A Reinforcement Learning approach to optimal part flow management for gas turbine maintenance

Michele Compare[1,2,*], Luca Bellani[2], Enrico Cobelli[1], Enrico Zio[1,2,3,4], Francesco Annunziata[5], Fausto Carlevaro[5], Marzia Sepe[5]

[1]Energy Department, Politecnico di Milano, Italy

[2]Aramis S.r.l., Italy

[3]MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France

[4]Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea

[5]Baker Hughes, a GE company (BHGE), Florence, Italy

[*]corresponding author; michele.compare@polimi.it

Abstract

We consider the maintenance process of Gas Turbines (GTs) used in the Oil & Gas industry: the capital parts are first removed from the GTs and replaced by parts of the same type taken from the warehouse; then, they are repaired at the workshop and returned to the warehouse, for use in future maintenance events. Experience-based rules are used to manage the flow of the parts, for a profitable GT operation. In this paper we formalize the part-flow management as a Sequential Decision Problem (SDP) and propose Reinforcement Learning (RL) for its solution. An application to a scaled-down case study derived from real industrial practice shows that RL can find policies outperforming those based on experience-based rules.

Keywords: Part Flow; Reinforcement Learning; Gas Turbine

Symbols & Acronyms

DM	Decision Maker
GT	Gas Turbine

MRC	Most Residual Cycles
MS	Maintenance Shutdown
PFM	Part Flow Management
RL	Reinforcement Learning
RUL	Remaining Useful Life
SDP	Sequential Decision Problem
γ	Discount factor
$\mathbf{S}_k$	State vector at the k-th MS, $\mathbf{S}_k = [S_{k,1}, ..., S_{k,R+1}]$
A_k	Action taken at the k-th MS
$a_{k,\rho}$	Boolean variable equal to 1, if action ρ is taken at the k-th MS and 0 otherwise
$C^{rep}(r)$	Repair cost for a part with r maintenance cycles remaining
C^{scrap}	Cost of scrapping a part
C_k	Cost incurred at the k-th MS
d_k	RUL of the part removed from the GT maintained at the k-th MS
G	Total number of GTs
g	Index of the GT undergoing maintenance at the k-th MS
k	MS index, $k = 1, ..., T$
N	Total number of RL episodes
n	Index of RL episodes, $n = 1, ..., N$
$Q_\pi(\mathbf{S}_k, A_k)$	State-Action pair value following policy π from the k-th MS on
R	Maximum RUL
r	RUL index
T	Total number of scheduled MS
V	Total value of the maintenance expenditures
W	Maximum number of parts that can be stored at the warehouse
$w_{r,k}$	Number of parts with RUL=r available at the warehouse at the k-th MS

Z	Number of scheduled MSs for each GT

1 Introduction

Gas Turbines (GTs) employed in the Oil & Gas industry are made up of various expensive capital parts (e.g., buckets, nozzles, shrouds, etc.), which are affected by different degradation mechanisms (e.g., fracture and fatigue (1), (2), (3), fouling (4), (5), (6), corrosion (7),(8), oxidation (9)) that can lead to GT failures with costly forced outages.

Given the criticality of the GT degradation processes, attentive engineering analyses have been performed to characterize their behaviors. These studies, corroborated by economic considerations, have yielded the definition of the preventive maintenance policy, detailed in (10), determining both the optimal length of the working cycles before scheduled Maintenance Shutdowns (MSs) and the maximum number of these working cycles that every type of capital part can perform, provided that it is repaired after each cycle. The repaired parts are put back in the warehouse, ready to be installed at one of the next MS of a GT in the same Oil & Gas plant.

From the above, it clearly emerges that the management of GT maintenance is a very complex issue, which requires a specific expertise for performing the intricate procedures for GT disassembling and re-assembling, an effective logistic organization for managing the spares (i.e., their ordering, shipping, etc.), a deep knowledge about the degradation processes affecting the parts for their effective repair, etc. (e.g., see (11) for an overview). Whilst GT manufacturers are usually structured for addressing all these issues, their customers may not be fully qualified to do so. This is among the main justifications of the increasing diffusion of maintenance service contracts between the GT manufacturers (i.e., the maintenance service providers) and the GT owners (i.e., the recipients of the service) (12; 13).

Service contracts yield new business opportunities to GT manufacturers, who can sell the GTs production rates, instead of selling the GTs, with consequent added values if they assume portions of the clients' business risks ((12)). To do this, however, GT manufacturers need to develop effective and efficient maintenance strategies and spare part inventory management policies ((14; 15)).

