International Journal of Research in Industrial Engineering

www.riejournal.com

Int. J. Res. Ind. Eng. Vol. 13, No. 1 (2024) 71–87.

Paper Type: Research Paper

# The Joint Policy of Production, Maintenance, and Product Quality in a Multi-Machine Production System by Reinforcement Learning and Agent-based Modeling

**Mohammad Reza Nazabadi[1], Seyed Esmaeil Najafi[1,*], Ali Mohaghar[2], Farzad Movahedi Sobhani[1]**

[1] Department of Industrial Engineering, Science and Research Branch, Islamic Azad University Tehran, Iran; mrnazabadi@gmail.com; e.najafi@srbiau.ac.ir; farzadmovahedi@gmail.com.

[2] Faculty of Industrial Management, University of Tehran, Tehran, Iran; amohaghar@ut.ac.ir.

**Citation:**

| | |
|---|---|
| *Received: 06 August 2021*<br>*Revised: 08 October 2021*<br>*Accepted: 19 December 2021* | Nazabadi, M. R., Najafi, S. E., Mohaghar, A., & Movahedi Sobhani, F. (2024). The joint policy of production, maintenance, and product quality in a multi-machine production system by reinforcement learning and agent-based modeling. *International journal of research in industrial engineering, 5*(1), 71-87. |

## Abstract

Adopting an integrated production, maintenance, and quality policy in production systems is of great importance due to their interconnected influence. Consequently, investigating these aspects in isolation may yield an infeasible solution. This paper aims to address the joint optimal policy of production, maintenance, and quality in a two-machine-single-product production system with an intermediate buffer and final product storage. The production machines have degradation levels from as-good-as-new to the breakdown state. The failures increase the production machine's degradation level, and maintenance activities change the status to the initial state. Also, the quality of the final product depends on the level of degradation of the machines and the correlation between the degradation level of the production machines and the product's quality in the case that high degradation of the previous production machines leads to a high probability to produce wastage by the following machines is considered. The production system studied in this research has been modeled using the agent-based simulation, and the Reinforcement Learning (RL) algorithm has obtained the optimal integrated policy. The goal is to find an integrated optimal policy that minimizes production costs, maintenance costs, inventory costs, lost orders, breakdown of production machines, and low-quality production. The meta-heuristic technique evaluates the joint policy obtained by the decision-maker agent. The results show that the acquired joint policy by the RL algorithm offers acceptable performance and can be applied to the autonomous real-time decision-making process in manufacturing systems.

**Keywords:** Agent-based modeling, Reinforcement learning, Simulation-optimization, Production planning, Maintenance, Quality control.

## 1 | Introduction

Production planning, maintenance, and quality control are the most critical challenges of a production system that directly affect the system's performance. Production planning aims to decide how to divide and schedule the tasks to achieve specific goals, such as minimizing tardiness or maximizing production. The maintenance

Corresponding Author: e.najafi@srbiau.ac.ir

https://doi.org/10.22105/riej.2023.298557.1240

goal is to maximize machine availability time while minimizing associated costs. According to the definition of European standard EN13306 version 2010 [1], maintenance and repair is "the combination of all technical, administrative and managerial activities during the life of a device in order to maintain or return it to the required function."

Since performing any maintenance and repair activities will make the production machines inaccessible, optimizing maintenance and repair activities without considering the production planning limitations will lead to an infeasible solution. On the other hand, production planning is entirely affected by the time of availability of production machines. If the production machines are not available at the right time, achieving the desired goals in production planning will be almost impossible. In addition, if the maintenance and repair duration is allocated to the production, there will be a possibility of failure and breakdown of the production machines. Therefore, maintenance and production are two activities that conflict with each other [2].

The quality of the final products is also affected by the degradation level of the production machines. The high level of degradation increases the probability of producing low-quality products. Therefore, production planning, maintenance, and quality control must be considered as an integrated problem to obtain a feasible solution. However, very little research has been done on the integrated optimization of these aspects [3].

There has been significant development in artificial intelligence and machine learning in recent years, and their applications have been extended to many fields, including manufacturing systems. Machine learning is a subset of artificial intelligence in which computers can explore the patterns in the data and learn the policies. Machine learning is divided into three categories: supervised learning, unsupervised learning, and Reinforcement Learning (RL), which are shown in *Fig. 1* [4].

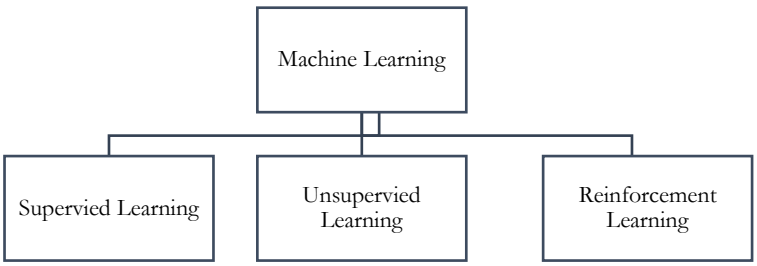


**Fig. 1. The machine learning categories.**

The required data for supervised learning has been labeled or categorized, e.g., data classification and regression for future prediction. In unsupervised learning, the data is unlabeled. This method is used to explore hidden patterns in the data, such as clustering methods. However, in RL, the learning process is done by trial and error in a dynamic environment. The agent takes action and receives associated rewards to maximize the expected rewards. This method is inspired by behavioral psychology. The agent's learning process can occur in a real-world environment or a simulation model. The application of RL in optimizing the production system is almost nascent. This research aims to apply RL to achieve the optimal joint policy of production planning, maintenance, and quality in a multi-machine production system. In order to evaluate the obtained policy, the Simulation-Optimization (SO) approach has been used. The main contributions of this paper are:

- *Applying RL and Agent-Based Modeling (ABM) to obtain a joint optimal policy of production, maintenance, and quality in a multi-machine single-product manufacturing system.*
- *Investigating the correlation between the degradation level of the production machines and the product's quality in the case that high degradation of the previous production machines leads to a high probability of produce wastage by the following machines.*
- *Comparing the acquired joint policy by the decision-maker agent with the meta-heuristic method and evaluating the performance of the RL-based policy.*

The article's structure will be as follows. In Section 2, the literature review will be presented. In Section 3, the production system, the agent-based model, and the RL algorithm are described. In Section 4, the obtained policy has been evaluated by SO techniques. Finally, the conclusion and suggestions are provided in Section 5.

# 2|Literature Review

As mentioned, RL is an emerging approach to optimize production systems. Zheng et al. [5] consider a production system consisting of two machines and one intermediate buffer [5]. The production system has been modeled by Discrete Event Simulation (DES), and RL has been used to obtain the optimal joint policy of production and maintenance based on the inventory level and the state of the production machines. Kuhnle et al. [6] optimized a parallel multi-machine production system using RL to achieve the best schedule for maintenance activities, increase production rates, and reduce maintenance costs [6]. ABM has been applied to model the production system, and each machine is considered an agent. In addition, each agent independently satisfies the demand. Xanthopoulos et al. [7] investigated a joint production and maintenance problem in a single machine–single product system. The system has a downstream buffer to store the final products and satisfy the demand. The optimal joint policy to minimize the inventory level and demand backorders has been acquired by RL. The extension of the previous research has been done by Paraschos et al. [8]. They consider the quality of products so that the production machine has a degradation level that affects the quality of the final products. The maintenance and repair activities can be performed to improve the degradation level. The optimal joint production, maintenance, and quality have been yielded by RL.

