing the advantage of the proposed method as an effective approach for improving the performance of dynamic flexible job shop scheduling problem. The work by Hidri et al. (2021) address the problem of joint production and maintenance with parallel machines with a single server and unavailability constraints with the objective to minimize the makespan. To solve this problem, a lower bound and three metaheuristics (SA, TS and GA) were proposed where a different tradeoff of solutions for different subset of instances were showed. Additionally, Diaz Cazanas et al. (2019) worked on a joint production and maintenance policy for identical parallel machines with the objective of minimizing the making space of all activities. The model was solved in a hybrid way, combining an approach based on dispatching rules and SA. The approach was evaluated in a case study on a plastic industry with seven identical injection machines and different alternative methods based on different rules and simulated annealing.

In contrast, other works have chosen to employ classical methods such as mathematical programming. Farahani et al. (2019) explored a mixed integer non-linear programming model to determine the optimal PM interval represented by a continuous-time Markov chain. The objective is to reduce costs per unit of time by considering both perfect and imperfect maintenance levels. The model was evaluated in a numerical example and solved by the Baron solver. Detti et al. (2019) worked on the problem of joint production maintenance scheduling where the objectives are the completion time and makespan considering different robust criteria. Maintenance activities are considered and are seen as additional tasks. The problem was solved using Mixed Integer Linear Programming and a Heuristic method tested on random generated instances.

A limited number of studies have employed AI/ML techniques. Jayasuriya et al. (2021) worked on a maintenance management system for tire manufacturing that provides an optimum time frame for PM actions on condition monitoring and production data. The scheduler was implemented using an Artificial Neural Network (ANN) where condition monitoring data, sensor inputs, and machine operator inputs are used to generate breakdown alarms, dynamic scheduling, and breakdown alarms. The system was evaluated using performance charts and regression using real data from the manufacturing plant. Giner et al. (2021) worked on the problem of maintenance scheduling in a production environment where maintenance actions are decided by a policy trained using RL. The evaluation was tested on a discrete event simulator where machines suffer from different degradation states, a buffer level is considered, and maintenance actions (corrective and condition-based) are performed by the decision maker (the agent). The objective is to minimize the maintenance cost and maximize the cumulative value of the products.

Other agent-based systems have also been used to tackle this kind of problem. Hedjazi et al. (2019) worked on maintenance scheduling on geodistributed assets in a distributed industrial environment. In this work, a multi-agent system solution was implemented where the mechanism promotes competition and cooperation of agents to obtain a global schedule. The results performed well in terms of Global Cost, Total Weighted Tardiness Cost, and makespan compared to Weighted Shortest Processing Time first–Heuristic–Earliest Due Date (WSPT-H-EDD) method. Bencheikh et al. (2022) worked on the problem of scheduling maintenance activities for a joint production and maintenance policy in a multi-agent approach. The proposed approach modeled the problem as a multi-agent with the objective to solve several subproblems (as customers, producers, managers) where all these agents are coordinated by a supervisor agent. The framework is evaluated based on different Key Performance Indicator (KPI) such as execution time, number of cycles, number of maintenance tasks, number of late jobs, total tardiness, and the load of each machine. In a similar work, Kedy et al. (2024) worked on joint maintenance and production scheduling with the objective of optimizing maintenance cost and production-related metrics. In their work a multi agent system is proposed where they improve a Contract-Net Protocol by addressing the challenges of agent myopia leading to suboptimal resource allocation and performance losses. Their work was tested on a simulation environment showing improvement in terms of scheduling efficiency, maintenance cost, and execution times.

Other methodologies have been employed to address the economic aspects of maintenance optimization. The study by Djassemi and Seifoddini (2019) examines the effect of different criticality policies based on the shortest mean time between failures, longest queue length, high level of utilization, and longest mean repair time. These policies were evaluated with a discrete-event simulation model that represents the dynamic of a real-world manufacturing cell that includes machine failure, maintenant resource allocation, material flow, job sequencing and scheduling. Policies were evaluated on the basis of machine availability and mean throughput time. Rokhforoz and Fink (2022) worked on the maintenance scheduling problem where the objective is to maximize revenue and availability (to find the optimal prices of goods and optimal maintenance scheduling); for this the authors proposed a two-level optimization solution based on game theory called leader-multiple-followers game. In this, customers are considered as followers seeking to obtain their consumption and the supplier as the leader who is responsible for obtaining the price of the network and maintenance scheduling of its manufacturing units, demonstrating the effectiveness of integrating network externalities and PdM scheduling in pricing strategies for suppliers, leading to increased revenue and profit. Chang (2023) worked on the problem to determine the best schedule for a preventive replacement last policy with the objective of minimizing the mean cost rate over a finite time horizon. A numerical example was presented in order to minimize the total mean cost rate incorporating costs related to repairs, maintenances, replacements, inventory and shortage. In a study by Al-Hourani (2020), a Criticality Analysis was conducted for a pharmaceutical company where a risk matrix was used to classify equipment and set priority levels. Historical maintenance data was used to evaluate the current status of the machine. The analysis obtained showed that the implementation of the rescheduled PM improve the maintenance effectiveness, reducing failures and optimizing the resource utilization. Yu and Han (2021) studied the problem of machine scheduling focusing on periodic machine maintenance for single-machine and flow shop scheduling models. The objectives were to minimize the total completion time and minimize the maximum lateness. Random examples were generated for the model studied and solved with a proposed algorithm called Smallest Sum of Processing and Removal Time First and improved with a Minimum Cost Insertion (SPRT-MCI). Shen and Zhu (2019) worked on the problem of maintenance scheduling for a parallel machine setup. In this problem, processing and maintenance time are considered uncertain variables. To solve this problem, an improved version of a Long Processing Time rule is implemented, and it is tested with numerical experiments and compared with a Heuristic Method called HPSOGA that is a combination of GA and PSO. The results showed the performance of the improvement version against the original rule, and narrowing respect to the HPSOGA. Simulation-based approaches have been also used to address the maintenance optimization as presented by Wu et al. (2020). In this work, they tackle the problem of joint production and maintenance policy for failure-prone parallel machines in make-to-order production environment. The objective of the model is to minimize the weighted long-run average waiting costs of the production system. A simulations were performed using the Value Iteration algorithm and compared against different dispatching rules in the literature demonstrating an improvement in the jobs waiting time and average machine downtime. Furthermore, Gupta and Jain (2022) worked on a stochastic flexible job shop scheduling that considers uncertainties and dynamic jobs arrivals. In this work, five input parameters are considered, that is, reliability-centered PM, percentage of machine failure, mean time to repair for

random machine breakdown, due date tightness factor, and routing flexibility. Additionally, Gupta et al. (2023) continue working on the problem of flexible job shop scheduling under reliability-based PM. In this work, a simulation-optimization approach was implemented in which the mean flow time, the maximum flow time, the mean tardiness and the number of late jobs were evaluated in different scenarios. Finally, Li et al. (2023) worked on the problem of opportunistic maintenance for a multi-unit system using Monte Carlo simulation. The main objective was to optimize the maintenance cost where it is shown using simulated data to model the performance of Computer Numerical Control (CNC) machine.

### 3.2. Sustainable scheduling: energy consumption as prime focus

For sustainable maintenance policies, environmental aspects are usually addressed in terms of reducing energy consumption. As was evidenced in Fig. 4(d), most articles make use of metaheuristic methods to solve sustainable maintenance scheduling problems. Sun et al. (2020) propose a joint energy, maintenance, and production model with the objective of minimizing the total electricity cost, maintenance cost, and minimizing production loss based on the production throughput of the manufacturing system. To evaluate the model, a numerical study with five machines and four buffers was implemented where particle swarm optimization was used to solve the model. Wang et al. (2020) also consider maintenance and peak power consumption for the problem of collaborative optimization of manufacturing scheduling for the hybrid flow shop. The objective is to minimize the makespan considering maintenance plans and peak power consumption. Despite these two research works, most studies use evolutionary algorithms. Ghaleb et al. (2020) studied a joint production and maintenance policy in a single machine considering machine deterioration and failures. The goal is to minimize the total cost including inspection, repair, energy consumption, and the makespan. The model is solved using GA and is evaluated against simulated annealing and imperialist competitive algorithms in artificial instances. Similarly, Sharifi and Taghipour (2021) worked on a single machine level for the joint production and maintenance schedule with multiple failures, inspired by a real-world problem for a manufacturing company on a Lathe CNC machine. The objective is to optimize the total cost of the system (incl. maintenance cost, machine energy consumption cost, makespan). In this work, GA, simulated annealing, and teaching–learning-based optimization algorithms are implemented, with GA demonstrating superior performance. Sin and Chung (2020) propose a bi-objective optimization model for a single machine considering electricity cost and PM using GA. The objectives are to minimize the total energy cost and machine unavailability. The results show that the proposed hybrid multi-objective GA yields better outcomes than the non-dominated sorting GA (NSGA-II) and is faster than the Baron solver. The work of Mirahmadi and Taghipour (2019) goes beyond the single machine level to the flexible job-shop scheduling problem considering maintenance, production, and energy aspects with the objective of minimizing the expected makespan. The optimization model is solved using GA and tested on a small scale for three industrial machines and four jobs. An et al. (2020) optimize, *using a multi-objective evolutionary algorithm with the pareto elite storage strategy*, the production process considering maintenance by minimizing the makespan, total tardiness, total production cost, and total energy consumption. Cui et al. (Cui and Lu, 2020; Cui et al., 2020; Cui and Lu, 2021) also work on the problem of joint production and maintenance optimization involving energy consumption, which is also solved by GA.

Other studies employ alternative methods beyond metaheuristics. Asghar et al. (2019) develop a joint production-maintenance policy to optimize, *using the Kuhn–Tucker method*, production quantity, production rate, and manufacturing reliability considering variable energy consumption cost. The study presents several insights, including that a controllable production rate is preferable when dealing with an unreliable manufacturing system and that the expected total cost is influenced by decision variables such as production quantity, production rate, and reliability parameter. Xia et al. (2021) develop a joint policy considering preventive and replacement policies in an energy efficient way. The aim is to minimize the total non-value-added energy consumption obtaining the best intervals for preventive and replacement maintenance. The model is evaluated with data collected from Boehringer NG200 Crankshaft Turning CNC. Furthermore, Xia et al. (2022) studied an energy-oriented selective maintenance policy for series-parallel multi-unit systems, the objective being to maximize energy efficiency by optimizing the maintenance actions of machines at each breakdown. The model is solved using a modified branch-and-bound algorithm and tested on a numerical example based on the production system of an engine craft. Lastly, Gupta and Jain (2021) study reliability-based maintenance in a simulation environment considering setup time and energy-related metrics. The approach is evaluated against makespan, mean flow time, mean tardiness, number of tardy jobs, total setup time, average operation energy consumption, and average idle energy consumption. The results demonstrate the advantage of a reliability-centered periodic PM approach.

### 3.3. Beyond energy consumption

In addition to energy consumption, reducing the overall system carbon footprint (e.g. a production line, process) is one of the main objectives of today's companies. Attia et al. (2024) studied the relationship between the production makespan, maintenance activities, energy consumption, and carbon footprint (maintenance activities being of the highest importance in this relationship). Dong and Ye (2020) address a two-stage joint optimization problem of green manufacturing and maintenance for semiconductor wafer considering inspection and repair stages simultaneously. The authors present a hybrid multi-objective multiverse optimization algorithm that minimizes the makespan, total carbon emissions, and total PM cost. Finally, Mi et al. (2020) address the problem of maintenance scheduling for complex equipment considering green performance, which is divided into two sub-objectives: (i) minimizing the global maintenance cost resulting from resource consumption, production delay, and fault risk; (ii) minimizing carbon emissions during maintenance resource scheduling. The algorithm was implemented using NSGA-II and tested in a use case involving a grinding roll fault in a large vertical mill. Another study by Paraschos et al. (2023) focuses on the environmental pillar by examining different aspects of manufacturing and waste management. An optimization framework for process planning in a degrading multi-state system is proposed in this respect. The objective is to boost revenues through the sale and production of high-quality products, recycling, and minimizing production and maintenance costs.

In addition to the environmental pillar, we found a unique article by Qin et al. (2022) that addresses the social dimension. This study aims to minimize the number of days of work required for temporary employment in the maintenance management of the steel industry. This goal is achieved by reducing the number of maintenance days and minimizing the deviation between the interval times of maintenance tasks and the maintenance period of the equipment nodes. To this end, a two-stage optimization strategy is developed, which consists of generating a pre-schedule using a rule-based method, followed by a GA-based optimization. The test with real data from a Baowu steel company showed an improvement of 40.3% on the basis of pre-scheduling.

### 3.4. Reinforcement learning for maintenance optimization

Although Fig. 4(d) shows that there are few ML methods focused on maintenance optimization with an emphasis on sustainability, it is important to note that various manufacturing problems can be effectively modeled within a RL framework to determine the optimal policy for different operational scenarios (Ochella et al., 2022; Arena et al., 2024). In

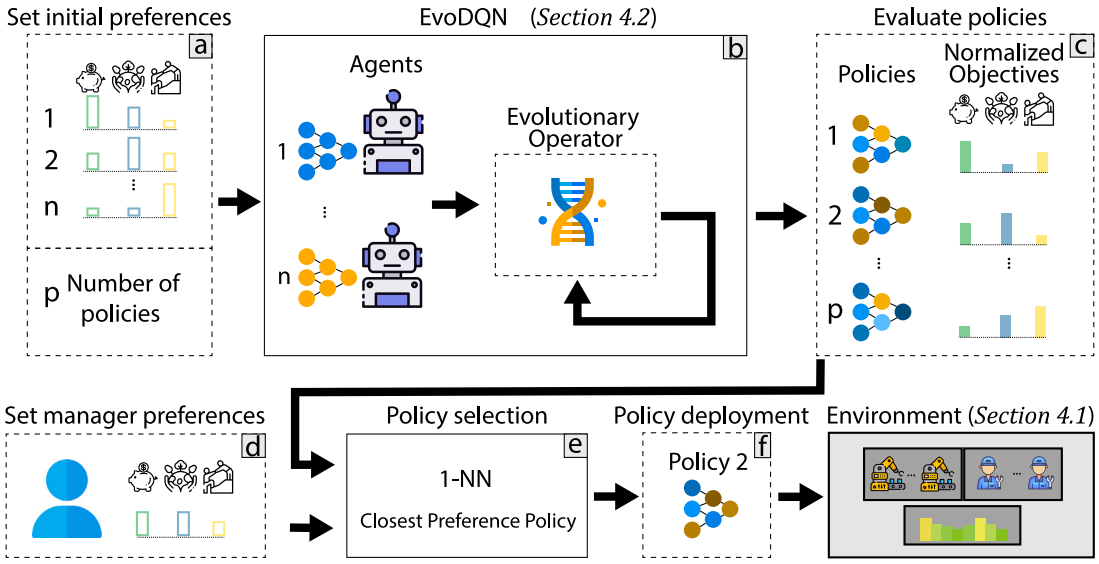



**Fig. 5.** Overview of the EvoDQN framework. Agents evolve maintenance policies based on initial preferences and are evaluated using normalized TBL objectives. The most suitable policy is selected based on manager preferences and deployed in the environment.