In particular, effective strategies are required to manage the periodic MSs, where decisions must be made on both the removed part (send it to the workshop for repair or scrap it) and the part to be installed on the GT (new part or part taken from the warehouse). These decisions strongly impact on the profitability of the GT maintenance service contract, as they determine both the direct costs incurred by the service provider for repairing the parts and the indirect costs from the risk of forced outages due to GT failures, which entail penalties to the maintenance service provider. For example, scrapping old parts reduces risk and workshop costs but increases the number of purchase actions taken by the maintenance service provider. Furthermore, at the end of the maintenance service contract the warehouse may contain healthy parts ready for installation, whose value is lost by the service provider.

The parts installed on the GTs are no longer available at the warehouse for replacement at the next MS and when they return to the warehouse (if not scrapped), they do so with a reduced number of remaining working cycles. Thus, the decisions at every MS influence the decisions at the next MSs: in this sense, the Part Flow Management (PFM) can be framed as a Sequential Decision Problem (SDP)(16), seeking for the best sequence of future maintenance decisions (i.e., the optimal policy) over the duration of maintenance service contract. This requires the Decision Maker (DM) to consider variables such as the remaining time up to the end of the service contract, the availability of spares, the costs related to the repair actions, etc.

Despite the relevance of PFM for the profitability of the maintenance service contracts, to the authors' best knowledge systemic approaches to address it are still lacking. Indeed, the literature on service is very vast (13; 15), but it covers issues different from that of optimizing the part flow. For example, methods for setting the optimal price of service contracts are proposed in (12; 13), within the game theory framework. The same issue, i.e., contract pricing optimization, is investigated in (17) in combination with the optimization of logistics (i.e., facility locations, capacities and inventories with given service level), and in combination with the issue of optimally scheduling preventive maintenance in (14; 18). In (19) the maintenance, the spares inventory, and the repair capacity are optimized together under the performance contracting framework. Other optimization objectives are the minimization of the warehouse costs through the reduction of the average number of parts sojourning therein (e.g., (14)), the identification of the optimal times for performing maintenance actions and ordering parts (e.g., (20; 21)), the level of repair ((22)), etc.

The focus of this paper is on the search of the best PFM strategy that minimizes the service contract costs over a finite time horizon. Currently, the management of the part flow is dealt with experience-based rules, such as the Most Residual Cycles (MRC) one: the removed parts are always repaired till the end of the maintenance service contract and, at each MS, the parts with the largest residual life among those available at the warehouse are installed on the GT; a new part is purchased only when the warehouse is empty. This simple and intuitive rule guarantees the smallest repair cost at the smallest probability of failure; nonetheless, it is a greedy policy, which may not yield the best PFM strategy on a finite time horizon.

This work, together with (23), introduces the PFM problem, formalizes it as a SDP and proposes the use of Reinforcement Learning (RL, (16; 24; 25)) for its solution. RL is a machine learning technique suitable for addressing SDPs (24), widely applied to decision-making problems in diverse industrial sectors, such as the electricity market (26; 27), military trucks (28), process industry (29), supply chain, maintenance and inventory management (20; 30; 31; 32; 33), to cite a few.

The problem formulation and solution framework proposed in this paper is applied to a scaled-down case study. Differently from (23), the aleatory uncertainty in the part failure behavior is not considered, for ease of conceptualization and illustration of the fundamental concepts and algorithms introduced. This allows:

- proving that the current policy is not optimal even in a simple case with no uncertainty: in the case study, it turns out that the solution given by the MRC rule is not optimal, being outperformed by the policy found by RL
- performing a sensitivity analysis of the profitability of the contract with respect to the decision variables. The analysis of the policies found by MRC and RL shows the extreme impact of the decision variables on the profitability of the contract in this simple deterministic application, which is expected to increase in more realistic cases.

To sum up, the main contribution of this paper is the formulation of the PFM problem optimization and the demonstration of the limitations of the experience-based approach currently used. Even with improvements to the current rules the experience-based approach is shown to be always outperformed by the optimal policy found by RL, although other optimization algorithms (e.g., evolutionary, linear programming, etc...) can be used. Notice, however, that RL is the only approach that can be extended to PFM realistic applications, in which the aleatory uncertainty in the part failure behavior is considered together with many other GT operational characteristics and maintenance rules. This capability of RL is due to the fact that it learns from the scenarios simulated and evaluated, whereby the effort in modeling the optimization problem strongly simplifies. Given the relevance of PFM for maintenance service contract management and the complexity of the problem, which dramatically increases as more variables are considered, our work is expected to give rise to a dedicated line of research on the PFM, which is indeed an important driver in the maintenance policy of the GTs.