Yang et al. [9] have considered the joint optimal Preventive Maintenance (PM) and production scheduling policy in a similar production system by RL approach [9]. Deep Reinforcement Learning (DRL) has acquired the PM policy of a multi-machine, single-product system in [10]. Su et al. [11] investigate the challenge of designing PM policies for large-scale manufacturing systems and propose a novel approach using multi-agent RL to address the complexity of such systems [11]. They discuss that Designing efficient PM policies for large-scale manufacturing systems is difficult due to non-linearity and stochasticity in these complex systems. Zhao and Smidts [12] address challenges in maintenance policy optimization, in which decision-maker agents encounter an imperfect understanding of system degradation models and have a limited ability to observe system degradation states [12]. They proposed RL to tackle these challenges, specifically for maintenance problems with Markov degradation processes. Ye et al. [13] investigate the joint optimization of manufacturing systems, specifically focusing on large-scale dynamic systems, such as manufacturing networks, which have complex structures [13]. The paper proposes a novel approach using RL, specifically the Deep Deterministic Policy Gradient (DDPG) algorithm, to achieve joint optimization of PM and work-in-process quality inspection in manufacturing networks with reliability-quality interactions.

SO is another method to obtain optimal policies in production systems. In a study conducted by Lavoie et al. [14] they proved the effectiveness of SO algorithms in optimizing production systems. These algorithms have also been used in joint production planning, maintenance, and quality control optimization. Bouslah et al. [15] investigated a production system with two machines. In this research, hybrid optimization was performed using SO techniques to minimize the total costs. The problem of production planning and quality control is presented in [16]. They use SO to optimize a single machine-single product system jointly. The extension of the previous research has been done in [3]. They optimized combined production planning, maintenance, and quality control of a continuous single-machine production machine. Tambe and Kulkarni [17] introduce an integrated planning approach for optimizing the three core functions of shop floor management: maintenance, production scheduling, and quality. The methodology focuses on the conditional reliability of components and their impact on the overall system operation. The primary objective is to minimize the system operation cost through combined decision-making and investigate the integrated policy's cost-effectiveness compared to non-integrated planning approaches. The mathematical model is used to build a system model, and the meta-heuristics methods, such as simulated annealing and genetic algorithm, are applied to solve the optimization problem.

A summary of the presented papers is given in *Table 1*.

**Table 1. The summary of the papers.**

| No | Rf. | Method | Production System | Production Planning | Maintenance | Quality |
|---|---|---|---|---|---|---|
| 1 | [5] | RL | Two machines–Single product | ✓ | ✓ | |
| 2 | [6] | RL | Multi parallel machines–Single product | ✓ | ✓ | |
| 3 | [7] | RL | Single machine–Single product | ✓ | ✓ | |
| 4 | [8] | RL | Single machine–Single product | ✓ | ✓ | ✓ |
| 5 | [9] | RL | Single machine–Single product | ✓ | ✓ | |
| 6 | [10] | RL | Multi Machine–Single Product | ✓ | ✓ | |
| 7 | [15] | SO | Two machines–Single product | ✓ | ✓ | ✓ |
| 8 | [16] | SO | Single machine–Single product | ✓ | ✓ | |
| 9 | [3] | SO | Single machine–Single product | ✓ | ✓ | ✓ |
| 10 | [17] | SO | Multi Machine–Single Product | ✓ | ✓ | ✓ |
| 11 | [11] | RL | Multi Machine–Single Product | | ✓ | |
| 12 | [12] | RL | Single machine–Single product | | ✓ | |
| 13 | [13] | RL | Multi Machine–Single Product | | ✓ | ✓ |

The main contribution of this paper is to propose an RL-based method to jointly optimize the production, maintenance, and quality problem in a multi-machine single-product system by considering the correlation between the degradation level of the production machines and the product's quality in the case that high degradation of the previous production machines leads to a high probability to produce wastage by the following machines. To the author's knowledge, the proposed method has not been discussed in other research.

To evaluate the obtained RL-based policy, the SO technique has been implemented to compare the results. For this purpose, an agent-based model of the production system is developed, and a commercial SO package is used to obtain the optimal joint policy. The SO package combines heuristic and meta-heuristic methods such as tabu search, neural network, and scattered search to optimize the objective function.

# 3 | Problem Description

In this research, a production system consisting of two production machines $M_1$ and $M_2$, and one intermediate buffer with a maximum capacity of $B_{max}$ and a final product warehouse with a capacity of $I_{max}$ has been investigated (*Fig. 2*).

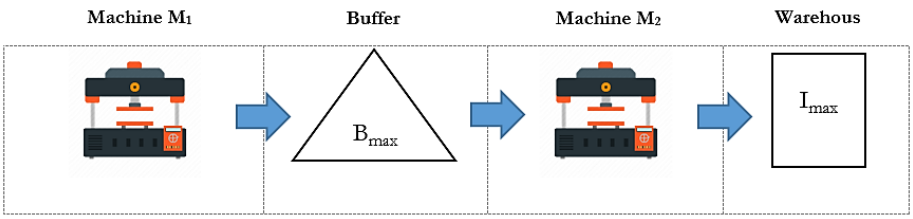


**Fig. 2. The production system.**

Manufacturing machines have depreciation levels that vary from the as-good-as-new $d_0$ to the breakdown $d_{max}$. Depreciation level increases due to failures at the rate of $\lambda_f$ during the production process. The probability of breakdown at each level of degradation levels has a probability of $p_{b,d_i}$. The higher degradation level increases the breakdown probability. Maintenance and repair activities return the degradation level of the production machine to the as-good-as-new state $d_0$.

The semi-final parts are produced by machine $M_1$. The production time follows the exponential distribution with the parameter $\lambda_{p_1}$. After that, the semi-final parts are stored in the intermediate buffer to be processed by the machine $M_2$. The processing time to produce the final parts follows the exponential distribution with parameter $\lambda_{p_2}$. The quality of the semi-final and final parts varies depending on the degradation level of the machine $M_1$ and $M_2$. The probability of producing low-quality semi-final parts in each degradation level of the machine $M_1$ is $p_{d_i}$. The quality of the final parts is related to the quality of the semi-final part and the degradation level of the machine $M_2$. The probability of producing high-quality parts by machine $M_2$ is $1\text{-}p_{d_{i,q_h}}$ when the semi-final product's quality is high, or $1\text{-}p_{d_{i,q_l}}$ when the quality is low. On the other hand, the wastage is produced with the portability $p_{d_{i,q_h}}$ and $p_{d_{i,q_l}}$ when the quality of semi-final parts is high or low, respectively. The probability of producing high-quality parts from the low-quality semi-final parts increases when the degradation level of the machine $M_2$ is close to $d_0$. Therefore, at a certain degradation level, $p_{d_{i,q_h}} < p_{d_{i,q_l}}$.