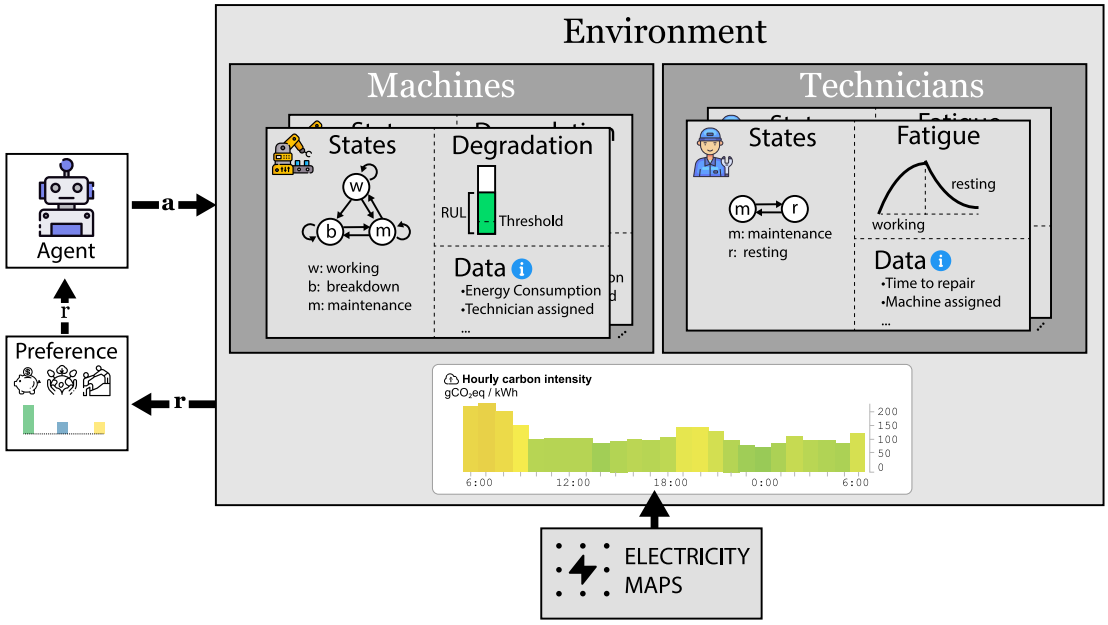



**Fig. 6.** EvoDQN environment for sustainable maintenance policies generation, modeling the states of machines and technicians, along with carbon intensity data to support TBL-based decision-making.

this sense, numerous studies have addressed maintenance optimization using a purely economic approach within the RL framework.

The work presented by Yan et al. (2022) addresses the problem of flexible PM considering machine and labor resources. They implemented a Q-learning algorithm with the objective of minimizing the makespan. Another classical approach based on R-Learning algorithm was proposed by Paraschos et al. (2020). They proposed a joint production, maintenance, and product quality policy on a stochastic production environment that suffers from degradation.

Other works have used Deep Reinforcement Learning (DRL) for their maintenance optimization. Huang et al. (2020) proposed a Double Deep Q-Network (DQN) to learn a PM policy on a 6-machine-5-buffer production line showing the capabilities of DRL to learn even opportunistic and group maintenance policies. Feng and Li (2022) developed a DRL-based joint production and maintenance policy for a multistage serial production line where machines undergo multiple deterioration stages. The objective was to optimize production and maintenance costs. Ye et al. (2023) worked on the problem of joint maintenance-inspection optimization in a large-scale manufacturing network through DRL with the objective of maximizing profit.

As systems become more complex, traditional DRL maintenance optimization methods are limited in interactions and interdependencies between different components. Some approaches extend from single-agent to a multi-agent reinforcement learning (MARL), where multiple agents work together to optimize maintenance strategies. For example, Nguyen et al. (2022) worked on maintenance optimization for multi-state component systems, where an initial ANN-based predictor is used to estimate maintenance cost and the MARL algorithm to optimize maintenance decisions. Bhatta and Chang (2023a,b) worked on the problem of a flexible manufacturing system operated by mobile multi-skilled robots. The robot assignment and maintenance scheduling are addressed by the MARL approach. Do et al. (2024) developed a

Condition-Based Maintenance (CBM) approach based on MARL for two serial parallel systems considering the dependencies of the components. The objective was to minimize maintenance and downtime costs. Ruiz Rodríguez et al. (2022) developed a maintenance optimization strategy for identical parallel machines. Their approach involved training an MARL system to identify an optimal maintenance policy that would maximize uptime and prevent failures. Su et al. (2022) worked on a maintenance optimization for a production line with multiple levels of PM.

Finally, other approaches use different RL-based systems. Jia et al. (2023) address the problem of distributed assembly hybrid flow-shop scheduling with flexible PM. For this, a multi-population memetic algorithm with Q-learning is proposed with the objective of minimizing the makespan. Abbas et al. (2024) addresses the challenges of decision support in PdM, specifically in a turbofan engine use case, by proposing a behavioral cloning-based specialized RL agent with the goal of minimizing maintenance and failure costs. Yan et al. (2024) proposed a hybrid method based on evolutionary algorithms and RL for the distributed flexible job-shop scheduling problem with the objective to minimize the makespan, maintenance cost, and total energy consumption. In this framework, RL is used to guide the local search process by dynamically refining the neighborhood structure of solutions, thereby increasing the search efficiency and helping to escape local optima.

## 4. EvoDQN for sustainable maintenance policies generation

This section presents the evolutionary multi-objective multi-agent DRL (EvoDQN) approach for sustainable maintenance scheduling, which allows a maintenance manager or decision maker to select, *when starting a new production batch*, a sustainable maintenance policy based on the specified TBL criteria preferences. These criteria are used to identify the closest policy among all those trained with EvoDQN, as detailed in Section 4.2. To facilitate understanding of the approach, Fig. 5 illustrates the overall process. Initially, based on a predefined set of preferences (see frame a), multiple agents are created (see frame b). The overall goal is to enable agents to learn the policies that are distributed across the preference space. The agents' preferences are used to scalarize the reward during training. New agents will be created during the evolutionary process (see Evolutionary Operator in frame b) with the objective of maximizing hypervolume. The policies obtained with EvoDQN are then evaluated and normalized (see frame c), which allows the identification of the policy that aligns the most closely with the preference specified by the manager (see frame d). This preference is defined by the manager with the aim of selecting the current interest of the company and balancing between the different objectives. A trained policy will be selected based on how closely the policy behaves with respect to the manager's interest (frame e). The best/selected policy (see frame f) is then designated as the solution to implement in the environment.

Section 4.1 introduces the problem and formalization underlying our DRL system and its interaction. Section 4.2 presents the EvoDQN algorithm, detailing the interactions agents have with the environment.

### 4.1. Problem formulation and modeling

The formalization of the sustainable maintenance scheduling problem is carried out in an environment that involves a set of parallel machines that are repaired by technicians considering TBL. In this regard, Section 4.1.1 describes the environment in which the maintenance scheduling problem exists, outlining the key components and interactions within the system. Following this, Section 4.1.2 formalizes the problem within a DRL framework, capturing the decision-making process required to optimize multiple conflicting objectives simultaneously.

#### 4.1.1. System description

Let us consider a factory where the production is performed by a set of $M$ independent parallel machines as shown in Fig. 6. Each machine $m \in M$ suffers from a degradation modeled by a Gamma process $\Gamma_m$ with shape parameter $\alpha_m$ and scale parameter $\beta_m$. The degradation reduces the efficiency of the production cycle linearly (Arık, 2021), from full efficiency without any degradation to decreasing towards half efficiency. The production cycle stops completely when the machine is not operating (breakdown or maintenance). Machine $m \in M$ fails if the degradation reaches the threshold $\rho_m$. When machine $m$ fails, a breakdown cost $C^B$ is incurred, an additional downtime cost $C^D$ is added for each time step in which the state of machine $\omega_m$ is in a breakdown ($\omega_m = 1$) or maintenance ($\omega_m = 2$). A machine $m$ can receive maintenance by any available technician $t \in T$ to restore the state of the machine (imperfect maintenance action $x^i$), or replacement (perfect maintenance $x^p$) leaving the machine completely restored. When a technician performs maintenance ($\omega_t = 1$), the technician cannot be assigned to any other task until maintenance is completed, after which the technician is available again ($\omega_t = 0$). Each $t \in T$ has different maintenance times $\mathcal{T}_{tm}$ for each machine $m$, which somehow represents the technicians' skills. When a technician performs maintenance, the technician fatigue level increases. In contrast, when the technician is not assigned to any maintenance activity, he/she recovers from fatigue. The fatigue and recovery models described in Jaber et al. (2013) are employed. These are detailed in (1) and (2), respectively.

$$F(\tau^F) = 1 - e^{-\lambda^F \tau^F} \tag{1}$$

$$\mathcal{R}(\tau^{\mathcal{R}}) = F(\tau) e^{-\mu^R \tau_i^{\mathcal{R}}} \tag{2}$$

The fatigue accumulated by time $\tau^F$ is represented by $F(\tau^F)$, while the residual fatigue after a rest period of length $\tau^{\mathcal{R}} \geq 0$ is given by $\mathcal{R}(\tau^{\mathcal{R}})$. The value of $\mathcal{R} = 0$ represents full recovery (no residual fatigue), while $\mathcal{R} = 1$ represents no recovery (maximum fatigue). In (1) and (2), $\lambda^F$ and $\mu^R$ are fatigue and recovery parameters, respectively. These parameters control the speed of fatigue accumulation and recovery relief. This fatigue level increases the time it takes a technician to perform maintenance by $(1+F(\tau^F))$. An example illustrating the different phases of fatigue and recovery for a technician is provided in Fig. 7 based on a scenario of maintenance interventions over time.

In addition, an unexpected breakdown increases the total repair time by $\tau^B \sim \mathcal{N}(\mu_{\text{break}}, \sigma^2_{\text{break}})$. When a technician $t \in T$ performs maintenance, a maintenance cost is incurred depending on the type of maintenance action, $C^P$ for perfect maintenance and $C^I$ for imperfect maintenance.

Each machine $m$ has an energy consumption (kWh) given by $E_m$. This energy consumption is a key factor in reducing the overall carbon footprint of the production process. Due to the fact that the carbon intensity, measured in (gCO2eq/kWh), fluctuates throughout the day, scheduling maintenance activities during periods of high carbon intensity can effectively reduce the total carbon emissions associated with energy consumption. As an example, Fig. 7 shows the different types of signals present in the system and the behavior of different maintenance policies that are adapted based on the TBL. For example, an economic maintenance policy (CM, PM, PdM) can focus on optimizing only the cost that takes into account the maintenance interventions based on the degradation signal (or time-based). A $P_{\text{eco\_soc}}$ can strike a balance between looking for the best maintenance window to reduce the cost and maintaining a low level of fatigue of the technicians to perform better maintenance interventions. A $P_{\text{eco\_env}}$ can decide to perform maintenance interventions to reduce costs while at the same time looking to do it when the carbon intensity is high. Finally, $P_{\text{eco\_hollistic}}$ can look for a balance between all these objectives. The primary objective is to obtain $P_{\text{eco\_hollistic}}$ policies that balance the costs associated with maintenance, machine breakdowns, and downtime with the goal of reducing the carbon footprint of the machines and the fatigue of the technicians.

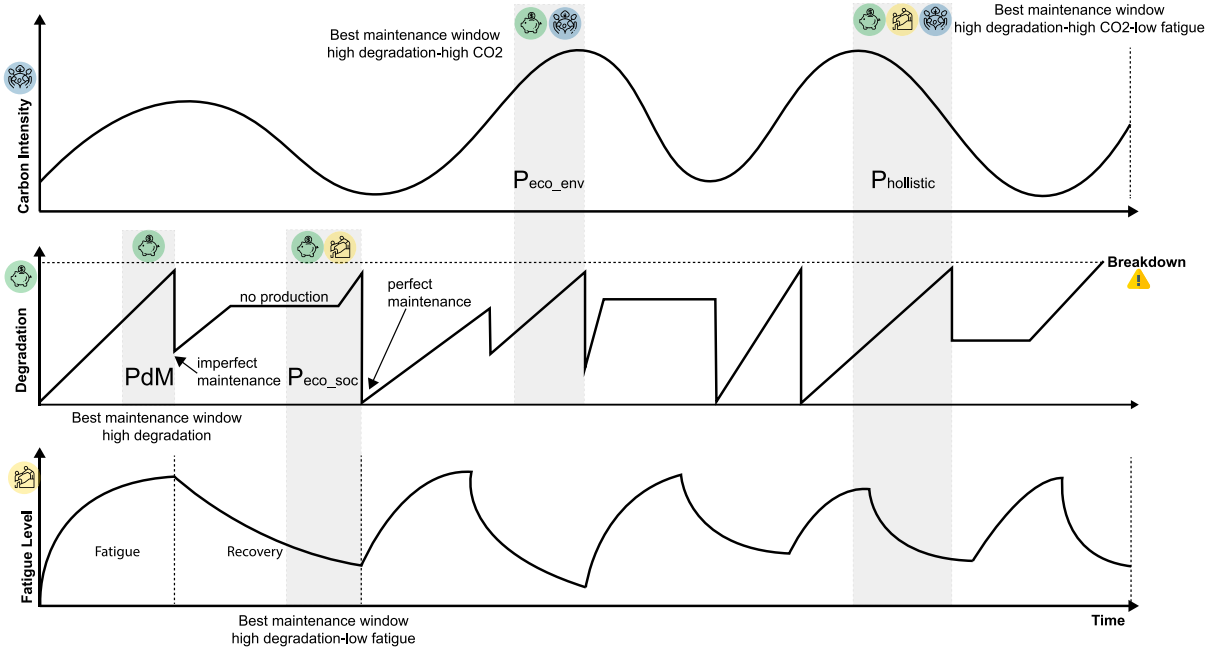



**Fig. 7.** Illustration of system signals, including carbon intensity, degradation, and fatigue, and their influence on identifying optimal maintenance windows based on TBL objectives.

To achieve this, we model the general sustainable maintenance scheduling problem as a multi-objective optimization problem. The objective is to determine an optimal *maintenance scheduling policy* that minimizes:

- **Economic costs** (including maintenance, breakdown, and downtime costs),
- **Environmental impact** (carbon emissions due to energy consumption),
- **Social impact** (technician fatigue and workload balance).

