7th International Conference on Reliability and Safety Engineering (ICRSE2023)

Optimal Preventive Maintenance Policy for non-Indentical Components: Traddition Renewal Theory vs Modern Reinforecment Learning

Shaghayegh Eidi[1], Abdollah Safari [1*], Firoozeh Haghighi [1]

1-Department of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran

Corresponding author, a.safari@ut.ac.ir

Abstract

In this paper, we compare traditional approach aginst reinforcement learning algorithms to find the optimal preventive maintenance policy for equipments composed of multi-non-identical components with different time-to-failure distributions. As an application, we used the data from military trucks which consisted of multiple components with very different failure behaviour such as tires, transmissions, wheel rims, couplings, motors, brakes, steering wheels, and shifting gears. Four different strategies have been proposed for preventive maintance of these components in the literature. To find the optimal preventive maintancence policy, we used both traditional approach (renewal theory based) and the convetional reinforcement learning algorithms and compared their performance. The main advantages of the latter approach is that, unlike the traditional approach, they are not required to estimate the model parameters (e.g., transition probabilities) and without any explicit mathematical formula, they converge to the optimal solution. Our results showed that when the compoenent time-to-falirue distributions are available, the traditional approach works best. However, where no such information is available or the distributions are misspecified, the reinforcement learning approach outperforms.

Keyword: Opportunistic maintenance; Preventive maintenance; Markov decision process; Monte Carlo; Q-learning; Reinforcement learning

Introduction

In the recently released European Standards regarding maintenance, maintenance is defined as the combination of all technical, administrative and managerial actions during the life cycle of an item intended to retain it in, or restore it to, a state in which it can perform the required function; see Marquez and Gupta [1]. Maintenance problems can be solved by using traditional approaches as well as machine learning methods. In recent years, the use of reinforcement learning (RL) algorithms has become very popular and widely used. RL is one of the newer approaches of machine learning that has gained prominence in various fields of human life today. In general, RL is a technique that allows a decision-making (agent) to maximize his total reward by interacting with the environment. Ravichandiran [2] introduced the steps of a typical RL algorithm as follows:

1. First, the agent interacts with the environment by performing an action
2. The agent performs an action and moves from one state to another
3. And then the agent will receive a reward based on the action it performed
4. Based on the reward, the agent will understand whether the action was good or bad
5. If the action was good, that is, if the agent received a positive reward, then the agent will prefer performing that action again, otherwise, the agent will try performing an other action which results in a positive reward. So it is basically a trial and error learning process.

As mentioned earlier, RL algorithms are used a lot in most fields at present. Wang et al. [3] applied multi-agent RL to solve the maintenance problem for a flow line system consisting of two series machines with an intermediate finite buffer in between. Liang et al. [4] modelled the energy management problem by a Markov

decision process and solve it by using an Approximate Dynamic Programming (ADP)-based approach to match electricity supply and demand. Yousefi et al. [5] used an RL approach to develop a new dynamic maintenance policy for multi-component systems with individually repairable components, where each component is at risk of two competing failure processes of degradation and random shocks. Adsule et al. [6] modelled the Condition-based maintenance (CBM) decision-making as a continuous semi-Markov decision process and applied an RL algorithm to learn the optimal maintenance decisions and inspection schedule based on the current health state of the component.

In this paper, we consider military trucks composed of multi-non-identical components. Trucks are systems that are used continuously, so the possibility of them breaking down on the road is very high, which can result in financial and life-threatening costs. Additionally, trucks are used in the military, so minimizing the downtime of any truck is essential. It is important to efficiently obtain the optimal replacement times for each component of the system. Haleem and Yacout [7] and Barde et al. [8] tackled this problem before. They used the truck's eight more important components in their analysis: tires, transmissions, wheel rims, couplings, motors, brakes, steering wheels, and shifting gears. We will use the same set of components here as well. Abdel Haleem and Yacout [7] used renewal theory to estimate the components' replacement times. Barde et al. [8] estimated replacement times by using Monte Carlo reinforcement learning (MCRL). We aimed to optain the optimal maintenance policy by using the two existing approaches in the literature as well as employing a time difference (TD) learning approach, which is a more efficient RL algorithm than MCRL. We will evaluate the performance of all these approaches under two scenarios: when the true failure time distributions are available versus they are misspecified.

In the next section, we will present the problem assumptions and existing maintanence strategies; In Method section, we will present different algorithms. Finally, in Results section, we will report the numerical results comparing the TD-based RL algorithm's performance agains the two other methods proposed in the literature.

Motivation

We consider an equipment that contains multiple non-identical components. Our purpose in this article is to find a policy that minimizes the total downtime of the equipment. The downtime is defined as the non-productive time, which is the time that the system is not operational due to a failure or a preventive action. Each component of the equipment has a different time to failure distribution that is modeled by a Weibull distribution its own shape and scale parameters. The strategies are base on the following assumptions:

1. If we replace a component due to failure, it takes more time than if we replace a component preventively.
2. If we replace a group of components or a whole system, it takes less time than if we replace each component separately.
3. There are replacement opportunities at regular intervals.

The aforementioned assumptions can be found in many military applications, where the equipment's reliability is essential, downtime must be minimized, and cost considerations are less important; see Haleem and Yacout [7].