At the end of each day, the demand that follows the Poisson distribution with parameter $\lambda_d$ arrives, and satisfies when the inventory of the final parts is sufficient. Otherwise, the demand is backordered until the inventory is available. The maximum number of allowed backorders is $S_{max}$ and the number of missed orders is calculated by *Eq. (1)*.

$$\text{Missed orders} = |I\text{-}D+S_{max}|. \quad (1)$$

# 4 | Methodology

## 4.1 | The Agent-Based Modeling

There are three major paradigms in simulation modeling: DES, System Dynamics (SD), and ABM. Despite the traditional approaches such as DES and SD, ABM is relatively new and is more general. ABM enables the modelers to capture more complexities in dynamic systems [18]. So, in dynamic systems where the events are time-related, ABM can be used to model the system. There is no universal agreement for the definition of agents; e.g., the agent is defined as an entity with autonomous behavior [19] or an independent component [20]. In much literature, self-contained, autonomous, self-directed, and the ability to interact are mentioned as the essential characteristics of the agent [21].

The ABM consists of the following elements [21]:

- *The agents, characteristics, and behaviors.*
- *The way the agents interact.*
- *The environment.*

This paper uses ABM to simulate the production system due to the flexibility of the approach to capture all details and ease communication with the decision-maker agent. In the proposed production system, the agents are:

– *The Production machine $M_1$.*
– *The production machine $M_2$.*
– *The decision-maker agent.*

And the environment consists of the storage of the final products and the place where the agents interact.

### 4.1.1 | The agent's behavior

In *Fig. 3*, The behavior of the production agents is shown. The production agents can be in one of the following states:

– *ReadyForMessage: In this state, the production agent is waiting to receive a message from the decision-maker agent*
– *ReadyForProduction: In this state, the production agent investigates the buffer and storage volume and the remaining time till the end of the shift, and if all the required conditions are met, the production process will be started.*
– *Produce: The production agent starts the production process at the rate of $\lambda_p$ .*
– *Check condition: When the production process is going to be completed, depending on the degradation level of the production agent $M_1$, The production has high quality with 1-$p_{d_i}$ and low-quality with probability $p_{d_i}$. In the case of the production agent $M_2$, depending on the quality of the semi-final part and the degradation level of the agent, The production has high quality with the probability 1-$p_{d_{i,q}}$ or the production becomes wastage with $p_{d_{i,q}}$. Also, the agent's state transits to the Breakdown state with probability $p_{b,d_i}$or to the ReadyForProduction state with the probability 1-$p_{b,d_i}$:*
– *Breakdown: The operating agent will fail, and maintenance and repair activities will be necessary.*
– *Maintain: The maintenance and repair activities are performed to return the degradation level of the agent to as-good-as-new $d_0$.*
– *Idle: the production agent does not perform any activity during the shift.*

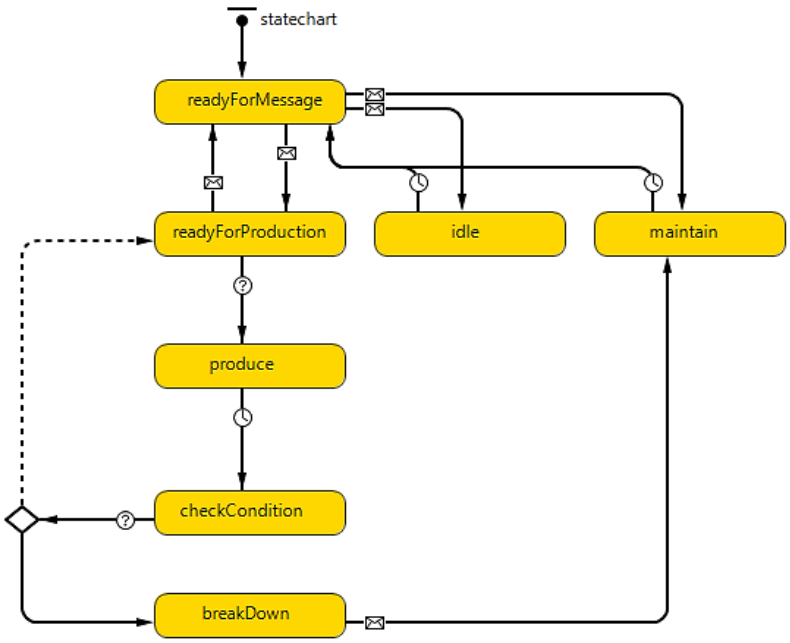


**Fig. 3. The production agent's behavior.**

### 4.1.2 | The decision-maker agent's behavior

In this paper, the decision-making process of the decision-maker agent is implemented based on the Markov-decision process, which consists of the following elements: state-space, actions, transition probability, and reward [4].

First, The decision-maker agent observes the state of the system and authorizes an action. The action is sent to the production agents as a message. Next, the agent considers the new state of the system and the corresponding reward of the authorized action. Then, this process continues until the optimal policy is obtained (*Fig. 4*).

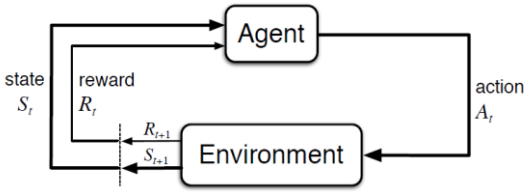


**Fig. 3. The Markov-decision process of the decision-maker agent.**

The decision-maker agent authorizes the actions that maximize the discounted future reward. Therefore, the agent authorizes an action in such a way that the following equation is maximized:

$$G_t=R_{t+1}+\gamma R_{t+1}+\gamma^2 R_{t+1}+...=\sum_{k=0}^{\infty}\gamma^k R_{t+k+1}. \quad (2)$$

### 4.1.3 | The State of the system

In every decision epoch, the decision-maker agent observes the state of the system as the following vector

$$S(t)=(S_{1,t},S_{2,t},S_{3,t},S_{4,t},S_{5,t})=(q_{lq},q_{hq},I,d_1,d_2). \quad (3)$$

Where $q_{lq}$ and $q_{hq}$ are the number of low-quality and high-quality semi-final parts in the intermediate buffer, respectively. I is the inventory level of final parts, and $d_1$ and $d_2$ are the degradation level of machine $M_1$ and $M_2$. The system state-space is

$$\begin{aligned}&S_1: 0, ..., B_{max},\\ &S_2: 0, ..., B_{max},\\ &S_3: -B_{max}, ..., I_{max},\\ &S_4: 0, ..., d_{max}, \text{ for the machine } M_1,\\ &S_5: 0, ..., d_{max}, \text{ for the machine } M_2.\end{aligned} \quad (4)$$

Consequently, the number of states that the decision-maker agent can find itself in is:

$$N(s)=(B_{max}+1)\times(B_{max}+1)\times(I_{max}+S_{max}+1)\times(d_{max\,M1}+1)\times(d_{max\,M2}+1). \quad (5)$$

### 4.1.4 | Actions

In every decision epoch, the decision-maker agent authorizes one of the following actions based on the system's state: produce, maintain, and idle. The feasible action set in every state is:

- *If the number of final parts reaches the maximum storage capacity $I_{max}$, the "maintain" or "idle" action can only be performed for machine $M_2$.*
- *The maintenance and repair activities are the only options if the production agents are in "breakdown" state.*
- *All the actions are feasible if the inventory of the final parts is less than the maximum allowed capacity and the production agent is in "readyformessage" state.*

#### 4.1.5 | Reward

Let $A = \{A_1, A_2, A_3, \dots\}$ denotes the actions that the decision-maker agent takes in each decision epoch. The agent receives a numerical reward in each decision epoch regarding the previously performed action by

$$R = C_{h_b} + C_{h_I} + C_s + C_p + C_m + C_w + C_b + C_l - S_p. \quad \textbf{(6)}$$

Where

$C_{h_b}$: The holding cost of the semi-final parts stored in the buffer.