*Decision variables*

Let $x_{m,t,k}$ be a binary decision variable indicating whether machine $m$ is maintained by technician $t$ at time step $k$:

$$x_{m,t,k} = \begin{cases} 1, & \text{if maintenance is performed on machine } m \text{ by technician } t \text{ at time } k, \\ 0, & \text{otherwise.} \end{cases}$$

Similarly, let $x^p_{m,t,k}$ and $x^i_{m,t,k}$ be binary variables for perfect and imperfect maintenance actions, respectively.

*State variables*

- $g_m(k)$: Degradation level of machine $m$ at time step $k$.
- $\rho_m$: Degradation threshold at which machine $m$ fails.
- $\omega_m(k)$: State of machine $m$ at time $k$, where 0 = operational, 1 = breakdown, 2 = under maintenance.
- $CI_k$: Carbon intensity (gCO2eq/kWh) at time step $k$.
- $E_m$: Energy consumption (kWh) of machine $m$.
- $\upsilon_t$: Fatigue level of technician $t$, which increases with maintenance actions and decreases with rest.

*Objective functions*

The multi-objective formulation is:

$$\min_{\mathbf{x}} \quad \left(f_{\text{eco}}(\mathbf{x}),\ f_{\text{env}}(\mathbf{x}),\ f_{\text{soc}}(\mathbf{x})\right)$$

where:

*1. Economic cost function.* The economic cost function represents the total cost associated with maintenance operations. It accounts for:

- **Breakdown costs** ($C^B \cdot \mathbb{1}_{\omega_m=1}$) incurred when a machine fails.
- **Downtime costs** ($C^D \cdot \mathbb{1}_{\omega_m=2}$) due to non-operational machines under maintenance.
- **Perfect maintenance costs** ($C^P \cdot x^p_{m,t,k}$) when a machine is fully restored.
- **Imperfect maintenance costs** ($C^I \cdot x^i_{m,t,k}$) when a machine is only partially restored.

$$f_{\text{eco}}(\mathbf{x}) = \sum_{k\in\mathcal{K}} \sum_{m\in M} \left( C^B \cdot \mathbb{1}_{\omega_m=1} + C^D \cdot \mathbb{1}_{\omega_m=2} + C^P\, x^p_{m,t,k} + C^I\, x^i_{m,t,k} \right). \quad (3)$$

*2. Environmental cost function.* The environmental cost function represents the carbon footprint of the production system by considering:

- **Machine energy consumption** ($E_m$) in kWh.
- **Carbon intensity** ($CI_k$) in gCO2eq/kWh, which varies throughout the day.
- **Operational state of machines** ($\mathbb{1}_{\omega_m=0}$), ensuring emissions are counted only when machines are running.

$$f_{\text{env}}(\mathbf{x}) = \sum_{k\in\mathcal{K}} \sum_{m\in M} \left( E_m \cdot CI_k \cdot \mathbb{1}_{\omega_m=0} \right). \quad (4)$$

*3. Social cost function.* The social cost function represents the fatigue of the technicians and the balance of workload, given by:

- **Minimizing overall fatigue** ($\sum_{t\in T} \upsilon_t$), ensuring technicians remain efficient and well-rested.
- **Balancing workload** by penalizing large fatigue differences between technicians.

The weight parameter $\xi$ controls the balance between fatigue reduction and workload fairness:

- $\xi$ close to 1 prioritizes fatigue minimization.
- $\xi$ close to 0 prioritizes workload balance.

$$f_{\text{soc}}(\mathbf{x}) = \xi \frac{\sum_{t\in T} \upsilon_t}{|T|} + (1-\xi)\frac{\sum_{t\neq t'} |\upsilon_t - \upsilon_{t'}|}{2|T|} \tag{5}$$

*Constraints*

*1. Technician assignment constraint:.*

$$\sum_{m\in M} x_{m,t,k} \leq 1, \quad \forall t \in T, \forall k \in \mathcal{K}.$$

Ensures that each technician can only be assigned to one machine at any given time.

*2. Machine degradation constraint:.*

$$\omega_m(k) = \begin{cases} 1, & \text{if } g_m(k) \geq \rho_m \text{ and } \omega_m(k-1) \neq 2, \\ 2, & \text{if } g_m(k) \geq \rho_m \text{ and } \sum_{t\in T} x_{m,t,k} > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Ensures that a machine enters breakdown ($\omega_m = 1$) if degradation reaches or exceeds $\rho_m$ without scheduled maintenance, enters maintenance ($\omega_m = 2$) if a technician is assigned, and remains operational ($\omega_m = 0$) otherwise.

*3. Maintenance action constraint:.*

$$\begin{aligned} x^p_{m,t,k} + x^i_{m,t,k} &= x_{m,t,k}, \\ x_{m,t,k} &\in \{0,1\}, \\ x^p_{m,t,k},\ x^i_{m,t,k} &\in \{0,1\}. \end{aligned}$$

Ensures that maintenance can be either perfect ($x^p_{m,t,k}$) or imperfect ($x^i_{m,t,k}$).

*4. Machine state constraint:.*

$$\mathbb{1}_{\omega_m=0} + \mathbb{1}_{\omega_m=1} + \mathbb{1}_{\omega_m=2} = 1, \quad \forall m \in M,\ k \in \mathcal{K}.$$

Ensures that each machine is always in exactly one state (operational, breakdown, or maintenance).

In the next section, we present a detailed formulation of the sustainable maintenance scheduling problem by integrating its economic, environmental, and social objectives into a sequential decision-making framework modeled as a Multi-objective Markov Decision Process, which provides the basis for our RL approach.

#### 4.1.2. Multi-objective Markov decision process

To obtain optimal or near-optimal maintenance policies, it is necessary for decision-making agents to learn how to explore the preference space of the multiple objectives in an environment whose dynamics are initially unknown (Sutton and Barto, 2018). This environment can be described by a Multi-Objective Markov Decision Process (MOMDP). MOMDP is a mathematical formulation of a problem in which an agent (decision-maker) selects actions sequentially to transit through different states guided by rewards. A MOMDP can be expressed as a 5-tuple $\langle S, A, \mathcal{P}, R, \gamma \rangle$, which consists of (i) a state space $S$ indicating all possible states the agent can be in; an action space $A$ indicating all possible actions the agent can take; (ii) a transition function $\mathcal{P} : S \times A \to S$ indicating the probability of transitioning from any state $s \in S$ to state $s' \in S$ given that the agent took action $a \in A$; (iii) a reward function $R : S \times A \times S \to \mathbb{R}^N$ that returns an immediate reward vector given by the transition from $(s, a)$ to $s'$; and (iv) a discount factor $\gamma \in [0, 1]$ indicating how myopic the agent is, where $\gamma = 0$ indicates that the agent only cares about immediate reward and $\gamma \to 1$ indicates it gives more weight to future state information.

The proposed sustainable maintenance policies aim to optimize three objectives: (i) minimize the maintenance, breakdown, and downtime costs; (ii) minimize the fatigue level of the technicians after performing maintenance actions; (iii) minimize the carbon footprint. Using the MOMDP framework, agents determine their actions based on the observations made by the system. Therefore, it is required to design local observations that provide the basis for the agent's actions and provide useful information for training.

The system state observed by the agent is represented by the vector given in (6), where $g_m^{\text{Norm}}\ \forall m \in M$ gives information about the normalized machine degradation, $q_t^{\text{Norm}}\ \forall t \in T$ is the normalized remaining time of each technician performing maintenance, $CI^{\text{forecast}}$ the normalized carbon footprint forecast, $\upsilon_t\ \forall t \in T$ the fatigue level of the technicians, $O(\omega_m)\ \forall m \in M$ the one hot encoding of the state of each machine, and $O(\omega_t)\ \forall t \in T$ the one hot encoding of the state of each technician.

$$\mathbf{o} = (g_m^{\text{Norm}}, q_t^{\text{Norm}}, CI^{\text{forecast}}, \upsilon_t, O(\omega_m), O(\omega_t)) \tag{6}$$

Depending on the observation, the agent decides at each time step which type of maintenance operation needs to be performed. Eight rules are defined based on the selection of the machine, technician, and type of maintenance, as detailed hereinafter:

- **Machine selection** based on Highest Degradation Level ($m_{HDL}$), and Highest Cycle Time ($m_{HCT}$)
- **Technician selection** based on Technician with the minimum level of fatigue ($t_{mF}$) or maximum skills ($t_{MS}$).
- **Maintenance operation** based on the type of maintenance, perfect maintenance ($x^p$), where the component is replace, or imperfect maintenance ($x^i$) where the state of a component is partially recovered.

The actions taken by the agent is given in (7). In addition, the policy can decide not to take any action (i.e. no technician will be assigned to perform maintenance) represented by $d$. The eight dispatching rules are detailed and explained in Table 2, and together with the "no-action" option there is a total of nine possible actions that the agent can take, as indicated by $|\mathbf{a}| = 9$.

$$\mathbf{a} \in \{M_{HDL}, M_{HCT}\} \times \{T_{mF}, T_{MS}\} \times \{x^p, x^i\} \cup \{d\} \tag{7}$$

To guide agents in learning the policy, the reward function is defined as the negative of the objective function defined in 4.1.1:

$$\mathbf{r} = -\mathbf{f}(\mathbf{x})$$

where:

$$\mathbf{r} = -\mathbf{f}(\mathbf{x})$$

$$\mathbf{r} = [r_{\text{eco}}, r_{\text{env}}, r_{\text{soc}}]$$

Thus, each element of the reward is the negative of the objectives defined in (3),(4),(5):

$$\begin{aligned} r_{eco} &= -f_{\text{eco}}(\mathbf{x}), \\ r_{env} &= -f_{\text{env}}(\mathbf{x}), \\ r_{soc} &= -f_{\text{soc}}(\mathbf{x}). \end{aligned}$$

This aligns with the MOMDP framework to maximize the rewards, and the sustainable maintenance scheduling formulation to minimize the objectives.

The transition function $\mathcal{P}$ captures the dynamics of the system, including the stochastic degradation of machines, CO2 forecast, and the relationship between production and maintenance. For a particular action $a$, given a state $s$, the next state $s'$ is determined as follows:

- Machine degradation following a $\Gamma$ process and the degradation is only restored by maintenance actions $x^p$ and $x^i$. A perfect maintenance action restores the machine completely. An imperfect maintenance action restores partially the original state of the machine.
- Remaining technician time decreases for each timestep and is normalized by the maximum repair time.

**Table 2**
The eight dispatching rules applied by MOMDP agents, combining machine state, technician condition, and maintenance type (perfect or imperfect).

| Rule | | Description |
|---|---|---|
| 1 | $M_{HDL} \times T_{mF} \times x^p$ | Perform a perfect maintenance action by the technician with the lowest fatigue level on the machine with the highest degradation level. |
| 2 | $M_{HDL} \times T_{mF} \times x^i$ | Perform an imperfect maintenance action by the technician with the lowest fatigue level on the machine with the highest degradation level. |
| 3 | $M_{HDL} \times T_{MS} \times x^p$ | Perform a perfect maintenance action by the best performance technician on the machine with the highest degradation level. |
| 4 | $M_{HDL} \times T_{MS} \times x^i$ | Perform an imperfect maintenance action by the best performance technician on the machine with the highest degradation level. |
| 5 | $M_{HCT} \times T_{mF} \times x^p$ | Perform a perfect maintenance action by the technician with the lowest fatigue level on the machine with the highest cycle time. |
| 6 | $M_{HCT} \times T_{mF} \times x^i$ | Perform an imperfect maintenance action by the technician with the lowest fatigue level on the machine with the highest cycle time. |
| 7 | $M_{HCT} \times T_{MS} \times x^p$ | Perform a perfect maintenance action by the best performance technician on the machine with the highest cycle time. |
| 8 | $M_{HCT} \times T_{MS} \times x^i$ | Perform an imperfect maintenance action by the best performance technician with the lowest fatigue level on the machine with the highest cycle time. |

- Forecast of CO2 is determined by the hourly data of the carbon intensity of Luxembourg from Electricity Maps (2024). The forecast is obtained by taking the values from the current step $k$ to $k + \Delta k$, where $\Delta k$ represents the number of forecast steps. The predicted carbon intensity, denoted as $CI_k^{\text{forecast}}$, is computed by adding a noise component $CI_k^{\text{noise}}$ to the actual carbon intensity $CI_k$. The noise component is generated using a linearly spaced vector ranging from 0 to 1 over $\Delta k$ steps, scaled by random values drawn from a normal distribution with mean $\mu_{\text{car}}$ and standard deviation $\sigma_{\text{car}}$:
- Fatigue level of each technician increases on maintenance interventions while decreases on the resting periods following the fatigue and recovery models formalized in (1) and (2).

Overall, the objective is to find policies that maximize the hypervolume.

### 4.2. EvoDQN algorithm

Combining evolutionary algorithms with ML, such as DRL, has been successful in many applications (Drugan, 2019), notably for dynamic scheduling problem solving (Chen et al., 2020; Köksal Ahmed et al., 2022; Song et al., 2023; Fu et al., 2022). Building upon this evidence, this research develops a novel Evolutionary multi-objective multi-agent DQN (EvoDQN), detailed in Algorithm 1, to generate sustainable policies that balance the different objectives of the TBL. DRL is responsible for optimizing the policies, while evolutionary computation generates new policies in the preference space, the goal being to maximize the hypervolume of these policies by training a limited set of policies in each generation.