Following Haleem and Yacout [7], the following four replacement strategies will be used and compared agains one another:

- **Strategy I**: Every component is replaced upon failure. It is a corrective maintenance (baseline).
- **Strategy II**: every component is replaced upon failure and at an individual fixed interval, T_i, for component i. It is based on a preventive maintenance. Haleem and Yacout [7] estimated T_i by minimizing the downtime per unit time, D_i, for component i. D_i is calculated from the following expression:

$$argmin_{T_i} D_i = \frac{tp_i R(T_i) + tf_i[1 - R(T_i)]}{(T_i + tp_i)R(T_i) + [tf_i + E(t|t \le T_i)][1 - R(T_i)]}, \forall i \quad (1)$$

Where tp_i is time to replace component i preventively, tf_i is time to replace component i upon failure, $R(T_i)$ is the reliability of component i at time T_i and $E[t|t \le T_i]$ is the expected time to failure given that it occurs before T_i.

- **Strategy III**: It is based on Strategy II, to which it is added a scheduled overhaul. In other words, as Strategy II, every component is replaced at failure and at replacement intervals T_i for component I, but also, the whole system is replaced at a known fixed time.
- **Strategy IV**: It is a group-based maintenance strategy. Any component i that fails or reaches its replacement interval T_i, the components of its group are also replaced with it.

Markov Decision Process

A Markov decision process (MDP) framework has the following key components:

1. S: Set of states ($s \in S$)
2. A: Set of actions ($a \in A$)
3. $P(s_{t+1}|s_t, a_t)$: Transition probabilities
4. $R(s, a)$: Reward function of doing action a in state s.

We use model-free RL due to two reasons: curse of dimensionality and curse of modeling. The curse of dimensionality arises from the much longer computational time and much larger memory space needed as the state space of a problem become larger. The curse of modeling arises from the need to estimate the transition probabilities, which is often difficult to estimate, especially when the state space is large; see Powell [9]. Now we present MDP formulation (state, action and reward function) for each preventive strategy (i.e., II, III and IV).

MDP formulation of strategy II : Let G_i be age of component i, $f_i = 1$ denotes that component i is failed and $f_i = 0$ denotes its normal status, so the state of the system at time t is the vector defined as follows:

$$s_t = (G_1, \dots, G_8, f_1, \dots, f_8).$$

Let a_i = 1 means PM action and a_i = 0 means "do nothing" action, then the action of the system at time t is:

$$a_t = (a_1, \dots, a_8).$$

Base on Barde et al. [8] work, the reward function can be defined as follows:

$$R(s_t, a_t) = \begin{cases} -\alpha_i . tp_i , & if\ a_i = 1 \\ -\alpha_i . \Delta . \left\lceil \frac{tf_i}{\Delta} \right\rceil, & if\ a_i = 0\ and\ f_i = 1 \\ \Delta , & otherwise \end{cases}$$

Where $\alpha_i = \frac{\Delta}{tp_i}$ is a scale factor as they assumed that tp_i is scaled such that it has at least the same time period than Δ (see Barde et al. [8] for more information); Δ is time interval between two epochs and $\lceil . \rceil$ is the ceiling function.

MDP formulation of strategy III : Let G_i be age of component i, $f_i = 1$ denotes that component i is failed whearas $f_i = 0$ denotes normal status and $O = 1$ means replace whole system whereas $O = 0$ denotes that don't replace whole system, then, the state of the system at time t is the vector defined as follows:

$$s_t = (G_1, \dots, G_8, O, f_1, \dots, f_8).$$

The action on the system is defined as:

$$a_t = (a_1, \dots, a_8).$$

The reward function is:

$$R(s_t, a_t) = \begin{cases} -\alpha_i . tp_i, & if\ a_i = 1 , O = 0 \\ -\alpha_i . \Delta . \left\lceil \frac{tf_i}{\Delta} \right\rceil, & if\ a_i = 0, O = 0, f_i = 1 \\ -\alpha_i . \Delta . \left\lceil \frac{\beta . \sum_{i=1}^{8} tp_i}{\Delta} \right\rceil, & if\ O = 1 \\ \Delta, & otherwise \end{cases}$$

Where $\beta \in (0\ , 1)$ comes from the assumption which states that the time to replace the whole system is less than the sum of times to replace each component separately.

MDP formulation of strategy IV: states and actions in Strategy IV is the same as those in Strategy II, but reward function is different due to group structure. The components are grouped as follows:
$(\phi_1, \phi_2, \phi_3, \phi_4, \phi_5) = (\{1,3\}, \{3,8\}, \{3,5\}, \{7,6\}, \{4,2\})$
The groups are formed based on technical reasons such as the difficulty or ease of reaching and changing a component when a neighboring component has failed. Then, the reward function is:

$$R(s_t, a_t) = \begin{cases} -\alpha_i . \beta . \sum_{l \in \phi_j} tp_l , & if\ a_k = 1\ and\ k \in \phi_j \\ -\alpha_i . \Delta . \left\lceil \frac{\beta . \sum_{l \in \phi_j} tf_l}{\Delta} \right\rceil, & if\ f_k = 1,\ \alpha_k = 0\ , k \in \phi_j \\ \Delta, & otherwise \end{cases}$$

Where $\alpha_i = \frac{\Delta}{\beta . \Delta . \sum_{l \in \phi_j} tp_l}$ and $\beta \in (0,1)$ is defined similarly as in Strategy III.