$C_{h_I}$: The holding cost of the final parts stored in the storage.

$C_s$: The cost of backordered orders.

$C_p$: The cost of production.

$C_m$: The cost of maintenance and repair activities.

$C_w$: The cost of producing wastages.

$C_b$: The cost of production machine breakdown.

$C_l$: The cost of missed orders.

$S_p$: The sales profit of the final parts.

The main goal of the decision-maker agent is to obtain a policy that maximizes the total acquired rewards (minimizing *Eq. (6)*).

## 4.2 | Reinforcement Learning

In this research, the optimal joint policy of the decision-maker agent is achieved by a RL algorithm called R-learning [22]. The R-learning algorithm has been applied in recent research [7], [8].

Let $t_d = \{t_{d,1}, t_{d,2}, t_{d,3}, \dots\}$ denotes the decision epochs that the decision-maker agent takes action and $R_{t_{d,i}}$ is the obtained reward in decision epoch i. The R-learning seeks to maximize the following equation:

$$\rho = \lim_{n\to\infty} \frac{1}{n} E\left\{\sum_{i=1}^{n} R_{t_{d,i}}\right\}. \quad \textbf{(7)}$$

In the R-learning, the value of the performed action in the state (state-action) is calculated by

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha\left[R_{t+1} - \varrho + \max_a Q(S_{t+1},a) - Q(S_t,A_t)\right]. \quad \textbf{(8)}$$

Where

$Q(S_t,A_t)$: The value of action A in state S in time t.

$\alpha$: Learning rate.

$R_{t+1}$: The obtained reward regarding the performed action A in state S in time t.

$\max_a Q(S_{t+1},a)$: The action that yields the maximum value in state S in time t+1.

Also, the following equation updates the average reward $\rho$:

$$Q(S_t,A_t) \leftarrow Q(S_t,A_t) + \alpha\left[R_{t+1} - \varrho + \max_a Q(S_{t+1},a) - Q(S_t,A_t)\right]. \quad \textbf{(9)}$$

Where $\beta$ is a real-valued parameter between 0 and 1. The decision-maker agent applied the $\varepsilon$-greedy "policy" to take action. In this policy, the agent chooses the action that yields the maximum value by the probability $1 - \varepsilon$, and selects the action randomly with the probability $\varepsilon$.

It is worth mentioning that in methods such as dynamic programming, the state transition probability matrix is needed to solve the MDP. However, in many real-world problems, it is impossible to calculate the state transition probabilities matrix of the system. In this case, the RL algorithms are helpful because there is no need to calculate the transition probabilities. The algorithm uses the simulation model to act and observe the next state of the system and reward.

## 4.3 | Simulation-Optimization

The term "SO" refers to techniques used to optimize stochastic problems in parametric optimization [23]. Specifically, it involves searching for the optimal values of input parameters in a simulation model to achieve a specific objective.

The integrated production, maintenance, and quality control in this research can be represented as a discrete parametric optimization problem. The input parameters in this context refer to the set of feasible actions available in each state of the system. SO techniques can be employed to find the best action in each state, maximizing the total reward or objective function. In this research, the SO package is utilized to find the optimal or near-optimal values for the combined optimization problem. The SO package integrates various metaheuristic approaches, including scatter search, tabu search, and neural networks, into a single optimization procedure, enabling efficient and effective optimization. The SO process is shown in *Fig. 5*.

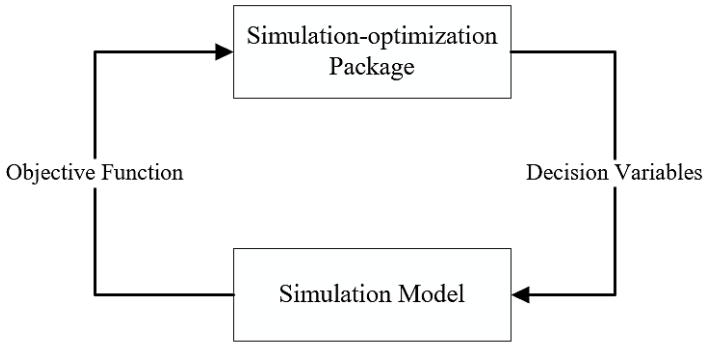


**Fig. 4. The process of SO.**

# 5 | Numerical Results

Four scenarios evaluate the efficiency of the proposed method. The first scenario is the base case, as illustrated in *Table 2*. In the second scenario, the efficiency of the policies to decrease missed orders is examined. In the third scenario, the effect of increasing the production rate is studied. The efficiency of the policies in reducing missed orders and wastages is evaluated in Scenario 4.

In the "policy learning" phase, the RL-based decision-maker agent acquires the optimal joint production, maintenance, and quality policy using the agent-based simulation of the production system of each scenario. Next, in the "policy evaluation" phase, the decision-maker agent selects the action based on the obtained policy in the simulation model of the scenarios. By the Monte-Carlo, the policy will be evaluated, the simulation model is iterated for a certain number of runs, and a unique random seed performs each iteration to capture all the events.

Two alternatives evaluate the RL-based policy: the random decision-maker agent and the SO method.

As discussed in Section 5, the decision-maker agent uses the ε-greedy policy to select an action in every decision epoch. If the value of ε is set to be 1, the agent chooses the action randomly in all decision epochs. This case is considered as an alternative to evaluating the policies.

The SO method is also studied to evaluate the policies. The SO package observes the simulation model of each scenario as a black box, and the system's states are defined as the input parameters of the simulation model. In each iteration, the state-action pairs are set as decision variables to minimize the obtained cost.