In a more technical way, EvoDQN distributes and optimizes multiple agents across different initial preferences, as was sketched schematically in Fig. 5. The initial set of policies then undergoes an evolutionary process in which new agents are generated through an evolutionary operator with the goal of maximizing hypervolume. According to the methodology proposed by Bodnar et al. (2020), EvoDQN combines agents that are closely aligned in terms of preferences. These agents are trained based on DQN (Mnih et al., 2013), where the reward vector is scalarized based on the preference vector. Once the agents have been trained, their selection is based on two functions. The first function, `Pareto_Agents`, identifies agents that form the Pareto front based on their multi-objective cumulative reward. Each agent is associated

**Algorithm 1:** Evolutionary multi-objective multi-agent Deep Q-Network (EvoDQN)

**Input:** *num_generations*, *initial_preferences*, *num_steps*, *num_policies*, *mutation_rate*, *crossover_rate*
**Initialization Phase**;
Initialize population of agents $P = \emptyset$;
**for** ***each*** *preference vector* $\mathbf{d}_i$ *in* *initial_preferences* **do**
  Initialize agent $i$ with:
    - replay memory $D_i$,
    - action-value function $Q_{\theta_i}$ with random weights $\theta_i$,
    - target action-value function $Q_{\theta_i'}$ with $\theta_i' = \theta_i$,
    - preference vector $\mathbf{d}_i$.
  Add $i$ to $P$;
**for** $g \leftarrow 1$ **to** *num_generations* **do**
  **Evolutionary Phase**;
  **for** ***each*** *agent* $i$ *in* $P$ **do**
    Find closest neighbor agent $k$ (by preference vector);
    **if** $g > 1$ **then**
      $i^*, k^* \leftarrow$ Evolutionary_Operator$(i, k,$ *mutation_rate*, *crossover_rate*$)$;
      Update $P \leftarrow P \cup \{i^*, k^*\}$;
  **Learning Phase**;
  **for** *step* $\leftarrow 1$ **to** *num_steps* **do**
    **for** ***each*** *agent* $i$ *in* $P$ **do**
      Choose action $a$ from state $s$ using an $\epsilon$-greedy policy;
      Take action $a$, observe reward $\mathbf{r}$ and next state $s'$;
      Compute scalar reward $r' \leftarrow \mathbf{r} \cdot \mathbf{d}_i$;
      Store transition $(s, a, r', s')$ in $D_i$;
      Sample minibatch $(s_j, a_j, r_j', s_j')$ from $D_i$;
      $$y_j = \begin{cases} r_j', & \text{if terminal,} \\ r_j' + \gamma \max_{a'} Q_{\theta_i'}(s_j', a'), & \text{otherwise.} \end{cases}$$
      Perform a gradient descent step on the loss $(y_j - Q_{\theta_i}(s_j, a_j))$ with respect to $\theta_i$;
      Update $s \leftarrow s'$;
  **Selection Phase**;
  $P \leftarrow$ Pareto_Agents$(P)$;
  $P \leftarrow$ Hypervolumen_Contrib$(P,$ *num_policies*$)$;

**Algorithm 2:** Evolutionary_Operator

**Input:** $a_1$ /*Agent 1*/
$a_2$ /*Agent 2*/
$a_1^* \leftarrow a_1$;
$a_2^* \leftarrow a_2$;
/*Crossover through the replay buffer*/;
Let N be the length of the replay buffer;
$rb\_co \sim \mathcal{U}(0.05, 0.95)$;
$co\_idx \leftarrow 1 - crossover\_rate \cdot rb\_co$;
$rb_{11} \leftarrow a_1.replay\_buffer(0, co\_idx)$;
$rb_{21} \leftarrow a_2.replay\_buffer(0, co\_idx)$;
$rb_{12} \leftarrow a_1.replay\_buffer(co\_idx, N)$;
$rb_{22} \leftarrow a_2.replay\_buffer(co\_idx, N)$;
$a_1^*.replay\_buffer \leftarrow \text{merge}(rb_{11}, rb_{22})$;
$a_2^*.replay\_buffer \leftarrow \text{merge}(rb_{21}, rb_{12})$;
/*Mutation through the weights and preferences*/;
$pv\_mu \sim \mathcal{U}(0.05, 0.95)$;
$mr \leftarrow 1 - mutation\_rate \cdot pv\_mu$;
$a_1^*.pref = (mr) \cdot a_1.pref + (1 - mr) \cdot a_2.pref$;
$a_2^*.pref = (1 - mr) \cdot a_1.pref + mr \cdot a_2.pref$;
$\epsilon_1, \epsilon_2 \sim \mathcal{N}(\mu, perturbation)$;
$a_1^*.policy\_weights = a_1.policy\_weights + \epsilon_1$;
$a_1^*.target\_weights \leftarrow a_1.policy\_weights$;
$a_2^*.policy\_weights = a_2.policy\_weights + \epsilon_2$;
$a_2^*.target\_weights \leftarrow a_2.policy\_weights$;
**return** $a_1^*, a_2^*$;

with a cumulative reward vector that represents its performance in several objectives. An agent is considered non-dominated if no other agent has a cumulative reward vector that is better or equal in every objective and strictly better in at least one objective. The second is the `Hypervolumen_Contrib` function that selects the top *num_policies* agents from the set $A = \{a_1, a_2, \ldots, a_n\}$, which are the agents that contribute the most to the overall hypervolume. First, the total hypervolume is computed as $HV(A)$. Then, for each agent $a_i \in A$, the hypervolume of the set without that agent is calculated as $HV(A \backslash \{a_i\})$. The contribution of agent $a_i$ is defined by

$$\Delta_i = HV(A) - HV(A \setminus \{a_i\}),$$

which represents the decrease in hypervolume if $a_i$ is removed. The agents are then ranked such that

$$\Delta_{i_1} \geq \Delta_{i_2} \geq \cdots \geq \Delta_{i_n},$$

and the top *num_policies* agents (those with the highest $\Delta_i$ values) are selected.

The evolutionary operator, detailed in Algorithm 2, is applied to each agent with its nearest neighbor in the preference space to generate two new child agents. This operator performs three key functions:

1. **Replay Buffer Crossover:** Each child's replay buffer is formed by merging a portion of its parent's experiences with a complementary portion from its neighbor. This ensures that the child inherits relevant learning experiences from both agents.
2. **Preference Vector Blending:** The child's preference vector is updated by computing a weighted linear combination of the parent's and the neighbor's preference vectors. The resulting vector is then normalized so that its components sum to one. This blending effectively shifts the child's preference towards that of its neighbor, promoting the exploration of intermediate trade-offs in the multi-objective space.
3. **Policy Weight Mutation:** The child's policy network is initially set by copying its parent's network parameters. Subsequently, a small Gaussian noise is added to the weights to introduce diversity into the policy. Optionally, the last layer of the network may be reinitialized to further enhance exploration.

**Table 3**
System evaluation parameters used in the simulation.

| Category | Parameter |
|---|---|
| Machine Parameters | |
| Number of machines | 3 |
| Energy consumption/m (kWh) | 100.0 |
| Gamma shape parameter | 2.0 |
| Gamma scale parameter | 1.0 |
| Failure threshold | 20.0 |
| Technician Parameters | |
| Number of technicians | 2 |
| Maintenance time for T1 | 6 |
| Maintenance time for T2 | 3 |
| Workload balance | 0.5 |
| Cost Parameters | |
| Perfect maintenance/m | 0.5k |
| Imperfect maintenance/m | 0.1k |
| Downtime | 1.0k |
| Breakdown | 10.0k |
| Forecast CO2 Parameters | |
| Steps | 10 |
| Mean noise | 0.0 |
| Std noise | 0.1 |

Overall, the evolutionary operator encourages exploration in both policy and preference spaces, thereby facilitating the development of more diverse and effective maintenance policies. To simplify understanding, an overview of how the evolutionary operator is presented in Fig. 8.

## 5. Implementation & evaluation

In the following, EvoDQN is compared with classical maintenance policies and DRL policies based on DQN and PPO. Section 5.1 introduces the experimental setup used for evaluation. Section 5.2 details the methodology employed to train EvoDQN and their baseline (cf. Fig. 5 to visualize where this step stands). The results obtained between the different policies and their sustainable implications are discussed in Section 5.3.

### 5.1. Evaluation scenario and setup parameters

The evaluation was conducted in a three-machine, two-technician setting. Each machine has an energy consumption of $E_m = 100.0$ kWh $\forall m \in M$. The gamma degradation model is defined with a shape parameter of $\alpha_m = 2.0$ and a scale parameter of $\beta_m = 1.0$ on all machines. The failure threshold is set to $\rho_m = 20.0$ $\forall m \in M$. The maintenance time of the two technicians are defined as $\mathcal{T}_{t=1,m} = 6$ and $\mathcal{T}_{t=2,m} = 3$ $\forall m \in M$ units of time to complete any intervention in a complete rest state. The parameters for the fatigue and recovery model presented in (1) and (2) are $\lambda^F = 0.03$ and $\mu^R = .15$. The cost of perfect maintenance is $C^P = 500.0$ units per machine, while imperfect maintenance costs $C^I = 100.0$ units per machine. The downtime cost is $C^D = 1.0k$ units and the cost of a machine breakdown is significantly higher at $C^B = 10.0k$ units. We obtained information on the carbon intensity in Luxembourg in 2022 from the Data Portal of Electricity Maps. The carbon intensity represents the amount of greenhouse gases emitted per unit of electricity consumed, measured in grams of CO2 equivalent per kilowatt-hour (gCO2eq/kWh). The carbon intensity forecast is computed over a horizon of $\Delta k = 10$ time steps, incorporating a noise component modeled as a normally distributed random variable with mean $\mu_{\text{car}} = 0.0$ and standard deviation $\sigma_{\text{car}} = 0.1$. A summary of the overall experimental configuration is provided in Table 3.

The evaluation of maintenance policies is based on three pillars of sustainability: economic, social, and environmental. The economic pillar considers maintenance costs, breakdown costs, and downtime costs. The social pillar assesses the fatigue level of technicians performing

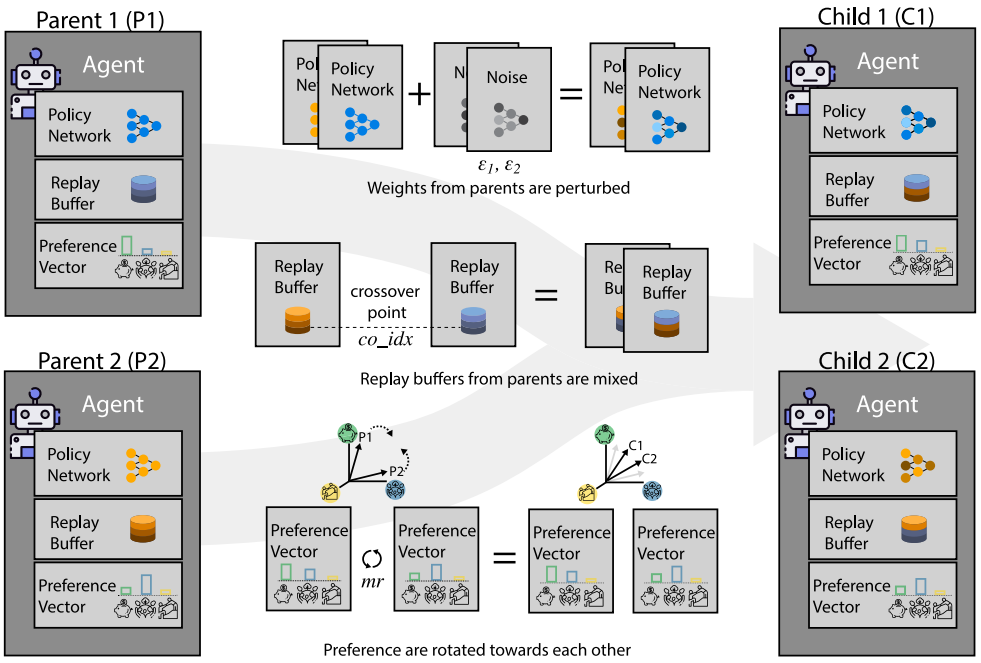



**Fig. 8.** Overview of the Evolutionary Operator. Child agents inherit perturbed policy networks, mixed replay buffers, and rotated preference vectors from two parent agents. Variables $\epsilon_1$, $\epsilon_2$, $co\_idx$, $mr$ are referenced from Algorithm 2.

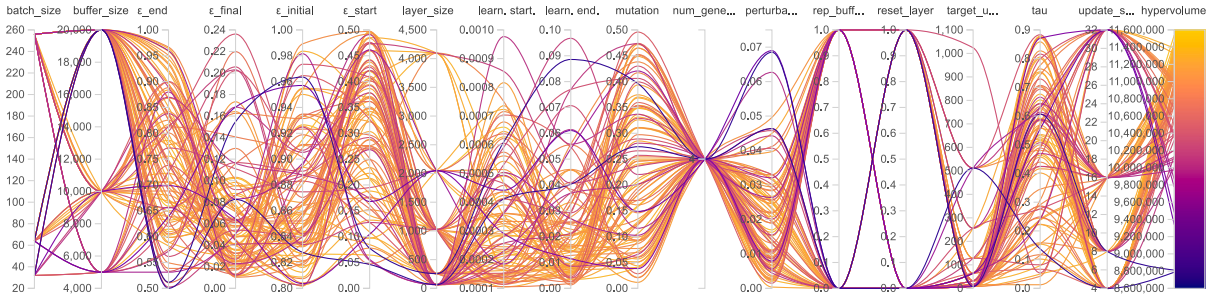



**Fig. 9.** Parallel coordinate plot showing the relationship between hyperparameter settings and their corresponding hypervolume performance for the EvoDQN.

maintenance. The environmental pillar measures the carbon footprint on the production of machines.

The policies obtained by EvoDQN are compared against two classical maintenance policies: CM, and CBM. These policies are described in the following.

- The CM policy performs perfect maintenance whenever the machine suffers a degradation that causes the equipment to go into a breakdown state. As half of the defined dispatching rules perform perfect maintenance (Rule 1,3,5,7 in Table 2), the CM policy only executes those rules;
- CBM policy performs maintenance activities by executing one of the eight dispatching rules when a machine reaches $\rho_m$.

CM only executes four dispatching rules since it needs to perform perfect maintenance (replacement) when the component breaks. On the other hand, CBM executes a dispatching rule when a particular threshold is reached on a machine. We evaluated 99 different thresholds for CBM (from 1 to 99). In total, 8 policies were obtained for CM and 792 policies for CBM (99 thresholds × 8 rules).

### 5.2. Training of evolutionary DRL agents

For training the EvoDQN models, a fully connected neural network with one hidden layer and ReLU activation was used. A Bayesian optimization hyperparameter search was conducted using Weights & Biases (wandb) sweeps. The search consisted of a single phase of 100k environment steps over four generations. After optimization, the selected hyperparameters were used to train a new model for 100k environment steps in 10 generations. Training was performed on a system equipped with an Intel Core i7-14700K processor, 96 GB of RAM, and an NVIDIA GeForce RTX 4080 SUPER GPU, ensuring efficient computation. Detailed hyperparameter ranges and settings used in this sweep are summarized in Table 4.

To provide further insight into the hyperparameter optimization process for EvoDQN, we present a parallel coordinate plot (Fig. 9) that illustrates the relationships and trade-offs between the hyperparameter settings and their corresponding impact on the optimization metric.

The set of initial preferences is the one reported in Table 5. Each of the agents receives one of these preferences that scalarizes the reward vector while being trained (cf., Algorithm 1). Following this, new agents are created through the evolutionary process (cf. Algorithm 2). The configuration of the sweep parameters and the best parameters obtained are detailed in Table 4, where the best parameters were those that maximized the hypervolume with a maximum selection of eight agents over generations.

After obtaining the parameters that maximize hypervolume, four more models were trained to evaluate the spread of the solutions, as given in Fig. 10. The different models are described below.