The structure of the paper is as follows: in Section 2, we introduce the mathematical formulation of the problem. In Section 3, details about the RL algorithm are briefly provided. In Section 4, a case study is introduced to compare the performance of the RL algorithm to PFM and that of MRC. Finally, conclusions are drawn in Section 5.

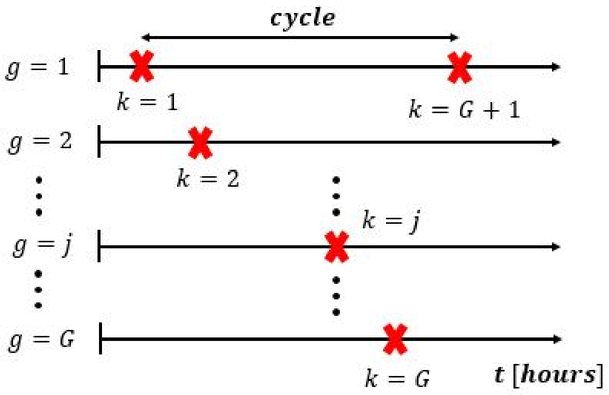

Figure 1: Times of the first MSs of the G gas turbines. The cross-markers denote the MSs. The turbines are indicated by the order of occurrence of their first MS. The MSs occur at fixed times.

2 Problem setting

As mentioned before, the setting considered in this work is derived from a real application of the Oil & Gas industry. However, some simplifying assumptions are made with respect to the real case, which do not affect its main features and the modeling issues. Rather, these assumptions allow disregarding cumbersome technicalities, which is beneficial for the clarity of illustration.

Consider a number G of GTs operated in an Oil & Gas plant. For simplicity of illustration, we consider tracking the flow of a single capital part throughout the maintenance management process without loss of generality of the proposed framework, because the flows of the different capital parts can be assumed independent on each other: although they share the same workshops, this has no practical effect on the mutual dependence of the part flows.

GTs in the plant are periodically maintained, with MSs staggering so that two MSs are never performed simultaneously. In coherence with the real industrial application inspiring this work, we assume that the uncertainty on scheduling time is negligible, as the time windows to perform maintenance are defined by stringent contract rules. The repair time is assumed negligible with respect to the time between two MSs, which implies that the part removed and, then, repaired at any MS is always available at the next MS. The Remaining Residual Useful Life (RUL) of the part is the number r of working cycles remaining. The RUL values range between $r = 0$, when the part must be scrapped, and $r = R$, when the part is new.

The GTs operate for the whole contract duration. In the real industrial practice, there are several ways to define the contract end date such as calendar time, total factored fired hours, achievement of a given number of major maintenance tasks or combinations of these. For ease of illustration, we assume that the contract duration is defined by the total number of scheduled MSs for every

GT, denoted by Z: then, a total number $T = Z \cdot G$ of MSs is performed during the maintenance service contract duration.

The MSs sequence recycles after the maintenance of the last GT. Under these assumptions, the GT undergoing maintenance at the k-th MS, $k \in \{1, ..., T\}$, is univocally identified and indicated by the order of occurrence of its first MS (Figure 1).

At the k-th MS the following decisions must be made:

- For the part removed from the g-th GT, decide whether to repair it or scrap it. $C^{rep}(r)$ is the cost of repairing a part with $r \in \{1, ..., R\}$ remaining cycles, whereas C^{scrap} is the cost of scrapping a part, whatever its RUL is.
- For replacing the removed part, decide whether to buy a new part or select one from those currently available at the warehouse, if any. C^{pur} is the cost of purchasing a new part, whereas the cost of selecting a part from the warehouse is zero, as repair costs have been accounted for at the end of its last working cycle.

We introduce the integer variables d_k and $w_{r,k}$ to indicate the RUL of the capital part removed from the GT maintained at the k-th MS and the number of parts with RUL equal to r available at the warehouse for the k-th MS, respectively, where $r \in \{1, ..., R\}$, $w \in \{0, ..., W\}$ and W is the maximum number of parts that can be stored in the warehouse for each RUL value. Obviously, this limitation is introduced to give due account to the typical constraints on space availability in warehouses and, at the same time, reduce the state space cardinality.

We also define the boolean variable $a_{k,\rho} \in \{0, 1\}$ to indicate whether action $\rho \in \{0, ..., 2R + 1\}$ is taken at the k-th MS. Specifically:

- $a_{k,0} = 1$ when a new part is purchased and installed, whereas the removed part is scrapped.
- $a_{k,\rho} = 1$, $\rho \in \{1, ..., R\}$, when a part with $RUL = \rho$ is installed and the removed part is scrapped.
- $a_{k,R+1} = 1$ when a new part is purchased and installed, and the removed part is repaired.
- $a_{k,\rho} = 1$, $\rho \in \{R + 2, ..., 2R + 1\}$, when a part with $RUL = \rho - R - 1$ is installed and the removed part is repaired.