**Table 2. The parameter's value of the production system.**

| | | | | | |
|---|---|---|---|---|---|
| **Environment** | $I_{max}$ | | | | 10 |
| | $B_{max}$ | | | | 10 |
| | $S_{max}$ | | | | 10 |
| | $C_{h_I}$ | | | | 1 |
| | $C_{h_b}$ | | | | 3 |
| | $C_s$ | | | | 5 |
| | $C_w$ | | | | 15 |
| | $C_l$ | | | | 50 |
| | $S_p$ | | | | 10 |
| **Machine $M_1$** | d | (0,1,2,3) | **Machine** $M_2$ | d | (0,1,2,3) |
| | $\lambda_f$ | $20\times\lambda_p$ | | $\lambda_f$ | $20\times\lambda_p$ |
| | $\lambda_p$ | 1.5 | | $\lambda_p$ | 2 |
| | $p_{b,d_i}$ | (0,0.05,0.2,0.8) | | $p_{b,d_i}$ | (0,0.06,0.25,0.8) |
| | $p_{d_i}$ | (0,0.1,0.5,1) | | $p_{d_i}$ | - |
| | $p_{d_i,q_l}$ | - | | $p_{d_i,q_l}$ | (0,0.2,0.5,0.9) |
| | $p_{d_i,q_h}$ | - | | $p_{d_i,q_h}$ | (0,0.1,0.25,0.6) |
| | $C_p$ | 0.5 | | $C_p$ | 0.5 |
| | $C_m$ | 100 | | $C_m$ | 120 |
| | $C_b$ | 150 | | $C_b$ | 200 |
| **The Agent-Based Model** | | Shift duration | | | 12 Hours |
| | | Model execution time | | | 2160 Hours |

Due to the stochasticity of the simulation model, the model is replicated for a certain number of runs, and the average obtained costs are considered as the objective function value. The best state-action pairs that yield the minimum total cost are the acquired optimal joint policy by the SO package. In *Table 3*, the input parameters of the RL algorithm, the Monte Carlo method, and the SO package are illustrated.

**Table 3. The input parameter's value.**

| | | |
|---|---|---|
| **RL** | Number of episodes | 25000 |
| | Number of steps | 2160 Hours |
| **Monte-Carlo** | Number of Iterations | 5000 |
| | The simulation model execution time | 2160 Hours |
| **SO Package** | Number of iterations | 25000 |
| | Minimum number of replications | 30 |
| | Maximum number of replications | 100 |
| | The simulation model execution time | 2160 Hours |

## 5.1|Scenario 1-Base Case

In Scenario 1, the input parameters of the agent-based simulation are set by the values of *Table 2*. The RL-based decision-maker agent observes the state of the system and authorizes the action. In the next step, the new state of the system and the corresponding reward (*Eq. (6)*) is returned. This process continues until the optimal policy is obtained. The acquired reward by the agent in each episode is shown in *Fig. 6*.

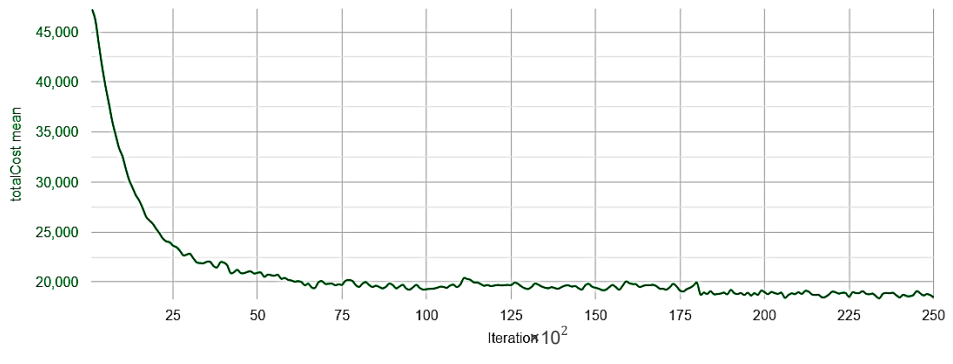


**Fig. 5. The obtained reward by the decision-maker agent in scenario 1.**

It is observed that the decision-maker agent initially receives a reward of about 47000 due to the random action selection. However, the policy gradually became goal-oriented and finally converged to the reward of about 19000. To evaluate the policy, the agent's decision-making process in the simulation model is implemented according to the acquired policy, and the Monte Carlo method is used for evaluation. The results are presented in *Table 4*.

**Table 2. The results obtained by the RL-based policy in scenario 1.**

| Title | Average | Std. Dev |
|---|---|---|
| Total cost | 18,884.25 | 1,900.18 |
| Number of performed maintenance activities on machine $M_1$ | 44.699 | 1.168 |
| Number of performed maintenance activities on machine $M_2$ | 44.609 | 1.14 |
| Number of breakdowns of machine $M_1$ | 8.134 | 2.556 |
| Number of breakdowns of machine $M_2$ | 8.428 | 2.607 |
| Number of missed orders | 165.097 | 31.549 |
| Number of wastages | 15.135 | 4.039 |

In order to evaluate the policy obtained by the decision-maker agent, the random decision-making policy is implemented in the agent-based simulation model. A comparison is given in *Table 5*.

**Table 3. The comparison of the RL-based and random policy in scenario 1.**

| Title | RL-Based Decision-Maker Agent | Random Decision-Maker Agent |
|---|---|---|
| Number of performed maintenance activities on machine $M_1$ | 44.699 | 58.452 |
| Number of performed maintenance activities on machine $M_2$ | 44.609 | 58.801 |
| Number of breakdowns of machine $M_1$ | 8.134 | 4.277 |
| Number of breakdowns of machine $M_2$ | 8.428 | 4.685 |
| Number of missed orders | 165.097 | 603.038 |
| Number of wastages | 15.135 | 8.092 |

The random decision-maker agent performed more maintenance and repair activities than the RL-based decision-maker agent, resulting in reduced produced wastage. However, the number of missed orders by the random policy is significantly higher than the RL-based policy. The RL-based policy performed maintenance and repair activities on time and succeeded in reducing 438 units of missed orders and not drastically increasing the wastage.

## 5.2|Scenario 2- Decreasing the Missed Orders

In this scenario, the cost associated with missed orders is denoted as $C_l$ is ten times higher than the base case. This intentional adjustment aims to incentivize the RL-based algorithm to minimize missed orders effectively. The achieved rewards by the agent in each episode are depicted in *Fig. 7*.

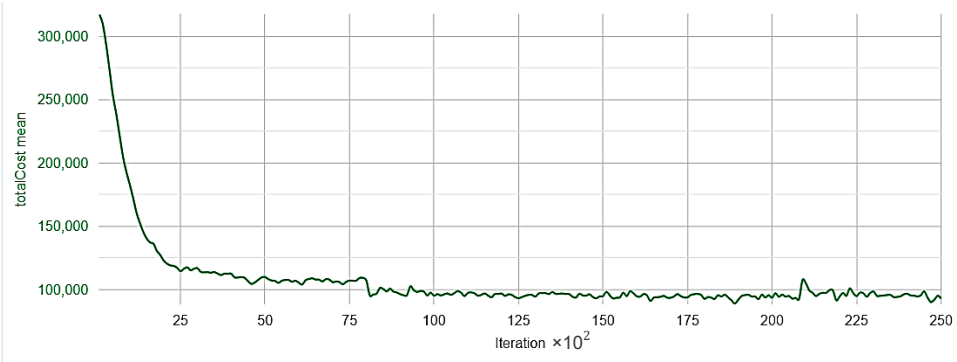


**Fig. 6. The obtained reward by the decision-maker agent in scenario 2.**

The agent receives a reward of about 320000 at the beginning of episodes, But with more learning, the reward converged to 93000. The results of the policy evaluation by the Monte-Carlo method are presented in *Table 6*.