**Table 4**
Summary of selected hyperparameters including their search ranges.

| Hyperparameter | Search range | Selected value |
|---|---|---|
| Batch Size | {16, 32, 64, 256} | 32 |
| Buffer Size | {5000, 10000, 20000} | 10,000 |
| Epsilon End | [0.5, 1.0] | 0.908 |
| Epsilon Final | [0.01, 0.25] | 0.191 |
| Epsilon Initial | [0.8, 0.95] | 0.873 |
| Epsilon Start | [0.0, 0.5] | 0.255 |
| Layer Size | {64, 256, 1024, 2048} | 1,024 |
| Learning Rate End | [0.0001, 0.001] | 0.000384 |
| Learning Rate Start | [0.001, 0.1] | 0.0323 |
| Mutation Rate | [0.05, 0.5] | 0.407 |
| Number of Generations | {4} | 10 (For training) |
| Perturbation | [1e−05, 0.1] | 0.0301 |
| Replay Buffer Strategy | {0, 1} | 1 |
| Reset Layer | {0, 1} | 0 |
| Target Update Steps | {32, 64, 128, 256} | 32 |
| Tau | [0.01, 0.1] | 0.0552 |
| Update Steps | {4, 8, 16, 32} | 16 |

**Table 5**
Matrix of initial preferences with assigned weights for economic (Eco), environmental (Env), and social (Soc) objectives.

| | Weights | | |
|---|---|---|---|
| | Eco | Env | Soc |
| Preference 1 | 0.25 | 0.5 | 0.25 |
| Preference 2 | 0.00 | 0.75 | 0.25 |
| Preference 3 | 0.00 | 1.00 | 0.00 |
| Preference 4 | 0.25 | 0.00 | 0.75 |
| Preference 5 | 0.00 | 0.25 | 0.75 |
| Preference 6 | 1.00 | 0.00 | 0.00 |
| Preference 7 | 0.75 | 0.00 | 0.25 |
| Preference 8 | 0.50 | 0.25 | 0.25 |
| Preference 9 | 0.25 | 0.75 | 0.00 |
| Preference 10 | 0.25 | 0.25 | 0.50 |
| Preference 11 | 0.75 | 0.25 | 0.00 |
| Preference 12 | 0.50 | 0.50 | 0.00 |
| Preference 13 | 0.00 | 0.00 | 1.00 |
| Preference 14 | 0.00 | 0.50 | 0.50 |
| Preference 15 | 0.50 | 0.00 | 0.50 |

- **EvoDQN8**: EvoDQN employs a selection process of 8 agents per generation, each trained for $100k$ steps using the initial preferences from Table 5. Total number of steps taken were $4.3M$;
- **EvoDQN16**: EvoDQN with 16 agents, each being trained for $100k$ steps with initial preferences from Table 5;
- **EvoDQN32**: EvoDQN with 32 agents, each being trained for $100k$ steps with initial preferences from Table 5.
- **DQN**: DQN with 15 agents based on the preferences from Table 5, each being trained for $286K$ steps with initial preferences from Table 5 for a total of $4.3M$ steps.
- **PPO**: PPO with 15 agents, based on the preferences from Table 5, each being trained for $286K$ steps with initial preferences from Table 5 for a total of $4.3M$ steps.

EvoDQN is compared with vanilla DQN and PPO (both implemented using the Stable-Baselines3 library). These methods are trained using the initial preferences outlined in Table 5, and a detailed description of the training setup and hyperparameter optimization process is provided in Appendix B. By relying on a single set of preferences to scalarize the reward, DQN and PPO inherently limit their ability to explore multiple objectives. Furthermore, we evaluated the hypervolume achieved by EvoDQN16 (16 agents) and EvoDQN32 (32 agents) using the same set of initial preferences. As illustrated in Fig. 10, EvoDQN discovers policies that are more evenly distributed across the preference space, with a hypervolume of 11442566.59 resulting in ≈ 11.6% higher than DQN and ≈ 18.9% PPO.

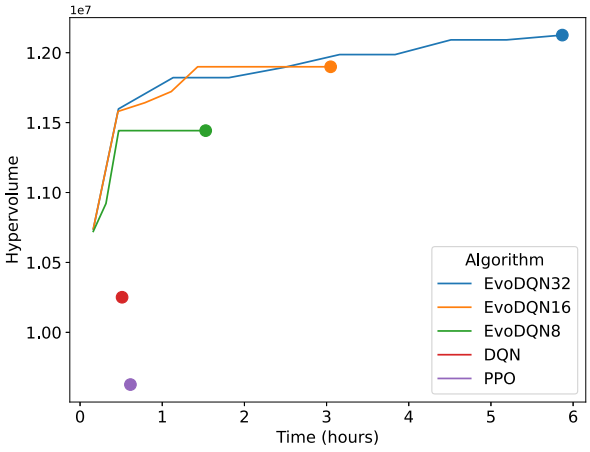



**Fig. 10.** Hypervolume over time comparing EvoDQN variants (32, 16, 8 agents) with baseline DQN and PPO algorithms.

### 5.3. Maintenance policy comparison analysis

As previously mentioned in Section 5.1, EvoDQN with eight policies is compared with the CBM and CM policies, where each policy represents a different dispatching rule. We selected CBM with a threshold of 77 (denoted by CBM77) as the set of policies to evaluate because it demonstrated the highest level of effectiveness compared to the other thresholds in terms of hypervolume. Although these policies are primarily focused on the economic aspect, we are interested in evaluating the impact on the other pillars. Before comparing those policies, we believe that it is important to provide insight into what one EvoDQN policy implies on the behavior of the system and technician over a period of time. In this regard, Section 5.3.1 showcases the sustainable implication for top-profit policy (P1) of EvoDQN. Thereafter, the comparison results are discussed in Section 5.3.2. To further evaluate the robustness of our approach in more complex settings, we conduct additional experiments presented in Appendix C, where the number of machines is scaled up to 50. This analysis offers valuable insights into EvoDQN's performance under more demanding and constrained real-world conditions.

#### 5.3.1. Sustainable implication of EvoDQN top-profit policy

Economic policies are effective due to the straightforward relationship between profit maximization and cost minimization. However, the inclusion of sustainable policies directly impacts the economic objective. For example, identifying windows of opportunity where the carbon footprint is high to perform maintenance interventions may benefit the environment, but it may also affect production and maintenance by performing interventions far from the point of failure (too early or too late). Similarly, reducing fatigue and distributing maintenance tasks can affect the selection of technicians and the timing of the intervention. In practice, only a subset of policies that cover the entire preference space is useful. For example, policies that prioritize sustainability over economic considerations may only perform maintenance when the carbon footprint is high. Extreme cases in the preference space may prevent machines from continuing in production to minimize the carbon footprint. On the other hand, the suggested socially beneficial policies would prevent technicians from experiencing high levels of fatigue by prioritizing rest over interventions. In the most extreme cases, technicians would never perform maintenance activities to avoid increasing fatigue. Fig. 11 presents a subset of the temporal data of a policy where the three objectives are balanced. The first three temporal data graphs (starting at the top) represent the degradation of the machines where the breakdown point is when the degradation reaches 1.0. The following two graphs illustrate the fatigue level of the

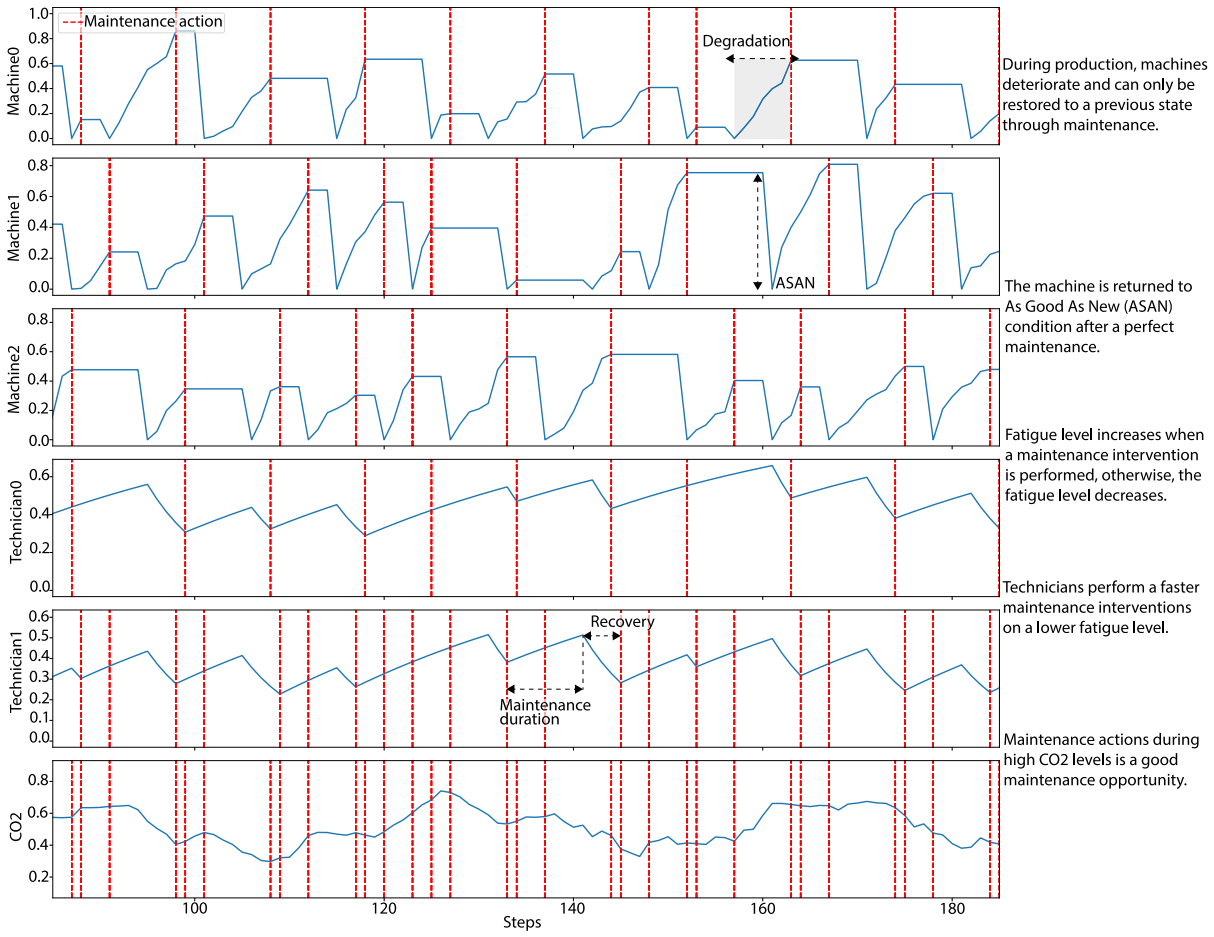


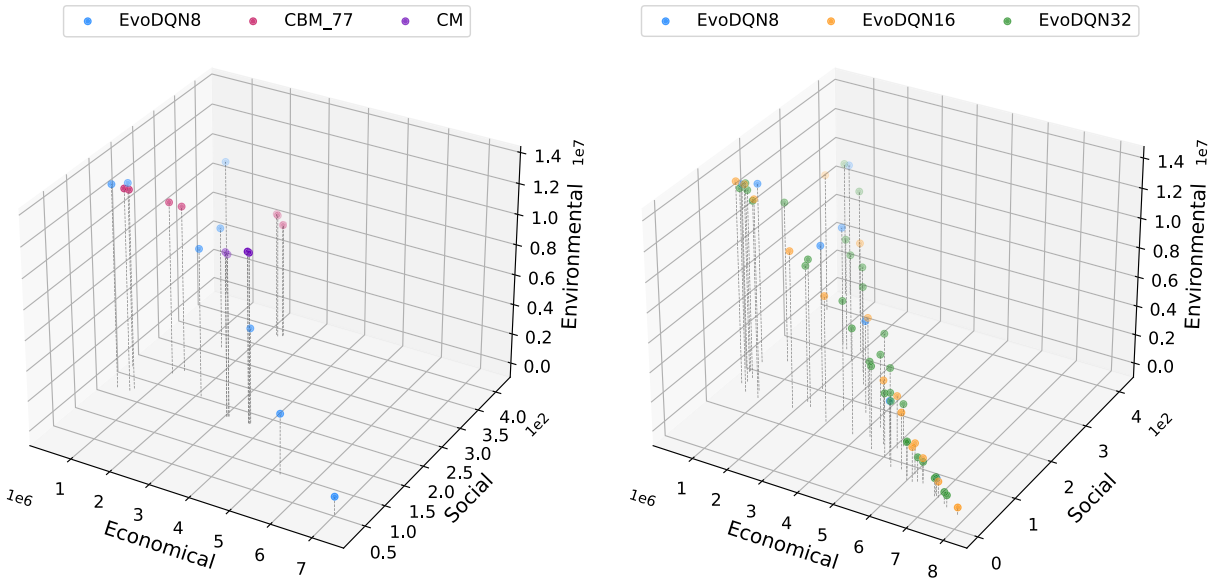

**Fig. 11.** Machine degradation, technician fatigue, and carbon intensity over time, with dashed red lines indicating maintenance actions and their impact on system dynamics.

two technicians involved in the maintenance process, the value of 0 and 1 representing respectively complete recovery and fatigue (cf., Eq. (1) and (2) for further details). Finally, the last graph (at the bottom) represents the carbon footprint signal of the overall manufacturing process.

### 5.3.2. Comparison results

The results illustrating the trade-off between economic, social, and environmental objectives for the EvoDQN, CBM77, and CM maintenance policies are shown in Fig. 12(a), while Fig. 12(b) presents how this trade-off evolves with an increasing number of EvoDQN policies, particularly for 8, 16, and 32 policies. As presented in Fig. 12(a), the lower the score on all axes, the better the policy. It can be seen that the EvoDQN policies clearly illustrate a trade-off between the economic, environmental, and social objectives. It is evident that as social and environmental costs increase, the economic cost decreases. In particular, we can observe a set of policies in the lower part of the economic axis that exhibit a slight trade-off between the other two objectives. From the entire set of EvoDQN policies, a subset has been identified as feasible for implementation in an industrial setting. This is because policies that significantly reduce carbon footprint values and technician fatigue tend to avoid maintenance activities to minimize fatigue, or even, in the most extreme cases, to keep machines in a state of downtime (for maintenance or breakdown) so that the carbon footprint is minimized. This kind of policy is, of course, not a viable option in real-world industrial scenarios. In the case of CBM77, certain policies demonstrate distinctive behavior, some exhibiting competitive economic results comparable to those of the EvoDQN policies. In contrast, CM policies exhibit a consistent, centralized cluster with similar behavior (see Table 6). As shown in Fig. 12(b), increasing the number of EvoDQN policies from 8 to 16 and 32 results in a denser and more diverse set of trade-offs between economic, social, and environmental objectives. The additional policies generated by EvoDQN16 and EvoDQN32 reveal a broader spectrum of trade-offs, making it easier to visualize how different preferences can balance these objectives. Compared to EvoDQN8, which provides a limited set of solutions, the policies obtained from EvoDQN16 and EvoDQN32 more clearly demonstrate how certain policies prioritize economic efficiency, while others focus on social and environmental improvements. This expanded solution space offers decision makers a more comprehensive range of policies that can be selected from, depending on their priorities. The results further confirm that, while some policies achieve a lower economic cost, they often come at the expense of higher environmental and social impacts. By increasing the number of policies, more detailed information is achieved on these trade-offs, enabling a more precise selection of maintenance strategies.