Reinforcement Learning

Barde et al. [8] used on-policy first visit MCRL algorithm to find the optimal replacement time T_i for each preventive strategy separately whearas we will use a TD learning approach that is more efficient than MCRL.

TD learning is a model-free approach that combines sampling and bootstrapping simoltanously. One of the advantages of TD over MCRL is that MCRL can only be used for episodic problems. In other word, MCRL learns from complete episodes only. Unlike MCRL, TD learning employes single steps to learn (be updated after every step) and does not need to wait until the end of an episode. Therefore, TD learning can be applied to both countinuing and episodic problems.

In this paper, for the military trucks problem, we will use a Q-learning (QL) algorithm, which is an off-policy TD control algorithm. QL is one of the most popular and efficient algorithms in RL. It is an off policy RL algorithm because the QL function learns from actions that are not necessarily taken under the current agent

policy. Similar to other RL algorithms, QL seeks to learn a policy that maximizes a pre-defined total reward in every state. The objective of QL algorithm is to learn and estimate optimal action-value function that defined as

$$Q^*(s_t, a_t) = \max_{\pi} Q_\pi(s_t, a_t),$$

Where

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} | \, s_t, a_t]$$

QL directly approximates optimal action-value function by taking the best action when bootstrapping:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R_{t+1} + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)], \quad (2)$$

Where $\alpha \in (0,1)$ is the learning rate that controls the importance of the old against learned value, $\gamma \in (0,1)$ is the discount factor determines how much importance we give to future rewards compared to the immediate reward R_{t+1}; see Sutton and Barto [10]. The algorithm's steps are shown in Fig. 1.

One RL crucial element is the trade-off between exploitation and exploration. Exploration consists of the agent trying all the possible actions at least once in order to make better action selection in the future, whereas exploitation consists of the agent using its current knowledge to obtain the highest reward. For achieving this balance, we use an ε-greedy policy to take the optimal actions. The ε-greedy approach selects the action with the highest estimated reward most of the times ($(1-\varepsilon) \times 100$ % of the times).

Initialize $Q(s,a)$ arbitrarily
Repeat (for each episode):
  Initialize s
  Repeat (for each step of episode):
    Choose a from s using policy derived from Q
    Take action a, observe r, s'
    Update
      $Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \max_{a'} Q(s',a') - Q(s,a)]$
    $s \leftarrow s'$;
  Until s is terminal

Fig. 1.Q_learning (off-policy TD control)

We chose ε in such a way that in the initial episodes, the agent starts to explore, gather information, and as time goes on and more information about the environment is collected by the agent, ε vanishes. Finally, when the agent acquired "enough knowledge", it will soley takes actions to maximize its reward (no exploration). Also, to ensure convergence to the optimal value, we chose ε as $\varepsilon_t = \frac{1}{t}$ for the t^{th} episode where both assumptions of $\sum_{t=0}^{\infty} \varepsilon_t^2 < \infty$ and $\sum_{t=0}^{\infty} \varepsilon_t = \infty$ are hold; see Tsitsiklis [11].

If the learning rate (α) is set to zero, the action-value function does not be updated and therefore, there will be no learning for the agent. If one chooses the learning rate be near to one, the learning process will be very quick; Therefore, we update the learning rate after each episode as follows:

$$\alpha(t) = \max(0.1, \ \min(1, 1 - \log(\frac{t+1}{\gamma}))),$$

where γ is a problem-specific decay parameter and needs to be chosen by trial and error.

Results

Scenario 1: True failure time distributions are available

It is assumed that each component failure probability is Independent from others, the algorithm searches for the optimal action-value function for each component, and T_i that corresponds to the age where the value of the action 'replace preventively' is higher than the value of the action 'do nothing'.

Let $P(t, \lambda_i, k_i)$ be the probability density function of Weibull, λ_i the scale parameter, and k_i the shape parameter of the distribution for the i^{th} component. Table 1 reports the component-specific Wibull distribution parameters as well as tp_i and tf_i.

Table 1.Components failure time distribution

	Tire	Transmission	Wheel	Coupling	Motor	brake	Steering	Gears
mean	14.06	5.903	4.218	8.332	2.039	23.32	4.868	12.13
$\boldsymbol{\lambda_i}$ (scale)	14.076	5.934	4.248	8.373	2.046	23.41	4.93	12.148
$\boldsymbol{k_i}$ (shape)	378.17	108.917	79.65	115.829	170.756	143.747	43.953	278.507
$\boldsymbol{tp_i}$	0.0024	0.032	0.0037	0.0051	0.0074	0.0042	0.0026	0.0052
$\boldsymbol{tf_i}$	0.012	0.039	0.015	0.036	0.03	0.021	0.018	0.021

The time interval between every two decision epochs is assumed to be 5 hours. This value is chosen because the probability that two components will fail during this time interval is approximately zero. We used a similar simulation settings as the one proposed by Barde et al' paper [8] to estimate each approach downtime for each strategy. Comparison of the performance of each strategy is performed between the traditional method, the MRCL and the QL approaches.