**Table 4. The results obtained by the RL-based policy in scenario 2.**

| Title | Average | Std. Dev |
|---|---|---|
| Total cost | 93,775.34 | 16,349.80 |
| Number of performed maintenance activities on machine $M_1$ | 42.927 | 1.948 |
| Number of performed maintenance activities on machine $M_2$ | 44.724 | 1.168 |
| Number of breakdowns of machine $M_1$ | 16.281 | 3.29 |
| Number of breakdowns of machine $M_2$ | 8.584 | 2.561 |
| Number of missed orders | 155.145 | 32.159 |
| Number of wastages | 15.955 | 4.138 |

The RL-based policy decreases missed orders compared to the base case. But the number of breakdowns of the machine $M_1$ has increased. It means that the machine $M_1$ must produce even at a high level of degradation to satisfy the demand.

The comparison between the RL-based policy and the random policy is presented in *Table 7*.

**Table 5. The comparison of the RL-based and random policy in scenario 2.**

| Title | RL-Based Decision-Maker Agent | Random Decision-Maker Agent |
|---|---|---|
| Number of performed maintenance activities on machine $M_1$ | 42.927 | 58.529 |
| Number of performed maintenance activities on machine $M_2$ | 44.724 | 58.866 |
| Number of breakdowns of machine $M_1$ | 16.281 | 4.269 |
| Number of breakdowns of machine $M_2$ | 8.584 | 4.631 |
| Number of missed orders | 155.145 | 603.481 |
| Number of wastages | 15.955 | 8.048 |

Although the RL-based decision-maker agent decreases the missed orders, it is not significant. The main reason is that the production rate of machine $M_1$ and $M_2$, by considering the required maintenance and repair activities, can not meet the demand. In the following scenario, the effect of increasing the production rate is examined.

## 5.3|Scenario 3-Increasing the Production Rate

In order to increase the possibility of satisfying the demand, the production rate is tripled. The cost of missed orders is the same as in the previous scenario. In this case, the acquired reward by the agent in each episode is shown in *Fig. 8*.

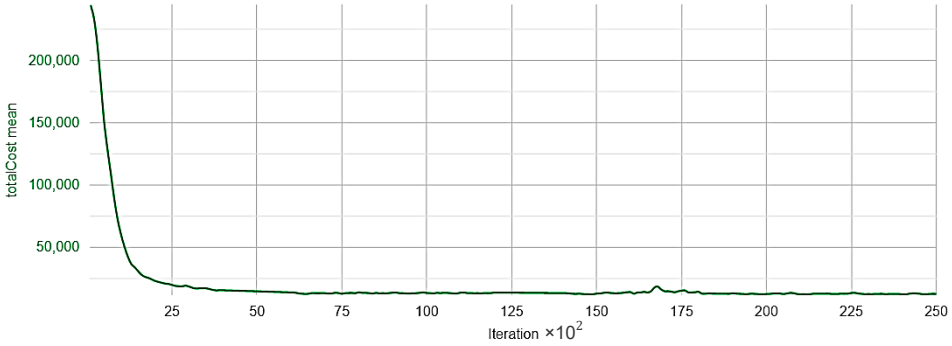


**Fig. 7. The obtained reward by the decision-maker agent in scenario 3.**

In scenario 3, the agent initially receives a reward of about 250000, but over time, the reward converges to 14000. The results of the policy evaluation by the Monte-Carlo method are presented in *Table 8*.

**Table 6. The results obtained by the RL-based policy in scenario 3.**

| Title | Average | Std. Dev |
|---|---|---|
| Total cost | 14,256.19 | 4,274.82 |
| Number of performed maintenance activities on machine $M_1$ | 56.095 | 2.24 |
| Number of performed maintenance activities on machine $M_2$ | 56.083 | 1.936 |
| Number of breakdowns of machine $M_1$ | 18.613 | 3.762 |
| Number of breakdowns of machine $M_2$ | 11.937 | 3.299 |
| Number of missed orders | 5.727 | 7.513 |
| Number of wastages | 21.553 | 5.65 |

The RL-based policy succeeded in decreasing missed orders by almost 96% compared to scenario 2. The number of maintenance, breakdowns, and wastages has also increased due to the increased production.

The comparison between the RL-based policy and the random policy is provided in *Table 9*.

**Table 7. The comparison of the RL-based and random policy in scenario 3.**

| Title | RL-Based Decision-Maker Agent | Random Decision-Maker Agent |
|---|---|---|
| Number of performed maintenance activities on machine $M_1$ | 56.095 | 62.92 |
| Number of performed maintenance activities on machine $M_2$ | 56.083 | 62.842 |
| Number of breakdowns of machine $M_1$ | 18.613 | 10.889 |
| Number of breakdowns of machine $M_2$ | 11.937 | 10.871 |
| Number of missed orders | 5.727 | 470.454 |
| Number of wastages | 21.553 | 18.571 |

As it is observed, the performance of the RL-based decision-maker agent is impressive in the current scenario. The increased production gives more flexibility to the agent in selecting the time of production, maintenance, and repair activities or being idle so that the total cost is minimized. *Table 9* shows that although there is no significant difference in the number of maintenance and repair activities performed between the two agents, the RL-based decision-making agent's missed orders are much lower. However, the increase in production rate has not led to a significant increase in the number of wastages, which has been due to the timely authorized maintenance and repair activities by the RL-based decision-maker agent.

## 5.4 | Scenario 4-Decreasing the Wastages

In this scenario, the algorithm's efficiency in maintaining the quality of the final parts and reducing the number of wastages is examined. So, the probability of producing low-quality semi-finished parts in machine $M_1$ at different degradation levels are increased. In addition, the probability of making waste from high-quality and

low-quality semi-final parts at different degradation levels of machine $M_2$ has also increased. The new values are presented in *Table 10*.

**Table 10. The quality-related probabilities in scenario 4.**

| Quality-Related Probabilities | The Production Machine | New Values |
|---|---|---|
| $p_{d_i}$ | Machine $M_1$ | (0, 0.3, 0.7, 1) |
| $p_{d_i,q_l}$ | Machine $M_2$ | (0, 0.3, 0.8, 0.9) |
| $p_{d_i,q_h}$ | Machine $M_2$ | (0, 0.2, 0.6, 0.8) |

Therefore, a large percentage of the final parts will be wasted in the case of insufficient maintenance and repair activities. Also, to jointly optimize the production, maintenance, and quality at the same time, the cost of missed order is the same as the third scenario, and the cost of wastage $C_w$ is examined in two cases: 15 and 500. The yielded reward by the agent in each episode is presented in *Fig. 9* ($C_w$=15) and *Fig. 10* ($C_w$=500).