For a deeper analysis, the results obtained from the eight EvoDQN policies, eight CBM77, and four CM policies are presented with their 2D projection in Fig. 13, together with a more in-depth breakdown of the economic pillar. The first three columns represent the projection of the trade-off between two objectives, the first column representing Economic-Social, the second column representing Economic-Environmental, and the third one representing Social-Environmental. As the CM and CBM policies are rule-based, which EvoDQN is able to select depending on the state of the system, we present the two best rules of the CM and CBM from the different projections in the first three columns and the best result of each policy for the last column. On the EvoDQN side, we present in each row the eight different policies found.




(a) EvoDQN vs. Classic maintenance policies (b) EvoDQN variants comparison

**Fig. 12.** Comparison of EvoDQN policies. (a) EvoDQN, CBM77, and CM policies for the TBL. (b) EvoDQN across different number of policies.

**Table 6**
Performance metrics (mean and standard deviation) for various policies.

| Policy | Total cost | | Fatigue | | CO2 | |
|---|---|---|---|---|---|---|
| | Mean | Std | Mean | Std | Mean | Std |
| EvoDQN8 P1 | 431 050.00 | 97 949.00 | 192.11 | 7.50 | 12239464.70 | 172 603.09 |
| EvoDQN8 P2 | 578 456.67 | 115 496.33 | 413.67 | 4.98 | 9106073.53 | 100 828.19 |
| EvoDQN8 P3 | 617 750.00 | 225 047.47 | 139.28 | 3.44 | 13399132.53 | 301 014.62 |
| EvoDQN8 P4 | 1736313.33 | 1082474.24 | 289.77 | 59.90 | 8094301.40 | 1531255.37 |
| EvoDQN8 P5 | 2428280.00 | 334 505.79 | 173.64 | 8.75 | 9799308.67 | 371 258.24 |
| EvoDQN8 P6 | 3936030.00 | 570 318.53 | 151.19 | 27.14 | 6279504.77 | 818 018.48 |
| EvoDQN8 P7 | 5513950.00 | 275 351.03 | 68.32 | 6.02 | 3992640.37 | 655 529.94 |
| EvoDQN8 P8 | 7255126.67 | 559 136.51 | 25.28 | 16.98 | 1180830.77 | 524 339.03 |
| CBM77 R1 | 969 090.00 | 220 347.24 | 147.33 | 15.26 | 13124996.92 | 527 897.96 |
| CBM77 R2 | 2547168.00 | 214 518.61 | 352.91 | 19.01 | 8248932.42 | 328 836.38 |
| CBM77 R3 | 930 810.00 | 204 269.26 | 140.21 | 14.04 | 13326043.14 | 485 182.85 |
| CBM77 R4 | 2600908.00 | 197 973.47 | 350.42 | 20.71 | 8246078.68 | 285 789.40 |
| CBM77 R5 | 1365580.00 | 275 681.47 | 202.72 | 21.43 | 11413359.98 | 686 179.06 |
| CBM77 R6 | 2692006.00 | 202 550.53 | 354.06 | 22.65 | 7586123.62 | 404 762.00 |
| CBM77 R7 | 1543310.00 | 289 439.75 | 215.63 | 30.50 | 10988890.86 | 908 023.59 |
| CBM77 R8 | 2693018.00 | 251 266.04 | 355.77 | 18.95 | 7570451.52 | 380 278.95 |
| CM R1 | 3940120.00 | 172 669.90 | 148.68 | 13.83 | 11208915.06 | 373 082.92 |
| CM R3 | 3901730.00 | 172 146.63 | 148.85 | 14.18 | 11253092.26 | 436 529.93 |
| CM R5 | 3391030.00 | 207 082.07 | 150.18 | 12.11 | 10667375.66 | 486 925.62 |
| CM R7 | 3355220.00 | 238 993.52 | 148.59 | 17.59 | 10827049.70 | 561 767.31 |

The fourth column provides in the form of a bar chart the different costs that conform to the Economic pillar. From an economic standpoint, policies P1, P2, and P3 demonstrate superior performance compared to traditional maintenance policies, and policy P4 exhibits comparable results. CM policies are associated with higher breakdown costs, which represent the primary maintenance expense. It is clear that the CM policy incurs higher breakdown costs due to its reactive approach, in which maintenance action is only performed once a breakdown has occurred. In contrast, the CBM77 policy can prevent more breakdowns by performing maintenance before they occur, but it is challenging to coordinate the technicians involved in the process. In terms of social impact, policies P3, P7, and P8 have demonstrated the most effective approach to reducing fatigue. The P3 policy, shown in Fig. 13, successfully balances the trade-off between social and economic aspects, achieving strong results in the social factor while only slightly affecting the economic aspect. In terms of reducing carbon footprint, policies P6 to P8 are the most effective. However, the policy P4, which significantly minimize the carbon footprint compared to the CM policies, and with the exception of one CBM77 policy, also optimize maintenance costs to a large extent.

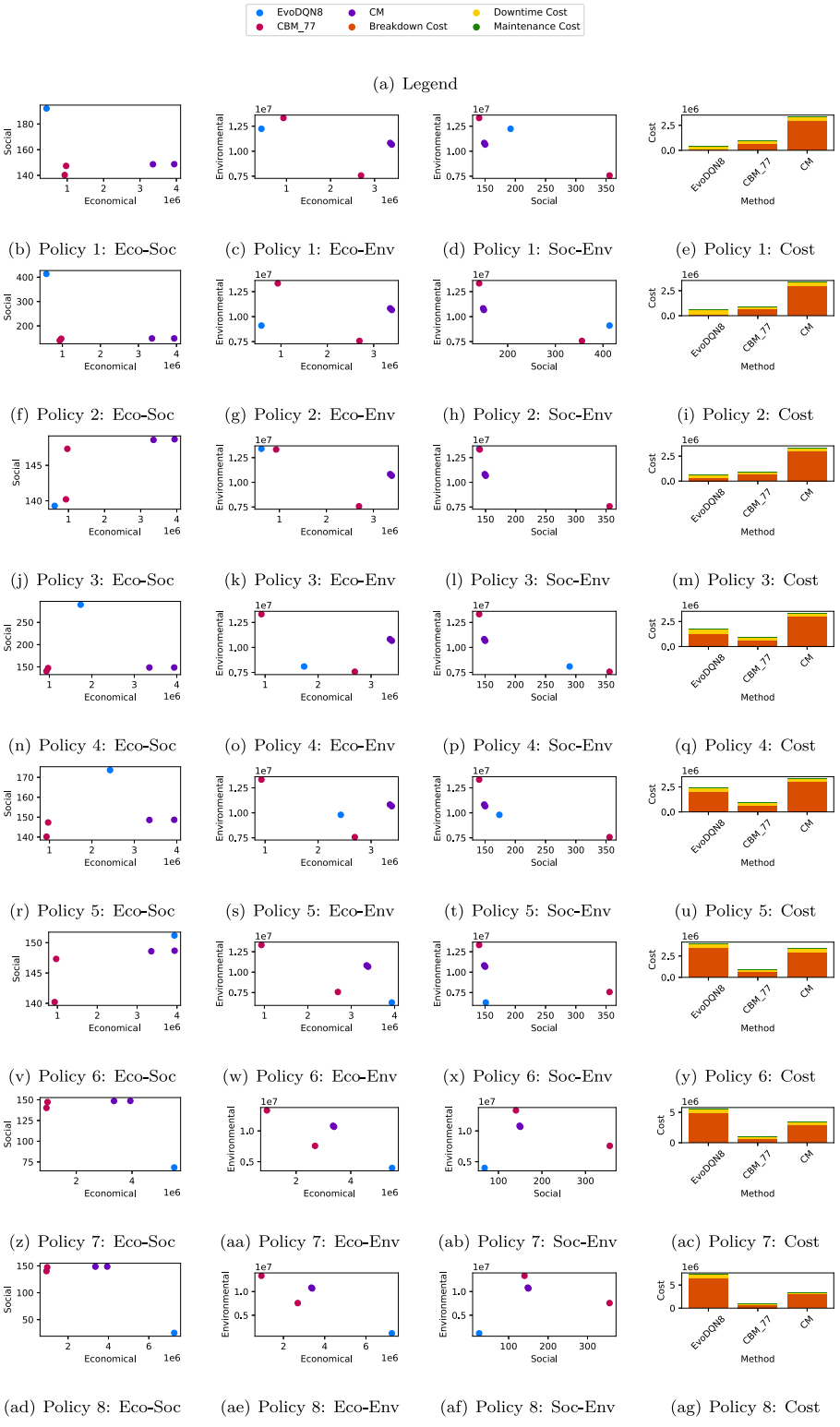



**Fig. 13.** 2D Projection of maintenance policy trade-offs across economic, social, and environmental objectives.

### 5.4. Operational feasibility in industrial operations

It is essential for companies to achieve a balance between the dual objectives of maximizing profits and minimizing costs while maintaining a sustainable approach. However, policies that give more priority to environmental or social criteria than economic ones are not relevant in an industrial context. In this regard, the policies P1, P2, P3 are well-suited for implementation in industrial contexts where economic considerations are of primary importance. With respect to the pair of objectives, policy P3 shows notable performance in balancing social

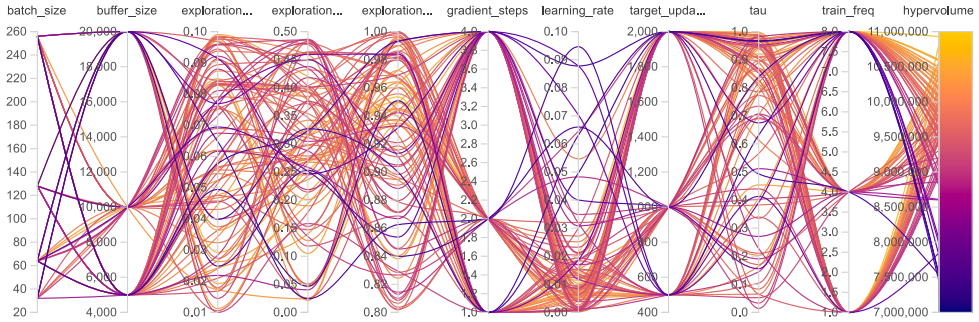



**Fig. B.14.** Parallel coordinate plot showing the relationship between hyperparameter settings and their corresponding hypervolume performance for the DQN algorithm.

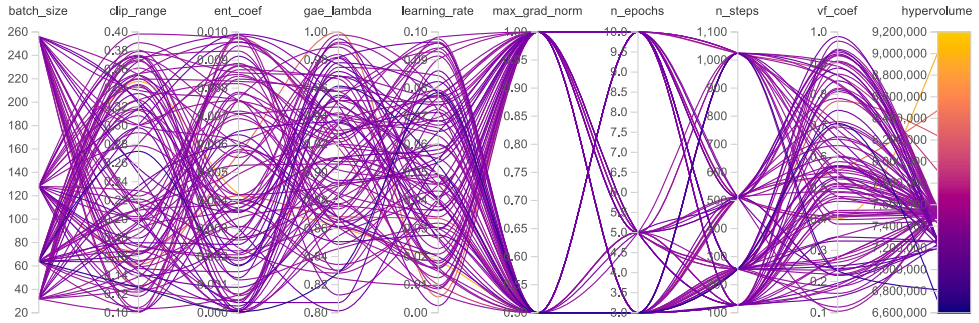
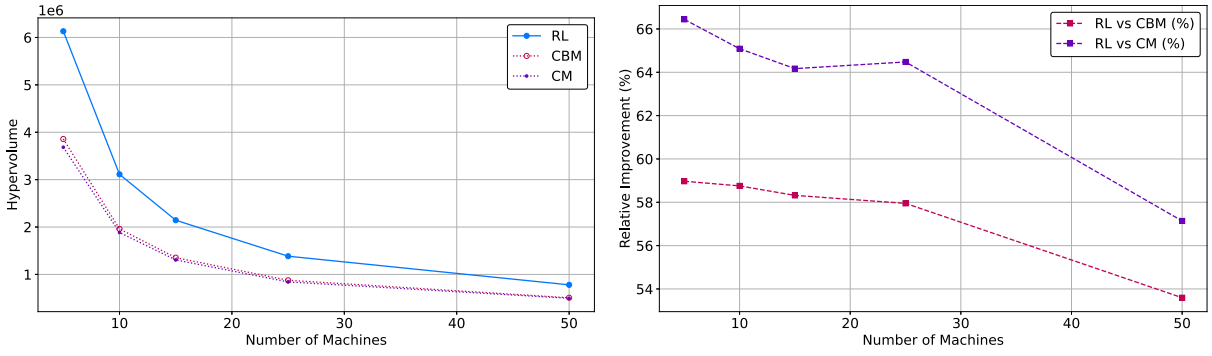



**Fig. B.15.** Parallel coordinate plot showing the relationship between hyperparameter settings and their corresponding hypervolume performance for the PPO algorithm.




(a) Hypervolume of maintenance policies (b) Relative improvement of maintenance policies

**Fig. C.16.** Comparison of maintenance policies under different system scales. (a) Hypervolume results of RL, CBM, and CM policies across different numbers of machines. (b) Relative improvement of RL over CBM and CM as the system scales.

and economic considerations, while policy P2 exhibits a particularly strong economic and environmental focus. It is important to note that while some EvoDQN policies may not be directly relevant for implementation in operational settings, as previously discussed, they remain essential for an optimal exploration of the search space.

Although the CBM and CM policies are mainly oriented to economics (they do not take well into account the balance between the three TBL criteria), we can still highlight that the policies are viable options for application in real industrial settings, especially when using the dispatch rule R1 and R3, which allow a good performance in the economic aspect and also reflected in the social aspect.

From an operational condition perspective, an important KPI to be analyzed is the production cycle of each policy. This KPI is summarized in Table 7 for all the 20 evaluated policies. The set of P1, P2, P3, and P4 policies demonstrate better performance compared to the traditional policies. P1 exhibits the most notable improvement in terms of the average production cycle, with a 8.02% increase compared to CBM77 and a 22.81% improvement for CM in its most cost-effective policies. These policies also represent the most cost-effective options, as the highest costs are, respectively, the breakdown and downtime, which represents production equipment with a higher production cycle.