In strategy II, agent is learnt the optimal maintenance policy to interact with the environment in ~400 episode by applying MCRL whearas, the agent is learnt the optimal policy in interacting with the environment in ~200 episode by applying QL algorithm. Table 2 demonstrates optimal replacement time for strategy II (in weeks) by using the three mentioned approaches. As can be seen, there is a slight difference between the optimal replacement times in all three approaches; however, the optimal replacement times of the MC algorithm seem to be slightly higher than those of the other two approaches.

Table 2. Optimal replacement times (in weeks) for Strategy II

Component Name	Traditional	Q-learning	MCRL
Tire	13.809	13.780	13.988
Transmission	5.770	5.804	8.860
Wheel	3.964	3.928	4.137
Coupling	7.917	7.827	8.125
Motor	1.970	2.024	2.024
Brake	22.381	22.292	22.798
Steering	4.339	4.226	4.643
Gears	11.875	11.875	11.905

Table 3 and Fig. 2 illustrates a performance comparison among the three approaches in Strategy II. It can be seen that the traditional method has a total downtime of 7.454 week with 16 failed components and 867 preventive replaced components due to preventive actions. Those numbers are 7.574 week, 25 failed components, and 860 preventive replaced components for the QL algorithm and 8.129 week, 68 failed components, and 806 preventive replaced components for the MC algorithm. The QL approach outperfmoed the MC approach, its performance seems to be similar to the traditional approach with the traditional approach has has a slightly lower system downtime and number of failed components.

Table 3. System downtime (in weeks), number of failed and replaces component of each approach for Strategy II

	Traditional	Q-learning	MCRL
System downtime	7.454	7.574	8.129
number of failed component	16	25	68
Number of prevention action	867	860	806

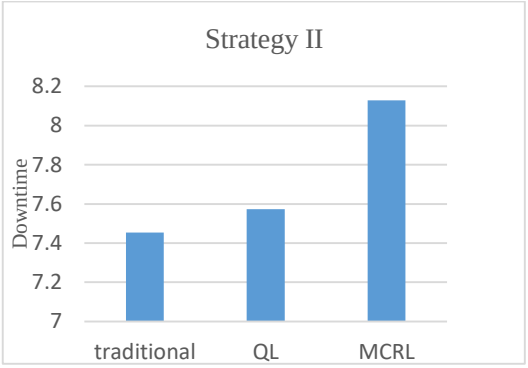


Fig. 2.System downtime (in weeks) of different appcoaches for Strategy II

In Strategy III, the agent finds the optimal policy in interacting with the environment in ~400 and ~250 episodes with MC and QL algorithms, respectively. Table 4 reports the optimal replacement times of each component by suing different appproasches in Strategy III.

Table 4. Optimal replacement times (in weeks) for Strategy III

Component Name	Traditional	Q-learning	MCRL
Tire	13.809	13.810	13.839
Transmission	5.770	5.804	5.923
Wheel	3.964	3.988	4.137
Coupling	7.917	7.738	8.185
Motor	1.970	2.024	2.024
Brake	22.381	22.530	23.036
Steering	4.339	4.137	4.643
Gears	11.875	11.964	12.054

Table 5 shows total system downtime of the three approaches in Strategy III. The traditional approach achieves the lowest downtime of 7.657 weeks at a scheduled overhaul of 21 weeks, with 9 failed components, and 956 preventive replaced components due to preventive action; The QL algorithm achieves the lowest downtime of 7.763 weeks at a scheduled overhaul of 21 weeks, with 12 and 985 failed and preventive replaced components; Finally, the MC algorithm achieves the lowest downtime of 8.985 weeks at a scheduled overhaul of 15 weeks, with 120 and 912 failed and preventive replaced components. In Strategy III, the traditional method has the lowest overall downtime compared to the other two approaches. The difference between average downtime of QL and that of the traditional approach was less than one day (~18 hours). However, such difference between the MC and the traditional approaches was as high as one week.

Table 5. System downtime (in weeks) obtained by different approaches for Strategy III with different overhaul times

Schedule overhaul	System downtime		
	Traditional	Q-learning	MCRL
3	10.987	11.054	11.121
6	10.159	10.334	11.223

9	8.515	8.898	9.821
12	8.673	8.635	9.638
15	7.934	8.040	8.985
18	8.273	8.457	9.731
21	7.657	7.763	9.339
24	8.230	8.160	9.561
27	7.731	7.909	9.553
30	7.903	7.981	9.505

Table 6, Table 7 and Fig. 3 illustrates similar comparion results among the three approaches for Strategy IV. The agent finds the optimal policy in interacting with the environment in ~1000 and ~200 episodes in MC and QL algorithms. Table 6 reports the optimal replacement times for different components by the three approaches in Strategy IV. The replacement times estimated by using the RL algorithms are much lower than the replacement times obtained through the traditional approach. The reduction in replacement times is due to the group structure in this strategy. Components with a lower mean failure time dominate the overall replacement time of their fellow components. It can be seen that in the Table 7 that the traditional approach has a total downtime of 8.375 week with 27 failed components and 1237 preventive replaced components due to preventive action. The overall system downtime were 9.117 week (with 39 failed and 1348 preventive replaced components) and 10.815 week (with 84 failed and 1618 preventive replaced components) for the QL and MC algorithms. In Strategy IV, the traditional approach has the lowest downtime, the lowest number of failed components, and also has fewer preventive replacements than the other two approaches (which will lead to lower maintenance costs).