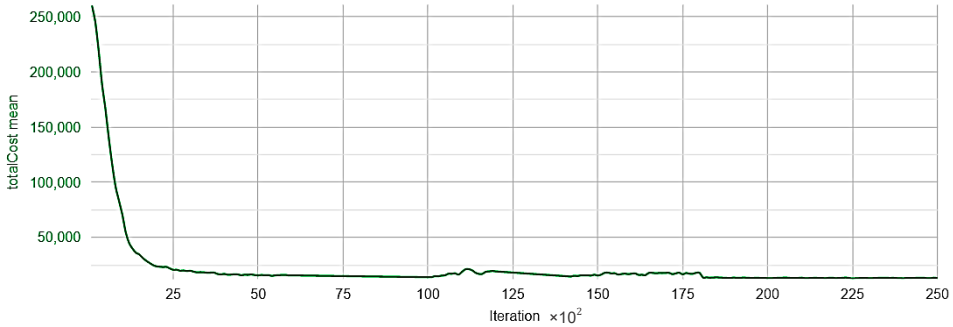


**Fig. 8. The obtained reward by the decision-maker agent in scenario 4 ($C_w$=15 ).**

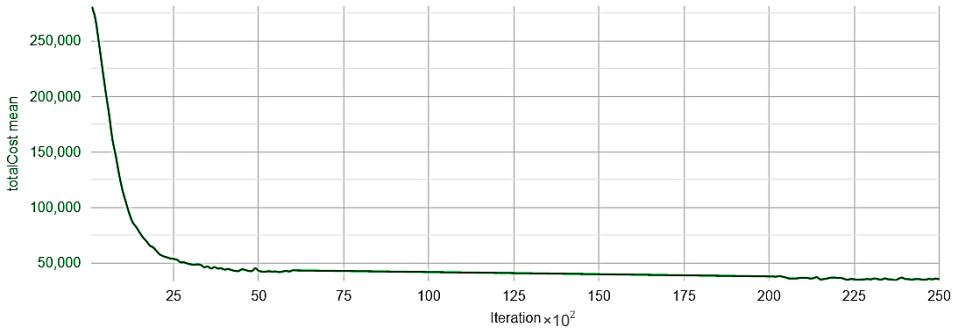


**Fig. 9. The obtained reward by the decision-maker agent in scenario 4 ($C_w$=500 ).**

Similar to the previous scenarios, the performance of the RL-based decision-maker agent and the random decision-maker agent are compared in *Table 11*.

**Table 8. The comparison of the RL-based and random policy in scenario 4.**

| Title | Scenario 4-1 ($C_w$=15) | | Scenario 4-2 ($C_w$=500) | |
|---|---|---|---|---|
| | RL-Based Decision-Maker Agent | Random Decision-Maker Agent | RL-Based Decision-Maker Agent | Random Decision-Maker Agent |
| Number of performed maintenance activities on machine $M_1$ | 57.845 | 62.926 | 58.049 | 62.992 |
| Number of performed maintenance activities on machine $M_2$ | 58.13 | 63.067 | 57.909 | 63.065 |
| Number of breakdowns of machine $M_1$ | 18.75 | 11.056 | 18.846 | 10.977 |
| Number of breakdowns of machine $M_2$ | 12.415 | 11.052 | 11.721 | 11.028 |
| Number of missed orders | 2.199 | 488.505 | 2.852 | 489.384 |
| Number of wastages | 47.048 | 39.222 | 44.876 | 38.969 |

It can be seen that although the increased probability of producing wastages leads to a decrease in final products, the RL-based agent has succeeded in balancing the number of missed orders and wastages in the current scenario.

## 5.5 | The Policy Evaluation by SO

Meta-heuristic methods have been widely used to obtain optimal or near-optimal policies for different mixes of production, maintenance, and quality problems [3], [16], [17], [20]. In this research, in addition to the Random Agent (RA), the SO method is also used as an alternative to evaluating the obtained policy by the RL-based decision-maker agent. The SO package is initialized by related parameters (*Table 3),* and the connection of the package and simulation model of each scenario is established, as shown in *Fig. 5*. Similar to the evaluation process of RL-based policy, the acquired joint policy by SO package is evaluated by the Monte Carlo method. The results are shown in *Fig. 11*.

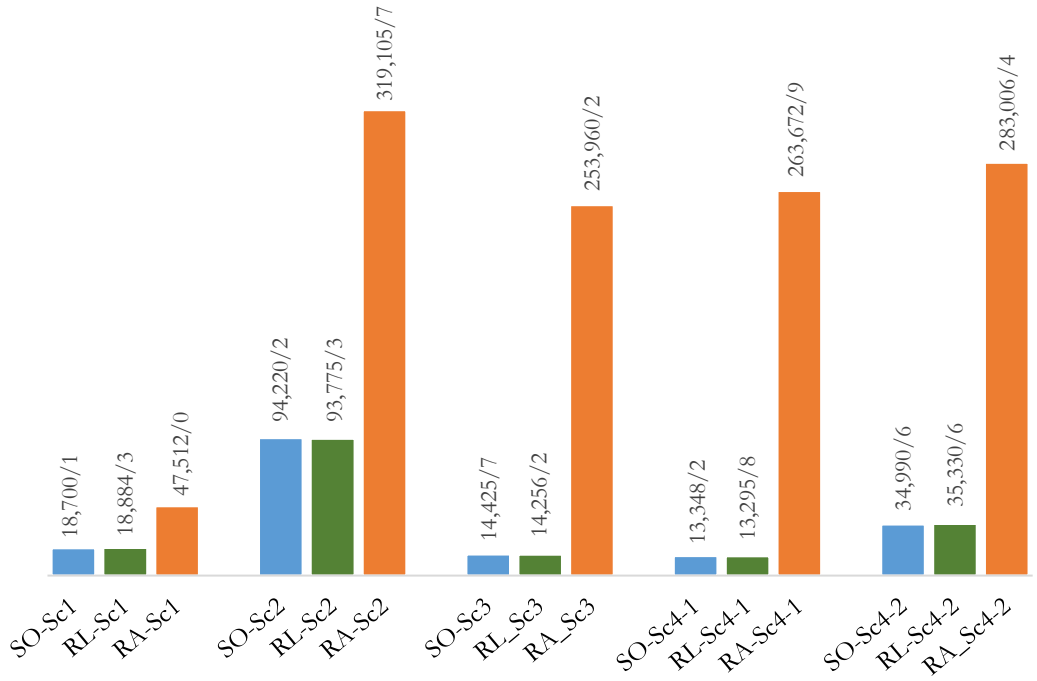


**Fig. 10. The comparison between SO, RL, and RA policies.**

The results indicate that the decision-maker agent, through RL, has successfully achieved an optimal or near-optimal policy. Its performance is very close to, and in some cases, even better than, the SO approach, and both methods have superior performance in minimizing the cumulative reward of *Eq. (6)*. SO methods, due to the use of metaheuristic algorithms, provide near-optimal solutions, which serve as a suitable benchmark for evaluating the performance of other proposed methods. Therefore, it can be concluded that the derived joint policy through the RL algorithm can be an effective policy for the real-time decision-making process in manufacturing systems. RL can provide the best action regarding production, maintenance, repair, and quality in various states of the production system.

# 6 | Conclusion

Production planning, maintenance, and quality control are always the most critical challenges of production systems. Regarding mutual interactions, it is necessary to investigate these issues jointly to achieve the optimal integrated policy of the system.