## 6. Conclusion

This article presents a framework for developing sustainable maintenance policies based on a hybrid approach that combines Evolutionary Computation and DRL. Our proposal involves multiple agents, each with a preference vector that determines the reward. After training, the agents undergo an evolutionary process in which new agents are created and trained with the goal of maximizing the hypervolume and identifying a set of distributed policies within the preference space. In addition, a scenario based on parallel machines that experience

**Table 7**
Production cycle mean and standard deviation for various policies.

| Policy | PC Mean | PC Std |
|---|---|---|
| EvoDQN8 P1 | 31 407.06 | 318.96 |
| EvoDQN8 P2 | 31 768.42 | 545.01 |
| EvoDQN8 P3 | 30 555.22 | 414.28 |
| EvoDQN8 P4 | 29 928.67 | 1785.59 |
| EvoDQN8 P5 | 27 951.21 | 575.46 |
| EvoDQN8 P6 | 25 921.69 | 1032.73 |
| EvoDQN8 P7 | 23 440.13 | 417.65 |
| EvoDQN8 P8 | 20 496.47 | 849.48 |
| CBM77 R1 | 28 863.94 | 269.56 |
| CBM77 R2 | 24 800.80 | 324.57 |
| CBM77 R3 | 28 860.07 | 249.15 |
| CBM77 R4 | 24 740.38 | 329.07 |
| CBM77 R5 | 29 074.47 | 287.84 |
| CBM77 R6 | 25 796.58 | 576.20 |
| CBM77 R7 | 28 996.30 | 317.74 |
| CBM77 R8 | 25 749.26 | 511.02 |
| CM R1 | 24 238.93 | 176.51 |
| CM R3 | 24 286.44 | 183.50 |
| CM R5 | 25 561.90 | 382.69 |
| CM R7 | 25 572.46 | 412.26 |

degradation and repair by technicians was proposed, which incorporates the three pillars of sustainability. The proposed EvoDQN approach obtains distributed policies in the preference space demonstrating different trade-offs among the TBL objectives. The results showed how the economic pillar contrasts with the proposed social and environmental pillars. In addition, we found that our approach produces superior results in terms of the production cycle compared to classical maintenance policies that lead to higher profits.

### 6.1. Implications

Maintenance optimization is a process that involves understanding the condition of the equipment by incorporating the use of technology to monitor variables in real time such as temperature, vibration, pressure, and humidity. The collection and analysis of this data using ML algorithms is essential for identifying patterns, correlations, and/or abnormalities that could indicate potential equipment failures or performance issues. However, it is important to consider the additional costs associated with implementing this technology, including the installation of sensors, network infrastructure, and computational power to train the ML models. Furthermore, incorporating social aspects can improve well-being and job satisfaction, as well as help identify specific training needs, enabling the development of new skills and the improvement of existing ones.

### 6.2. Limitations and future work

This research provides valuable insights into the optimization of maintenance scheduling through sustainable practices. However, more work is required to address some challenges and explore potential research directions.

- Expanding the problem to a production line where the interdependence of machines is a key consideration in maintenance scheduling optimization. In addition, to consider energy consumption as part of the cost and production as a profit.
- Addressing the scalability of the system as the search space for policies increases with the addition of new machines to the production line, while also enabling evaluations over longer horizons, such as months or years.
- Exploring additional effects of technician fatigue, such as temporary unavailability or reduced maintenance quality, to better capture other effects on system performance.
- Improving the transparency of policies generated by DRL algorithms, which often lack explainability. Explainable RL is an important area of focus to ensure that decision-makers can better understand and trust the policies generated.
- Ensuring that the selection of TBL weights by the operator or manager is aligned with the company's objectives. Techniques like the Analytic Hierarchy Process (AHP) can be employed to guide the selection process.

## CRediT authorship contribution statement

**Marcelo Luis Ruiz-Rodríguez:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology, Conceptualization. **Sylvain Kubler:** Writing – review & editing, Writing – original draft, Supervision, Methodology, Conceptualization. **Jérémy Robert:** Supervision, Methodology, Conceptualization. **Alexandre Voisin:** Supervision, Methodology, Conceptualization. **Yves Le Traon:** Supervision, Conceptualization.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This research was funded in whole or in part by the Luxembourg National Research Fund (FNR), grant reference 16756339. For the purpose of open access, and in fulfilment of the obligations arising from the grant agreement, the author has applied a Creative Commons Attribution 4.0 International (CC BY 4.0) license to any Author Accepted Manuscript version arising from this submission.

## Appendix A. List of acronyms and variables

Table A.8 provides the list of acronyms and variables used throughout this paper.

## Appendix B. Hyperparameters definition

The hyperparameters for the baseline models, DQN and PPO (implemented using Stable-Baselines3), were selected based on a hyperparameter search using Weights & Biases (wandb) sweeps. The search consisted of a single phase that involved 100k environment steps, during which the best performing parameters were identified for use in longer training runs. The explored ranges and the selected hyperparameter values are summarized in Tables B.9.

To provide further insight into the hyperparameter optimization process, we present parallel coordinates plot that illustrates the relationships and trade-offs between the hyperparameter settings and their corresponding impact on the optimization metric for DQN and PPO (see Figs. B.14 and B.15).

## Appendix C. Scalability analysis

To assess the scalability of our approach, we evaluated the performance of the system under varying levels of complexity by increasing the number of machines to 5, 10, 15, 25, and 50. In all cases, the number of technicians was kept fixed (two technicians) across the different maintenance strategies (RL, CBM, and CM). For consistency, we used the same parameter configuration for the training as used in EvoDQN8 (see Section 5.2). The results are illustrated in Fig. C.16, showing the impact of system scale on both hypervolume and relative improvement.

**Table A.8**
List of acronym/variables and their descriptions.

| Acronym/Variable | Description |
|---|---|
| AD | Artificial Datasets |
| CBM | Condition-Based Maintenance |
| CM | Corrective Maintenance |
| CNC | Computer Numerical Control |
| DQN | Deep Q-Network |
| DRL | Deep Reinforcement Learning |
| Eco | Economic |
| Env | Environmental |
| EvoDQN | Evolutionary multi-objective multi-agent DQN |
| Fac | Factory |
| GA | Genetic Algorithm |
| KPI | Key Performance Indicator |
| Mac | Machine |
| Me | Metaheuristic |
| ML | Machine Learning |
| MARL | Multi-agent Reinforcement Learning |
| MOMDP | multi-objective Markov decision process |
| MP | Mathematical Programming |
| Mu | Multiple Objectives |
| NSGA-II | Non-dominated Sorting Genetic Algorithm II |
| PdM | Predictive Maintenance |
| PM | Preventive Maintenance |
| PL | Production Line |
| RD | Real Data |
| RL | Reinforcement Learning |
| Si | Single Objective |
| Soc | Social |
| TBL | Triple Bottom Line |
| $M$ | A set of machines: $M = \{1, \ldots, m\}$ |
| $T$ | A set of technicians |
| $E_m$ | Energy consumption (kWh) of machine $m \in M$ |
| $\omega_m$ | State of machine for $m \in M$ |
| $\omega_t$ | State of technicians for $t \in T$ |
| $\Gamma_m(\alpha_m, \beta_m)$ | Gamma process for machine $m \in M$, with shape $\alpha_m(t)$ and scale $\beta_m$ parameters |
| $\rho_m$ | Failure threshold for $m \in M$ |
| $g_{m,k}$ | Degradation on machine $m$ at time step $k$ |
| $q_t$ | Remaining time of technician $t$ performing maintenance |
| $\mu_{\text{car}}$ | Mean of the normal distribution for the CO2 forecast parameter |
| $\sigma_{\text{car}}$ | Standard deviation of the normal distribution for the CO2 forecast parameter |
| $C^P$ | Perfect maintenance cost |
| $C^I$ | Imperfect maintenance cost |
| $C^D$ | Downtime cost |
| $C^B$ | Breakdown cost |
| $\tau^B$ | Increased breakdown repair time |
| $\xi$ | Weights for balance workload |
| $\mathcal{T}_{t,m}$ | Maintenance time of technician $t$ in mach. $m$ |
| $F(\tau^F)$ | Technician fatigue accumulation function at maintenance time $\tau^F$ |
| $\mathcal{R}(\tau^R)$ | Technician fatigue recovery function after rest time $\tau^R$ |
| $\lambda^F$ | Fatigue parameter |
| $\mu^R$ | Recovery parameter |
| $\mu_{\text{break}}$ | mean for the additional repair time after a breakdown |
| $\sigma^2_{\text{break}}$ | variance for the additional repair time after breakdown |
| $CI$ | Carbon intensity (gCO2eq/kWh) |
| $CI^{\text{forecast}}$ | Carbon intensity forecast |
| $CI^n$ | Carbon intensity noise |
| $\Delta^{CI}$ | Carbon intensity forecast steps |
| $v_t$ | Fatigue level of technician $t$ |
| $m_{HDL}$ | Machine with the Highest degradation level |
| $m_{HCT}$ | Machine with the Highest cycle time |
| $t_{mF}$ | Technician with the minimal level of fatigue |
| $t_{MS}$ | Technician with the maximum skills |
| $x_{m,t,k}$ | Binary decision variable indicating if maintenance is performed on machine $m$ by technician $t$ at time step $k$ |
| $x^p_{m,t,k}$ | Binary decision variable indicating if *perfect maintenance* is performed on machine $m$ by technician $t$ at time step $k$ |
| $x^i_{m,t,k}$ | Binary decision variable indicating if *imperfect maintenance* is performed on machine $m$ by technician $t$ at time step $k$ |
| $S$ | State space |
| $A$ | Action space |
| $\mathcal{P}$ | Transition function |
| $R$ | Reward function |
| $\gamma$ | Discount factor |

**Table B.9**
Summary of selected hyperparameters for DQN and PPO, including their search ranges.

| Algorithm | Hyperparameter | Search range | Selected value |
|---|---|---|---|
| DQN | Batch Size | {32, 64, 128, 256} | 256 |
| | Buffer Size | {5000, 10000, 20000} | 5,000 |
| | Exploration Final Epsilon | [0.01, 0.1] | 0.0146 |
| | Exploration Fraction | [0.01, 0.5] | 0.392 |
| | Exploration Initial Epsilon | [0.8, 1.0] | 0.895 |
| | Gradient Steps | {1, 2, 4} | 1 |
| | Learning Rate | [0.0001, 0.1] | 0.0211 |
| | Target Update Interval | {500, 1000, 2000} | 1,000 |
| | Tau | [0.01, 1.0] | 0.617 |
| | Train Frequency | {1, 4, 8} | 4 |
| PPO | Batch Size | {32, 64, 128, 256} | 256 |
| | Clip Range | [0.1, 0.4] | 0.200 |
| | Entropy Coefficient | [1e−08, 0.01] | 0.00711 |
| | GAE Lambda | [0.8, 1.0] | 0.969 |
| | Learning Rate | [0.0001, 0.1] | 0.00247 |
| | Max Gradient Norm | {0.5, 1} | 0.5 |
| | Number of Epochs | {3, 5, 10} | 10 |
| | Number of Steps | {128, 256, 512, 1024} | 1,024 |
| | Value Function Coefficient | [0.1, 1.0] | 0.826 |

We computed the hypervolume of the solutions obtained to quantify the performance across the three objectives, as presented in Fig. C.16(a), as well as the relative improvement of RL over CBM and CM, as shown in Fig. C.16(b). The results show a decrease in hypervolume as the number of machines increases, reflecting the increasing challenge of balancing trade-offs between objectives. Specifically, with more machines and a fixed number of technicians, the system incurs higher costs and becomes less efficient, resulting in a reduced hypervolume.

Additionally, the higher number of machines contributes to a greater environmental impact, particularly in terms of carbon footprint. From a human perspective, the workload on technicians also increases significantly, as the workforce does not scale with the size of the system. This combination of factors highlights the importance of intelligent policy design, particularly in resource-constrained environments.

## Data availability

Data will be made available on request.

## References


Abbas, A.N., Chasparis, G.C., Kelleher, J.D., 2024. Hierarchical framework for interpretable and specialized deep reinforcement learning-based predictive maintenance. Data Knowl. Eng. 149.

Aissani, N., Beldjilali, B., Trentesaux, D., 2009. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. Eng. Appl. Artif. Intell. 22 (7), 1089–1103.

Al-Hourani, S., 2020. Rescheduling preventive maintenance for utilities equipment using criticality analysis. In: 2020 Industrial and Systems Engineering Conference, ISEC 2020. Institute of Electrical and Electronics Engineers Inc..

An, Y., Chen, X., Zhang, J., Li, Y., 2020. A hybrid multi-objective evolutionary algorithm to integrate optimization of the production scheduling and imperfect cutting tool maintenance considering total energy consumption. J. Clean. Prod. 268.

Arena, S., Florian, E., Sgarbossa, F., Sølvsberg, E., Zennaro, I., 2024. A conceptual framework for machine learning algorithm selection for predictive maintenance. Eng. Appl. Artif. Intell. 133, 108340.

Arık, O.A., 2021. Population-based tabu search with evolutionary strategies for permutation flow shop scheduling problems under effects of position-dependent learning and linear deterioration. Soft Comput. 25 (2), 1501–1518.

Asghar, I., Sarkar, B., jun Kim, S., 2019. Economic analysis of an integrated production–inventory system under stochastic production capacity and energy consumption. Energies 12 (16).

Attia, A.M., Alatwi, A.O., Al Hanbali, A., Alsawafy, O.G., 2024. Joint maintenance planning and production scheduling optimization model for green environment. J. Qual. Maint. Eng. 30 (1), 153–174.

Baykasoğlu, A., Madenoğlu, F.S., 2021. Greedy randomized adaptive search procedure for simultaneous scheduling of production and preventive maintenance activities in dynamic flexible job shops. Soft Comput. 25 (23), 14893–14932.

Ben Abdellafou, K., Hadda, H., Korbaa, O., 2021. Heuristic algorithms for scheduling intrees on m machines with non-availability constraints. Oper. Res. 21 (1), 55–71.

Bencheikh, G., Letouzey, A., Desforges, X., 2022. An approach for joint scheduling of production and predictive maintenance activities. J. Manuf. Syst. 64, 546–560.

Bhatta, K., Chang, Q., 2023a. An integrated control strategy for simultaneous robot assignment, tool change and preventive maintenance scheduling using heterogeneous graph neural network. Robot. Comput.-Integr. Manuf. 84.

Bhatta, K., Chang, Q., 2023b. Integrating robot assignment and maintenance management: A multi-agent reinforcement learning approach for holistic control. IEEE Robot. Autom. Lett. 8 (9), 5338–5344.

Bodnar, C., Day, B., Lió, P., 2020. Proximal distilled evolutionary reinforcement learning. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, (vol. 04), pp. 3283–3290.

Boufellouh, R., Belkaid, F., 2019. Multi-objective approach to optimize production and maintenance scheduling in flow shop environment under non-renewable resources constraints. In: 2019 International Conference on Advanced Electrical Engineering ICAEE 2019. Institute of Electrical and Electronics Engineers Inc..

Cacereño, A., Greiner, D., Galván, B.J., 2021. Multi-objective optimum design and maintenance of safety systems: An in-depth comparison study including encoding and scheduling aspects with nsga-ii. Mathematics 9 (15).

Chang, C.C., 2023. Optimal preventive replacement last policy for a successive random works system with random lead time. Comm. Statist. Theory Methods 52 (4), 1202–1216.