The boolean variables $a_{k,\rho}$ are such that only one action can be taken at each MS:

$$\sum_{\rho=0}^{2R+1} a_{k,\rho} = 1 \tag{1}$$

The cost incurred at the k-th MS, then, reads:

$$C_k = (a_{k,0} + a_{k,R+1}) \cdot C^{pur} + \sum_{\rho=0}^{R} a_{k,\rho} \cdot C^{Scrap} + \sum_{\rho=R+1}^{2R+1} a_{k,\rho} \cdot C^{Rep}(d_k) \tag{2}$$

The total cost incurred over the entire maintenance service contract duration is the objective function to minimize and is given by:

$$V = \sum_{k=1}^{T} (C_k) \tag{3}$$

Finally, the following equations hold for $\kappa \in \{1, ..., T-G\}$ and $k \in \{1, ..., T\}$, respectively:

$$d_{\kappa+G} = (a_{\kappa,0} + a_{\kappa,R+1}) \cdot R + \sum_{\rho=1}^{R} (a_{\kappa,\rho} \cdot \rho) + \sum_{\rho=R+2}^{2R+1} (a_{\kappa,\rho} \cdot (\rho - R - 1)) - 1 \tag{4}$$

$$w_{r,k+1} = w_{r,k} - (a_{k,r} + a_{k,r+R+1}) + z_{k,r} \tag{5}$$

where:

$$z_{k,r} = \begin{cases} 1, & \text{if} \quad r = d_k \wedge \sum_{\rho=R+1}^{2R+1} a_{k,\rho} = 1 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

Specifically, Eq. (4) is the updating rule of the part RULs between two consecutive MSs of the same GT and Eq. (5) is the updating rule of the number of parts available at the warehouse for any pair of consecutive MSs.

3 Algorithm

In this Section, we give some details about the model-free RL algorithm here developed for part flow optimization. Generally speaking, RL is based on the idea that the DM, who is usually referred to as agent, learns from his/her interactions with the environment to achieve prefixed goals, without knowledge on the updating dynamics of the environment and the specific effect of his/her actions. Thus, we only need to define the state of the environment, the actions available at each state and the corresponding rewards (16).

The state at the k-th MS is defined by the vector $\mathbf{S}_k \in \mathbb{N}^{R+1}$, $k \in \{1, ..., T\}$, whose j-th element is:

$$S_{k,j} = \begin{cases} w_{j,k} & \text{if} \quad j \in \{1, ..., R\} \\ k & \text{if} \quad j = R+1 \end{cases} \tag{7}$$

In words, the first R entries of the state vector at the k-th MS define the number of parts with the different RUL values available at the warehouse, whereas the last entry updates the number of MSs performed (34). Then, the total number of possible states is $T \cdot (W+1)^R$.

Notice that the state vector $\mathbf{S}_k$ does not encode any information about the parts currently installed on the GTs. This leads the SDP to not fully satisfy the Markov property (16), (35), which requires that the knowledge of the current state of the environment be sufficient to predict its future evolution. To see that this property is here infringed, we can notice from Eq. (5) that the state reached by taking any action is completely defined only if we know d_k. Since this variable is not encoded in the state vector, we observe that we have transitions towards different states even if we take the same action on the environment in a given state. As pointed out in (16), the loss of the Markov property typically affects the RL capability of fast convergence to the optimal solution, although RL is eventually able to find it.

The choice of not including the RUL values of the parts installed on the GTs into the state vector has a twofold justification. On one side, including them would broaden the vector state size, which would become $T \cdot (W+1)^R \cdot R^G$: this leads to heavy computational burdens, undermining the applicability of the proposed framework. If for example, we consider that in a real industrial application $R = 6$ and $G = 10$, then the proposed definition of state would reduce the state vector size by 6^{10}. On the other side, we observe from Eq. (4) that the environment state after G MSs is known for any sequence of G actions. Then, the process describing the evolution of the state is a G-order Markov process, in the sense that the knowledge of the sequence of states at the last G events is sufficient to predict its future evolution. Thus, the information about the RUL of the parts installed on the GTs becomes redundant after G steps.