This paper examined the joint optimization of production, maintenance, and quality of the multi-machine single-product system with an intermediate buffer and final product. The ABM approach was applied to simulate the production system, and an RL-based agent was designed to interact with the simulation model

to obtain the combined policy. The random policy and the meta-heuristic methods in the form of the SO approach were used to evaluate the acquired RL-based optimal policy.

Four scenarios were considered to cover all aspects of the production system. In each scenario, the performance of the policies in authorizing maintenance and repair activities, reducing missed orders, and reducing wastage were examined. The results showed that RL-based policy has superior performance in minimizing production costs, maintenance costs, inventory costs, lost orders, breakdown of production machines, and low-quality production. By the RL-based achieved joint policy, the decision-maker agent authorizes the most proper action in each system's state. Thus, it can be used for an autonomous real-time decision-making process necessary for industry 4.0.

The results also showed that the RL algorithm has a high potential to solve problems defined in the Markov Decision Process (MDP). Dynamic programming is another approach to solving the MDPs, but the transition probabilities matrix is required, which is very hard or impossible to define in many real-world problems. In addition, the Curse-of-dimensionality is another challenge, specifically when the system's state space is numerous.

SO methods prove their efficiency in solving the complex optimization problem. However, it is computationally expensive to obtain a joint policy in large states and action spaces. It is necessary to assign an action to every state of the system as a decision variable in each iteration. However, with the advent of DRL, agents can now bypass exhaustive state exploration by integrating RL algorithms with neural networks. By harnessing the learning capabilities of neural networks, RL agents can adapt and make decisions even in unencountered states. This advantage positions RL as a promising approach to finding the joint optimal policy. However, there are also some limitations, such as reward design, stability, and convergence issues when applying RL. So, acquiring optimal policies for diverse problem domains relies crucially on selecting the most fitting methodology.

Finally, this research can be extended by considering the multi-machine multi-product system. In this case, DRL or Multi-agent RL is proposed to find the joint optimal or near-optimal policy because the state of the system will be dramatically increased. Dealing with multiple machines and products simultaneously introduces a more complex and extensive state space, making traditional RL methods less effective. By leveraging DRL techniques, such as Deep Q Networks (DQNs) or Policy Gradient methods, the agent can handle high-dimensional state representations and learn more sophisticated strategies. Alternatively, Multi-agent RL can be employed to model interactions and dependencies among multiple machines and products, allowing the agents to coordinate and collectively optimize the system performance.

# References


[1] CEN, E. (2001). *EN 13306: maintenance terminology*. European Committee For Standardization. https://dl.mpedia.ir/e-books/18-[BSI]BS-EN-13306-2010-maintenance-terminology[mpedia.ir].pdf

[2] Liu, Q., Dong, M., & Chen, F. F. (2018). Single-machine-based joint optimization of predictive maintenance planning and production scheduling. *Robotics and computer-integrated manufacturing*, *51*, 238–247.

[3] Rivera Gómez, H., Gharbi, A., Kenné, J. P., Montaño Arango, O., & Corona Armenta, J. R. (2020). Joint optimization of production and maintenance strategies considering a dynamic sampling strategy for a deteriorating system. *Computers & industrial engineering*, *140*, 106273. https://doi.org/10.1016/j.cie.2020.106273

[4] Sutton, R. S., Barto, A. G., & others. (1998). *Introduction to reinforcement learning* (Vol. 135). MIT press Cambridge.

[5] Zheng, W., Lei, Y., & Chang, Q. (2017). Comparison study of two reinforcement learning based real-time control policies for two-machine-one-buffer production system. *2017 13th ieee conference on automation science and engineering (CASE)* (pp. 1163–1168). IEEE.

[6] Kuhnle, A., Jakubik, J., & Lanza, G. (2019). Reinforcement learning for opportunistic maintenance optimization. *Production engineering*, *13*, 33–41.

[7] Xanthopoulos, A. S., Kiatipis, A., Koulouriotis, D. E., & Stieger, S. (2017). Reinforcement learning-based and parametric production-maintenance control policies for a deteriorating manufacturing system. *IEEE access*, *6*, 576–588.

[8] 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. *Journal of manufacturing systems*, *56*, 470–483.

[9] Yang, H., Li, W., & Wang, B. (2021). Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning. *Reliability engineering & system safety*, *214*, 107713. https://doi.org/10.1016/j.ress.2021.107713

[10] Huang, J., Chang, Q., & Arinez, J. (2020). Deep reinforcement learning based preventive maintenance policy for serial production lines. *Expert systems with applications*, *160*, 113701. https://doi.org/10.1016/j.eswa.2020.113701

[11] 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. *Expert systems with applications*, *192*, 116323. https://doi.org/10.1016/j.eswa.2021.116323

[12] Zhao, Y., & Smidts, C. (2022). Reinforcement learning for adaptive maintenance policy optimization under imperfect knowledge of the system degradation model and partial observability of system states. *Reliability engineering & system safety*, *224*, 108541. https://doi.org/10.1016/j.ress.2022.108541

[13] 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. *Reliability engineering & system safety*, *236*, 109290. https://doi.org/10.1016/j.ress.2023.109290

[14] Lavoie, P., Gharbi, A., & Kenne, J.-P. (2010). A comparative study of pull control mechanisms for unreliable homogenous transfer lines. *International journal of production economics*, *124*(1), 241–251.

[15] Bouslah, B., Gharbi, A., & Pellerin, R. (2018). Joint production, quality and maintenance control of a two-machine line subject to operation-dependent and quality-dependent failures. *International journal of production economics*, *195*, 210–226.

[16] Rivera Gomez, H., Gharbi, A., & Kenné, J. P. (2013). Joint production and major maintenance planning policy of a manufacturing system with deteriorating quality. *International journal of production economics*, *146*(2), 575–587.

[17] Tambe, P. P., & Kulkarni, M. S. (2022). A reliability based integrated model of maintenance planning with quality control and production decision for improving operational performance. *Reliability engineering & system safety*, *226*, 108681. https://doi.org/10.1016/j.ress.2022.108681

[18] Borshchev, A., & Filippov, A. (2004). From system dynamics and discrete event to practical agent based modeling: reasons, techniques, tools. *Proceedings of the 22nd international conference of the system dynamics society* (pp. 25–29). Oxfort England.

[19] Jennings, N. R. (2000). On agent-based software engineering. *Artificial intelligence*, *117*(2), 277–296.

[20] Bonabeau, E. (2002). Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the national academy of sciences*, *99*(3), 7280–7287.

[21] Macal, C. M., & North, M. J. (2010). Tutorial on agent-based modelling and simulation. *Journal of simulation*, *4*, 151–162.

[22] Schwartz, A. (1993). A reinforcement learning method for maximizing undiscounted rewards. *Proceedings of the 10th international conference on machine learning* (Vol. 298, pp. 298–305). Morgan Kaufmann Publishers. DOI: 10.1016/b978-1-55860-307-3.50045-9

[23] Gosavi, A., & Gosavi, A. (2015). Control optimization with reinforcement learning. In *Simulation-based optimization: parametric optimization techniques and reinforcement learning* (pp. 197–268). Springer.