Chen, R., Yang, B., Li, S., Wang, S., 2020. A self-learning genetic algorithm based on reinforcement learning for flexible job-shop scheduling problem. Comput. Ind. Eng. 149, 106778.

Cui, W., Lu, B., 2020. A bi-objective approach to minimize makespan and energy consumption in flow shops with peak demand constraint. Sustain. (Switzerland) 12 (10).

Cui, W., Lu, B., 2021. Energy-aware operations management for flow shops under TOU electricity tariff. Comput. Ind. Eng. 151.

Cui, W., Sun, H., Xia, B., 2020. Integrating production scheduling, maintenance planning and energy controlling for the sustainable manufacturing systems under TOU tariff. J. Oper. Res. Soc. 71 (11), 1760–1779.

Detti, P., Nicosia, G., Pacifici, A., Zabalo Manrique de Lara, G., 2019. Robust single machine scheduling with a flexible maintenance activity. Comput. Oper. Res. 107, 19–31.

Diaz Cazanas, R., Delgado Sobrino, D.R., Caganova, D., Kostal, P., Velisek, K., 2019. Joint programming of production-maintenance tasks: A simulated annealing-based method. Int. J. Simul. Model. 18 (4), 666–677.

Djassemi, M., Seifoddini, H., 2019. Analysis of critical machine reliability in manufacturing cells. J. Ind. Eng. Manag. 12 (1), 70–82.

Do, P., Nguyen, V.T., Voisin, A., Iung, B., Neto, W.A.F., 2024. Multi-agent deep reinforcement learning-based maintenance optimization for multi-dependent component systems. Expert Syst. Appl. 245.

Dong, J., Ye, C., 2020. Research on two-stage joint optimization problem of green manufacturing and maintenance for semiconductor wafer. Math. Probl. Eng. 2020.

Drugan, M.M., 2019. Reinforcement learning versus evolutionary computation: A survey on hybrid algorithms. Swarm Evol. Comput. 44, 228–246.

Electricity Maps, 2024. Data portal. (Accessed: 17 July 2024).

European Commission, 2019. The European green deal. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A52019DC0640Communication from the Commission to the European Parliament, the Council, the European Economic and Social Committee and the Committee of the Regions.

Farahani, A., Tohidi, H., Shoja, A., 2019. An integrated optimization of quality control chart parameters and preventive maintenance using Markov chain. Adv. Prod. Eng. Manag. 14 (1), 5–14.

Feng, M., Li, Y., 2022. Predictive maintenance decision making based on reinforcement learning in multistage production systems. IEEE Access 10, 18910–18921.

Fu, T., Gao, W., Coley, C., Sun, J., 2022. Reinforced genetic algorithm for structure-based drug design. Adv. Neural Inf. Process. Syst. 35, 12325–12338.

Ghaleb, M., Taghipour, S., Sharifi, M., Zolfagharinia, H., 2020. Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures. Comput. Ind. Eng. 143 (October 2019), 106432.

Giner, J., Lamprecht, R., Gallina, V., Laflamme, C., Sielaff, L., Sihn, W., 2021. Demonstrating reinforcement learning for maintenance scheduling in a production environment. In: IEEE International Conference on Emerging Technologies and Factory Automation, Vol. 2021-Septe. ETFA, Institute of Electrical and Electronics Engineers Inc..

Gupta, S., Jain, A., 2021. Assessing the effect of reliability-based maintenance approach in job shop scheduling with setup time and energy consideration using simulation; a simulation study. Smart Sci. 9 (4), 283–304.

Gupta, S., Jain, A., 2022. Analysis of integrated preventive maintenance and machine failure in stochastic flexible job shop scheduling with sequence-dependent setup time. Smart Sci. 10 (3), 175–197.

Gupta, S., Jain, A., Chan, F.T., Phanden, R.K., 2023. A study on simulation-based optimization of a stochastic flexible job shop scheduling undergoing preventive maintenance with sequence-dependent setup time. Int. J. Interact. Des. Manuf..

Gusenbauer, M., Haddaway, N.R., 2020. Which academic search systems are suitable for systematic reviews or meta-analyses? Evaluating retrieval qualities of google scholar, PubMed, and 26 other resources. Res. Synth. Methods 11 (2), 181–217.

Hayes, C.F., Rădulescu, R., Bargiacchi, E., Källström, J., Macfarlane, M., Reymond, M., Verstraeten, T., Zintgraf, L.M., Dazeley, R., Heintz, F., et al., 2022. A practical guide to multi-objective reinforcement learning and planning. Auton. Agents Multi- Agent Syst. 36 (1), 26.

Hedjazi, D., Layachi, F., Boubiche, D.E., 2019. A multi-agent system for distributed maintenance scheduling. Comput. Electr. Eng. 77, 1–11.

Hidri, L., Alqahtani, K., Gazdar, A., Badwelan, A., 2021. Integrated scheduling of tasks and preventive maintenance periods in a parallel machine environment with single robot server. IEEE Access 9, 74454–74470.

Huang, J., Chang, Q., Arinez, J., 2020. Deep reinforcement learning based preventive maintenance policy for serial production lines. Expert Syst. Appl. 160.

Jaber, M.Y., Givi, Z., Neumann, W.P., 2013. Incorporating human fatigue and recovery into the learning–forgetting process. Appl. Math. Model. 37 (12–13), 7287–7299.

Jawahir, I., Dillon, O., Rouch, K., Joshi, K.J., Venkatachalam, A., Jaafar, I.H., 2006. Total life-cycle considerations in product design for sustainability: A framework for comprehensive evaluation. In: Proceedings of the 10th International Research/Expert Conference, Barcelona, Spain, vol. 1, (no. 10), Citeseer.

Jayasuriya, R.P., Amarasinghe, P.A., Abeygunawardane, S.K., 2021. Application of artificial intelligence for maintenance modelling of critical machines in solid tire manufacturing. In: 2021 International Conference on Innovative Trends in Information Technology. ICITIIT 2021, Institute of Electrical and Electronics Engineers Inc..

Jia, Y., Yan, Q., Wang, H., 2023. Q-learning driven multi-population memetic algorithm for distributed three-stage assembly hybrid flow shop scheduling with flexible preventive maintenance. Expert Syst. Appl. 232.

Kedy, G.S.M., Penda, M.C., Nneme Nneme, L., Mayi, O.T.S., Lehman, L.G., 2024. Enhancing the effectiveness of joint production and maintenance scheduling based on a multi-agent system and a pigouvian approach of externalities. Prod. Eng. (0123456789).

Köksal Ahmed, E., Li, Z., Veeravalli, B., Ren, S., 2022. Reinforcement learning-enabled genetic algorithm for school bus scheduling. J. Intell. Transp. Syst. 26 (3), 269–283.

Lee, C.-Y., 1996. Machine scheduling with an availability constraint. J. Global Optim. 9 (3), 395–416.

Li, X., Ran, Y., Chen, B., Chen, F., Cai, Y., Zhang, G., 2023. Opportunistic maintenance strategy optimization considering imperfect maintenance under hybrid unit-level maintenance strategy. Comput. Ind. Eng. 185 (September), 109624.

Mi, S., Feng, Y., Zheng, H., Li, Z., Gao, Y., Tan, J., 2020. Integrated intelligent green scheduling of predictive maintenance for complex equipment based on information services. IEEE Access 8, 45797–45812.

Mirahmadi, N., Taghipour, S., 2019. Energy-efficient optimization of flexible job shop scheduling and preventive maintenance. In: Proceedings - Annual Reliability and Maintainability Symposium, vol. 2019-Janua, Ryerson Univ, Dept Mech & Ind Engn, Reliabil Risk & Maintenance Res Lab, Toronto, ON M5B 2K3, Canada.

Mnih, V., Kavukcuoglu, K., Silver, D., Graves, A., Antonoglou, I., Wierstra, D., Riedmiller, M., 2013. Playing atari with deep reinforcement learning.

Muhuri, P.K., Shukla, A.K., Abraham, A., 2019. Industry 4.0: A bibliometric analysis and detailed overview. Eng. Appl. Artif. Intell. 78, 218–235.

Nguyen, V.T., Do, P., Vosin, A., Iung, B., 2022. Artificial-intelligence-based maintenance decision-making and optimization for multi-state component systems. Reliab. Eng. Syst. Saf. 228.

Ochella, S., Shafiee, M., Dinmohammadi, F., 2022. Artificial intelligence in prognostics and health management of engineering systems. Eng. Appl. Artif. Intell. 108, 104552.

Paraschos, P.D., Koulinas, G.K., Koulouriotis, D.E., 2020. Reinforcement learning for combined production-maintenance and quality control of a manufacturing system with deterioration failures. J. Manuf. Syst. 56, 470–483.

Paraschos, P.D., Koulinas, G.K., Koulouriotis, D.E., 2023. A reinforcement learning/ad-hoc planning and scheduling mechanism for flexible and sustainable manufacturing systems. Flex. Serv. Manuf. J..

Perera, Y.S., Ratnaweera, D., Dasanayaka, C.H., Abeykoon, C., 2023. The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review. Eng. Appl. Artif. Intell. 121, 105988.

Qamhan, A.A., Ahmed, A., Al-Harkan, I.M., Badwelan, A., Al-Samhan, A.M., Hidri, L., 2020. An exact method and ant colony optimization for single machine scheduling problem with time window periodic maintenance. IEEE Access 8, 44836–44845.

Qin, W., Zhuang, Z., Liu, Y., Xu, J., 2022. Sustainable service oriented equipment maintenance management of steel enterprises using a two-stage optimization approach. Robot. Comput.-Integr. Manuf. 75, 102311.

Ran, Y., Zhou, X., Lin, P., Wen, Y., Deng, R., 2019. A survey of predictive maintenance: Systems, purposes and approaches.

Roijers, D.M., Vamplew, P., Whiteson, S., Dazeley, R., 2013. A survey of multi-objective sequential decision-making. J. Artificial Intelligence Res. 48, 67–113.

Rokhforoz, P., Fink, O., 2022. Maintenance scheduling of manufacturing systems based on optimal price of the network. Reliab. Eng. Syst. Saf. 217.

Ruiz Rodríguez, M.L., Kubler, S., de Giorgio, A., Cordy, M., Robert, J., Le Traon, Y., 2022. Multi-agent deep reinforcement learning based predictive maintenance on parallel machines. Robot. Comput.-Integr. Manuf. 78.

Ruiz-Rodríguez, M.L., Kubler, S., Robert, J., Le Traon, Y., 2024. Dynamic maintenance scheduling approach under uncertainty: Comparison between reinforcement learning, genetic algorithm simheuristic, dispatching rules. Expert Syst. Appl. 248, 123404.

Seidgar, H., Fazlollahtabar, H., Zandieh, M., 2020. Scheduling two-stage assembly flow shop with random machines breakdowns: integrated new self-adapted differential evolutionary and simulation approach. Soft Comput. 24 (11), 8377–8401.

Sharifi, M., Taghipour, S., 2021. Optimal production and maintenance scheduling for a degrading multi-failure modes single-machine production environment. Appl. Soft Comput. 106, 107312.

Shen, J., Zhu, Y., 2019. A parallel-machine scheduling problem with periodic maintenance under uncertainty. J. Ambient. Intell. Humaniz. Comput. 10 (8), 3171–3179.

Sin, I.H., Chung, B.D., 2020. Bi-objective optimization approach for energy aware scheduling considering electricity cost and preventive maintenance using genetic algorithm. J. Clean. Prod. 244.

Song, Y., Wei, L., Yang, Q., Wu, J., Xing, L., Chen, Y., 2023. RL-GA: A reinforcement learning-based genetic algorithm for electromagnetic detection satellite scheduling problem. Swarm Evol. Comput. 77, 101236.

Su, J., Huang, J., Adams, S., Chang, Q., Beling, P.A., 2022. Deep multi-agent reinforcement learning for multi-level preventive maintenance in manufacturing systems[formula presented]. Expert Syst. Appl. 192.

Sun, Z., Dababneh, F., Li, L., 2020. Joint energy, maintenance, and throughput modeling for sustainable manufacturing systems. IEEE Trans. Syst. Man, Cybern.: Syst. 50 (6), 2101–2112.

Sutton, R.S., Barto, A.G., 2018. Reinforcement learning: An introduction. MIT Press.

United Nations, 2015. Transforming our world: the 2030 agenda for sustainable development. ISBN: A/RES/70/1, p. 16301.

Wang, Z., Deng, Q., Zhang, L., Li, H., Li, F., 2023. Joint optimization of integrated mixed maintenance and distributed two-stage hybrid flow-shop production for multi-site maintenance requirements. Expert Syst. Appl. 215.

Wang, J., Du, H., Xing, J., Qiao, F., Ma, Y., 2020. Novel energy- and maintenance-aware collaborative scheduling for a hybrid flow shop based on dual memetic algorithms. IEEE Robot. Autom. Lett. 5 (4), 5613–5620.

Widiastuti, H., Sulistyani, A., Utami, E.R., 2022. Do environmental issues matter to investors? In: International Conference on Sustainable Innovation Track Accounting and Management Sciences (ICOSIAMS 2021). Atlantis Press, pp. 228–234.

Wu, C.H., Yao, Y.C., Dauzère-Pérès, S., Yu, C.J., 2020. Dynamic dispatching and preventive maintenance for parallel machines with dispatching-dependent deterioration. Comput. Oper. Res. 113, 104779.

Xia, T., Shi, G., Si, G., Du, S., Xi, L., 2021. Energy-oriented joint optimization of machine maintenance and tool replacement in sustainable manufacturing. J. Manuf. Syst. 59, 261–271.

Xia, T., Si, G., Shi, G., Zhang, K., Xi, L., 2022. Optimal selective maintenance scheduling for series–parallel systems based on energy efficiency optimization. Appl. Energy 314, 118927.

Yan, Q., Wang, H., Wu, F., 2022. Digital twin-enabled dynamic scheduling with preventive maintenance using a double-layer Q-learning algorithm. Comput. Oper. Res. 144.

Yan, Q., Wang, H., Yang, S., 2024. A learning-assisted bi-population evolutionary algorithm for distributed flexible job-shop scheduling with maintenance decisions. IEEE Trans. Evol. Comput..

Yazdani, R., Alipour-Vaezi, M., Kabirifar, K., Salahi Kojour, A., Soleimani, F., 2022. A lion optimization algorithm for an integrating maintenance planning and production scheduling problem with a total absolute deviation of completion times objective. Soft Comput. 26 (24), 13953–13968.

Ye, Z., Cai, Z., Yang, H., Si, S., Zhou, F., 2023. Joint optimization of maintenance and quality inspection for manufacturing networks based on deep reinforcement learning. Reliab. Eng. Syst. Saf. 236.

Yu, T.S., Han, J.H., 2021. Scheduling proportionate flow shops with preventive machine maintenance. Int. J. Prod. Econ. 231, 107874.