Table 6. Optimal replacement times (in weeks) for Strategy IV

Component Name	Traditional	Q-learning	MCRL
Tire	13.809	4.167	4.226
Transmission	5.770	5.804	5.893
Wheel	3.964	2.024	1.994
Coupling	7.917	5.832	5.893
Motor	1.970	2.024	1.994
Brake	22.381	4.375	4.643
Steering	4.339	4.375	4.643
Gears	11.875	4.167	4.167

Table 7. System downtime (in weeks), number of failed and replaces component of each approach for Strategy IV

	Traditional	Q-learning	MCRL
System downtime	8.375	9.117	10.815
number of failed component	27	39	84
Number of prevention action	1237	1348	1618

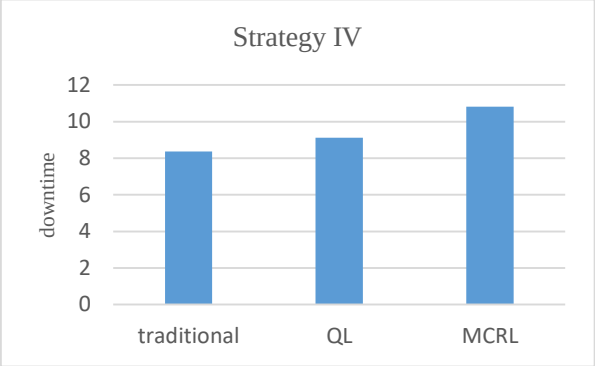


Fig. 3.System downtime (in weeks) for Strategy IV

According to the Table *8*, Table *9* and Table *10*, Strategy I is the worst strategy as it has longest downtime. This shows the clear advantage of the preventive strategies over the corrective maintenance strategies. Overall, the most efficient strategy among these proposed strategies is Strategy II.

Table 11 reports the average execution time to obtain the optimal policy by each approach and in strategy. As it can be seen, the traditional appraoch was the most time-efficient approach (less than a second). After the tradditional approach, the QL algorithm was about twice faster than the on-policy first visit MC algorithm. As expected, the more complex a strategy is, the more time requires to obtain its optimal policy.

Table 8. Evaluation of traditional approach for different strategies

	Strategy I	Strategy II	Strategy III	Strategy IV
System downtime	21.366	7.454	7.657	8.375
number of failed component	842	16	9	27
Number of prevention action	0	867	956	1237

Table 9. Evaluation of QL algorithm for different strategies

	Strategy II	Strategy III	Strategy IV
System downtime	7.574	7.762	9.117
number of failed component	25	12	39
Number of prevention action	860	985	1348

Table 10. Evaluation of MC algorithm for different strategies

	Strategy II	Strategy III	Strategy IV
System downtime	8.129	8.985	10.815
number of failed component	68	120	84
Number of prevention action	806	912	1618

Table 11.Execution time to converge to the optimal policy by different approaches in different strategies

	Strategy II	Strategy III	Strategy IV
Tradditional	< 1 sec	< 1 sec	< 1 sec
QL	2 min	2 min	22 min
MC	4 min	3 min	53 min

Scenario 2: Misspecified failure time distributions are available

The results reported in Scenario 1 are under the assumption of knowing the environment and therefore, the true failure time distribution of different components were available. However, under a more realistic scenario, the true failure time distribution of the components may not be available.

In this section, we evaluate the performance of the three approaches under a misspecified failure time distribution of the components. More specifically, we assumed the true distribution of failuire time of the components remain Weibull with the same parameters reported in Table 1**.** However, a misspecified Weibull distribution (either its shape or scale parameter is overestimated by different degrees) is assumed for each component while finding the optimal policy by each approach.

The RL free-model algorithms require no assumptions for the environment and the agent interacts with the environment directly (data driven). That is, they will not be impacted by the misspecified failure time distribution.

Table 12 reports the optimal replacement times obtained through "Eq. (1)" by assuming an overestimated Weibull shape parameter. As shown in Fig. 4, Fig. 7 and Fig. 6 , the misspecification of the Weibull shape parameter does not seem to have a larage impact on the estimated optimal replacement time of the components by the tradditonal approach. In the estimated system downtime, when using the optimal replacement time of the traditional approach, seemed to be impactd differently in different startegies. Specifically, the traditional system downtime in Strategy II and III were slightly impacted (increased by 1.5-3 days) only when the shape parameter is overestimated by at least 20%.

Table 12.Optimal replacement times with minor changed in the Weibull shape parameters.

Component Name	Shape (real)	Shape +10%	Shape +20%	Shape +30%	Shape +40%	Shape +50%
Tire	13.81	13.85	13.86	13.88	13.89	13.90
Transmission	5.77	8.78	5.79	5.80	5.80	5.81
Wheel	3.96	3.99	4.01	4.02	4.03	4.04
Coupling	7.92	7.95	7.98	8	8.02	8.04
Motor	1.97	1.98	1.99	1.99	1.99	1.99
Brake	22.38	22.47	22.54	22.60	22.67	22.68
Steering	4.34	4.39	4.42	4.45	4.48	4.50
Gears	11.88	11.89	11.90	11.92	11.94	11.95

Fig. 5, Fig. 8 and Fig. 9 illustrate system downtime obtained by the traditional approach comparing with that of RL algorithms. As the figures show, the Weibull scale parameter overestimation from 2% to 5% increased the system downtime by 1 to 8 weeks in different strategies. Moreover, Table 13 shows how minor changes in the Weibull scale parameter affects the optimal replacement times that obtained through "Eq. (1)".

Table 13.Optimal replacement times with minor changed in the Weibull scale parameter.

Component Name	Scale (real)	Scale +1%	Scale +2%	Scale +3%	Scale +4%
Tire	13.81	13.96	14.11	14.23	14.38
Transmission	5.77	5.83	5.89	5.92	6
Wheel	3.96	3.99	4.05	4.08	4.12
Coupling	7.92	7.98	8.07	8.16	8.21
Motor	1.97	1.99	2.01	2.03	2.05
Brake	22.38	22.62	22.86	23.07	23.27
Steering	4.34	4.39	4..43	4.49	4.51
Gears	11.88	11.99	12.11	12.20	12.34

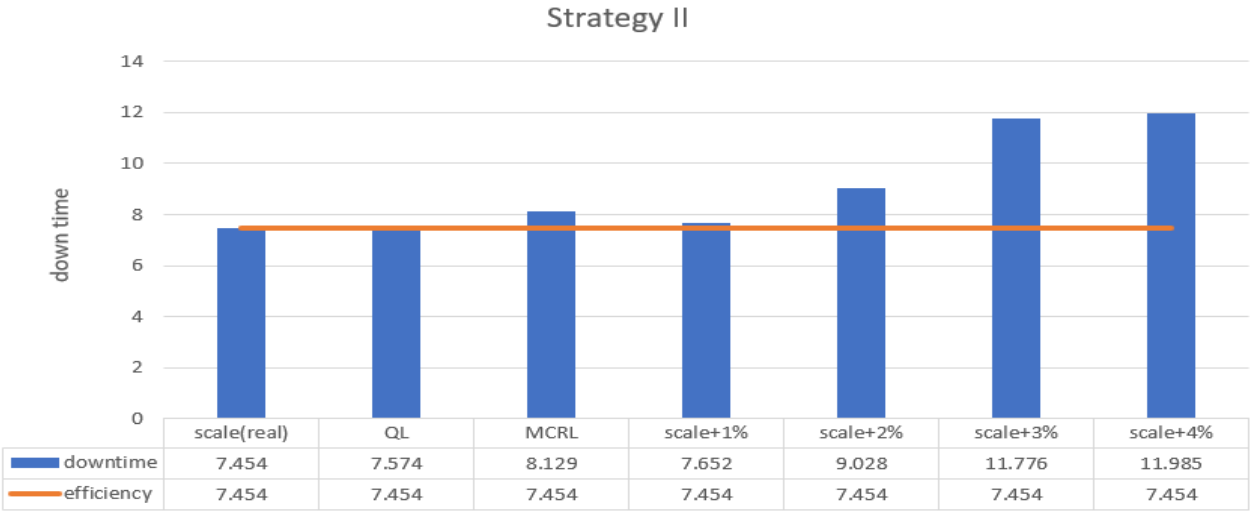
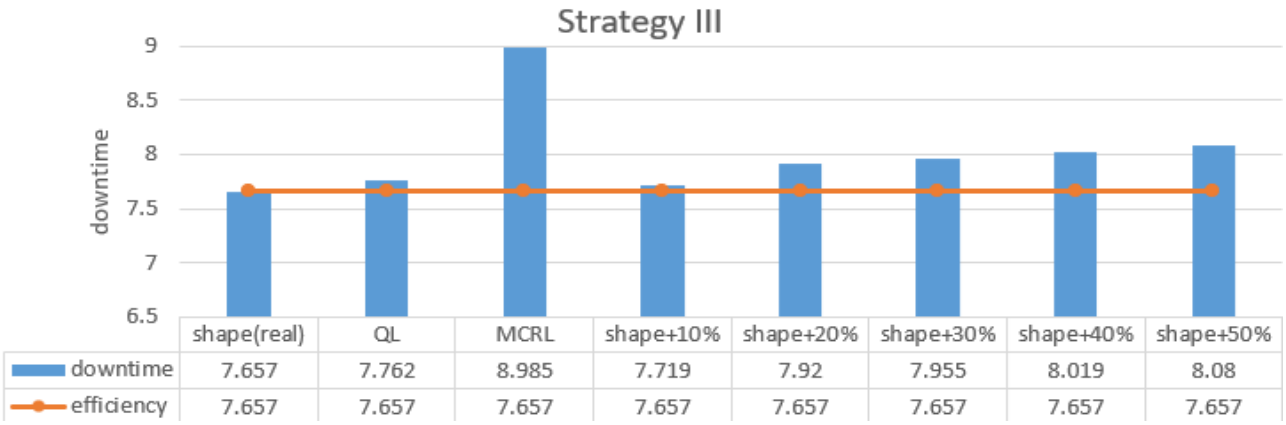


Fig. 4.System downtime of different approaches under minor changes in the Weibull shape parameters for Strategy II

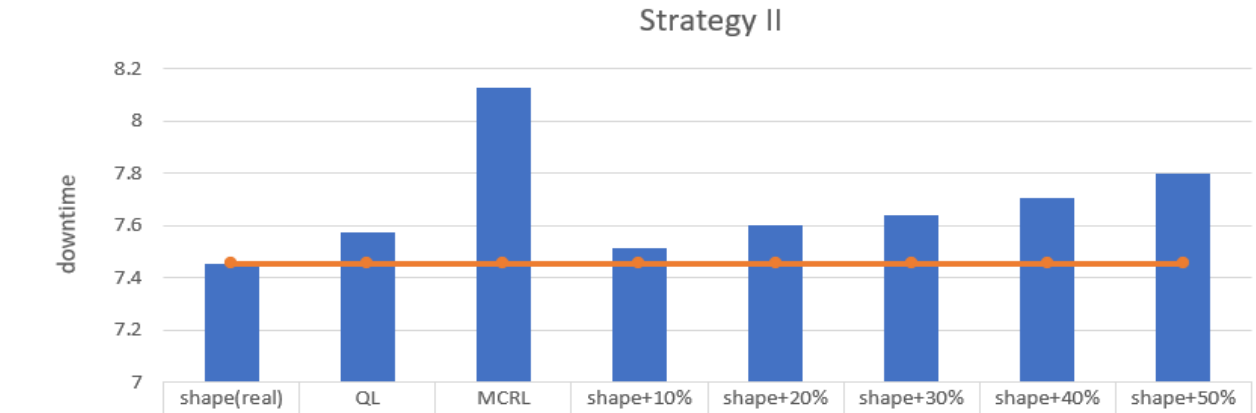


	shape(real)	QL	MCRL	shape+10%	shape+20%	shape+30%	shape+40%	shape+50%
downtime	7.657	7.762	8.985	7.719	7.92	7.955	8.019	8.08
efficiency	7.657	7.657	7.657	7.657	7.657	7.657	7.657	7.657

Fig. 7.System downtime of different approaches under minor changes in the Weibull shape parameters for Strategy III

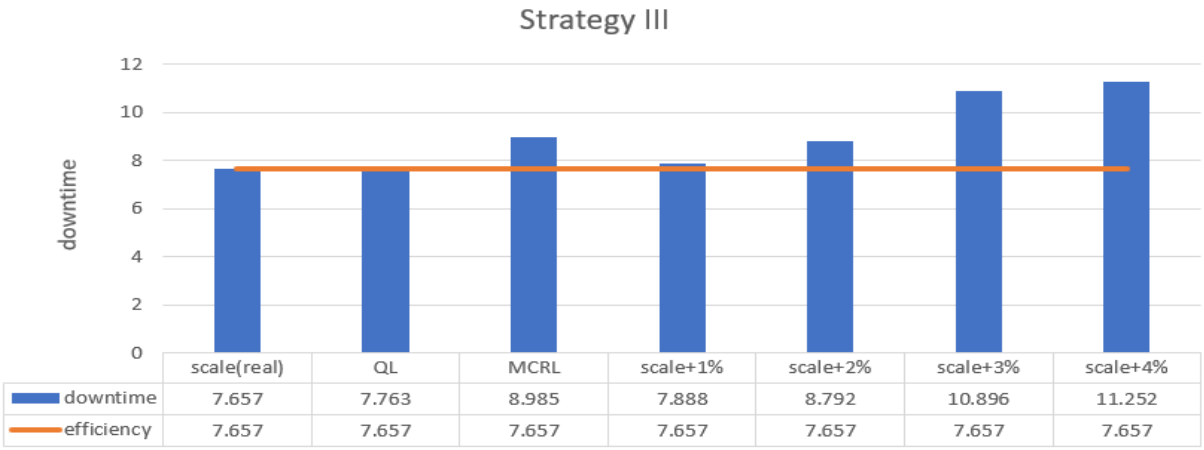
	shape(real)	QL	MCRL	shape+10%	shape+20%	shape+30%	shape+40%	shape+50%
downtime	8.375	9.117	10.815	8.592	8.69	8.768	8.776	9.115
efficiency	8.375	8.375	8.375	8.375	8.375	8.375	8.375	8.375

Fig. 6.System downtime of different approaches under minor changes in the Weibull shape parameters for Strategy IV



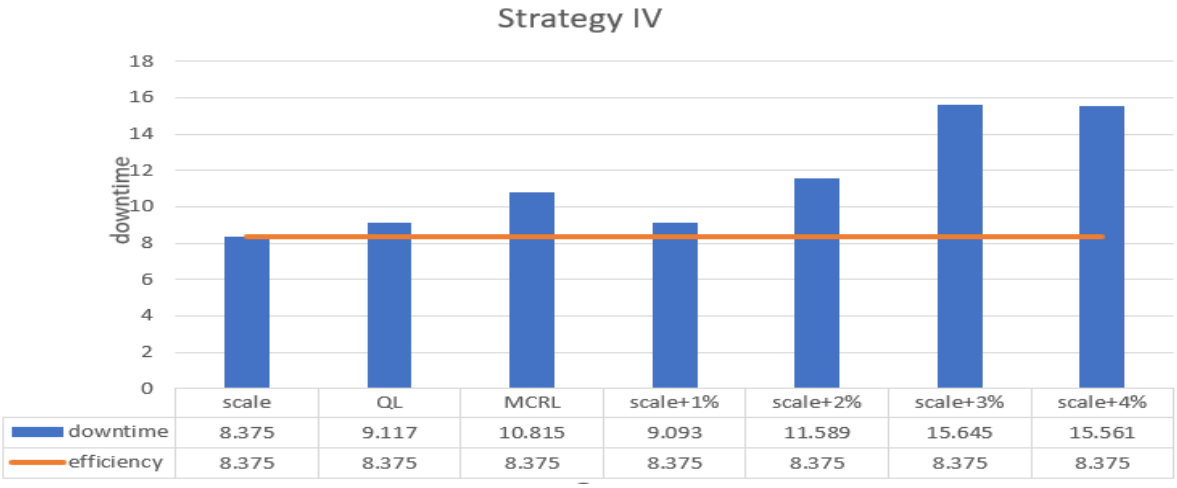
	shape(real)	QL	MCRL	shape+10%	shape+20%	shape+30%	shape+40%	shape+50%
downtime	7.454	7.574	8.129	7.515	7.599	7.64	7.707	7.797
efficiency	7.454	7.454	7.454	7.454	7.454	7.454	7.454	7.454

Fig. 5.System downtime of different approaches under minor changes in the Weibull scale parameters for Strategy II



	scale(real)	QL	MCRL	scale+1%	scale+2%	scale+3%	scale+4%
downtime	7.657	7.763	8.985	7.888	8.792	10.896	11.252
efficiency	7.657	7.657	7.657	7.657	7.657	7.657	7.657

Fig. 8.System downtime of different approaches under minor changes in the Weibull scale parameters for Strategy III



	scale	QL	MCRL	scale+1%	scale+2%	scale+3%	scale+4%
downtime	8.375	9.117	10.815	9.093	11.589	15.645	15.561
efficiency	8.375	8.375	8.375	8.375	8.375	8.375	8.375

Fig. 9.System downtime of different approaches under minor changes in the Weibull scale parameters for Strategy IV

Discussion

In this work, we employed three approaches of traditional (renewal theory), MCRL, and TDRL in order to find the optimal preventive maintenance policy for an equipment composed of multi-non-identical components. Three preventive maintenance strategies along with a corrective maintenance strategy (as baseline) were studied. Our results confirmed that preventive maintenance strategies perform better than the corrective maintenance policy, as expected, for our system. More importantly, our results showed that the tradditional approach (renewal theory) is sensitive to the misspecification of the components' failure time distribution. More specifically, under the assumption of the components Weibull distributed failure times, the optimal policy and consequently the performance of the traditional approach seem to be impacted only slightly by misspecifying the shape parameters up to 50% (downtime increased by < 3 days). However, even minor misspecification in the scale parameter (up to 5%) can lead to a huge increase in the system downtime following the traditional approach optimal policy by up to 8 weeks. On the other hand, since the model-free RL algorithms are data driven with no requirements of prior assumption on the environment distribution (e.g., failure time distributions), they can be minimally impacted by such misspecifications.

Different RL algorithms, however, can potentially perform very differently. Under the assumptions of our study, QL algorithm outperformed MC algorithm dramatically. Given the quick progress in developing RL algorithms nowdays, a natural next step to our work might be evaluating different RL algorithms for different systems with different assumptions.

References

[1] A. C. Marquez and J. N. Gupta, "Contemporary maintenance management: process, framework and supporting pillars," *Omega*, vol. 34, no. 3, pp. 313--326, 2006.

[2] S. Ravichandiran, Hands-on reinforcement learning with Python: master reinforcement and deep reinforcement learning using OpenAI gym and TensorFlow, Packt Publishing Ltd, 2018.

[3] X. Wang, H. Wang and C. Qi, "Multi-agent reinforcement learning based maintenance policy for a resource constrained flow line system," *Journal of Intelligent Manufacturing*, vol. 27, pp. 325--333, 2016.

[4] Y. Liang, T. Deng and Z. J. Max shin, "Demand-side energy management under time-varying prices," *IISE Transactions*, vol. 51, no. 4, pp. 422--436, 2019.

[5] N. Yousefi, S. Tsianikas and D. W. Coit, "Reinforcement learning for dynamic condition-based maintenance of a system with individually repairable components," *Quality Engineering*, vol. 32, pp. 388--408, 2020.

[6] A. Adsule, M. Kulkarni and A. Tewari, "Reinforcement learning for optimal policy learning in condition-based maintenance," *IET Collaborative Intelligent Manufacturing*, vol. 2, no. 4, pp. 182--188, 2020.

[7] B. Abdel Haleem and S. Yacout, "Simulation of components replacement policies for a fleet of military trucks," *Quality Engineering*, vol. 11, no. 2, pp. 303--308, 1998.

[8] S. R. Barde, S. Yacout and H. Shin, "Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks," *Journal of Intelligent Manufacturing*, vol. 30, pp. 147--161, 2016.

[9] W. B. Powell, Approximate Dynamic Programming: Solving the curses of dimensionality, vol. 703, John Wiley \& Sons, 2007.

[10] R. S. Sutton and A. G. Barto, Reinforcement learning: An introduction, MIT press, 2018.

[11] J. N. Tsitsiklis, "On the convergence of optimistic policy iteration," *Journal of Machine Learning Researc*, vol. 3, pp. 59--72, 2002.