The action taken at the k-th MS is indicated as:

$$A_k = \sum_{\rho=0}^{2R+1} (a_{k,\rho} \cdot \rho) \tag{8}$$

The base reward at the k-th MS is the opposite of the maintenance cost, $-C_k$, as RL is usually framed as a maximization task, whereby minimizing cost is equivalent to maximizing its opposite. In the RL framework, each state-action pair is described by $Q_\pi(\mathbf{S}_k, A_k)$, which measures the expected return starting from state $\mathbf{S}_k$, taking action A_k and thereafter following policy π (16):

$$Q_\pi(\mathbf{S}_k, A_k) = \mathbb{E}_\pi[\sum_{t=k}^{T} (\gamma^{t-k} \cdot (-C_t))|\mathbf{S}_k, A_k] \tag{9}$$

where $k \in \{1, ..., T\}$ and $\gamma \in [0,1]$ is the discount factor. Being the time horizon finite, we set

$\gamma = 1$.

In this work, we use the SARSA(λ) algorithm to find the best approximation of the values of $Q_\pi(\mathbf{S}_k, A_k)$, $k = 1, ..., T$, which simulates a large number of state-action episodes while guaranteeing a faster convergence (e.g., (16), (36)). The SARSA(λ) algorithm relies on the following updating formula at every MS, k:

$$Q(\mathbf{S}_z, A_z) \longleftarrow Q(\mathbf{S}_z, A_z) + (\gamma\lambda)^{(k-z)}\alpha_n \cdot [-C_k + \gamma Q(\mathbf{S}_{k+1}, A_{k+1}) - Q(\mathbf{S}_k, A_k)] \forall z \in \{1, ..., k\} \quad (10)$$

where $\lambda \in [0, 1]$ is the parameter governing the eligibility trace and $\alpha_n \in [0, 1]$ is the learning rate at the n-th episode (see Appendix for further mathematical details).

The choice of using SARSA(λ) among the available RL algorithms (e.g., (25)) is justified by the fact that within the family of value-based RL algorithms, SARSA(λ) has been shown to be a very effective on-policy method ((25)). This makes it simpler to extend it to the eligibility trace paradigm, which guarantees fast and robust convergence, especially in case of finite time horizon SDPs ((16), (36)). On the contrary, off-policy RL algorithms such as Q(λ) do not allow updates that use all the rewards up to the end of the finite horizon due to the presence of explorative actions.

On the other hand, the proposed RL solution suffers from some limitations that can still prevent its full application to the industrial practice in the current form: in complex problems, the state-space becomes very large, whereby the tabular representation of the state-action value function is not practicable. For this, action-value approximation techniques can be used, instead of the tabular approach hereby presented. This allows generalizing the state description, e.g., by removing the contraints on the maximum number of parts available in the warehouse for each RUL level or considering real-valued RUL estimations.

4 Case study

In this Section, we consider a case study derived from an industrial application. The main characteristics are summarized in Table 1. In the considered Oil & Gas plant there are $G = 2$ GTs (first column in Table 1), each one maintained for $Z = 10$ cycles (second column). The maximum component RUL, R, and the maximum number of available parts in the warehouse for each RUL value, W, are both set equal to 3 (third and forth columns in Table 1, respectively). The cost values are shown in the last four columns of Table 1. These values are given in arbitrary units and for illustration purposes, only.

The total number of possible states is $T \cdot (W + 1)^R = 1280$ and the total number of state-action pairs is $T \cdot (W+1)^R \cdot (2R+2) = 10240$.

Table 1: Initial scenario and parameters

G	Z	W	R	C^{Scrap}	$C^{rep}(r=1)$	$C^{rep}(r=2)$	C^{pur}
2	10	3	3	0	50	90	100

The application of the MRC rule to the considered case study is summarized in Table 2. Namely, the first column reports the MS counter, k, followed by three columns representing the situation of the warehouse at the corresponding MS. For example, at the beginning of the considered time horizon, i.e., at k=1, there are three parts with one remaining cycle, $w_{1,1} = 3$, one part with two remaining cycles, $w_{2,1} = 1$, and no new parts, $w_{3,1} = 0$.

Table 2: MRC policy

k	$w_{1,k}$	$w_{2,k}$	$w_{3,k}$	RUL@GT$g=1$	RUL@GT$g=2$	RUL Installed Part	Repair	Purchase	C_k
1	3	1	0	**2**	1	2	Y	N	50
2	3	1	0	2	**0**	2	N	N	0
3	3	0	0	**1**	2	1	Y	N	90
4	3	0	0	1	**1**	1	Y	N	90
5	3	0	0	**0**	1	1	N	N	0
6	2	0	0	1	**0**	1	N	N	0
7	1	0	0	**0**	1	1	N	N	0
8	0	0	0	1	**0**	3	N	Y	100
9	0	0	0	**0**	3	3	N	Y	100
10	0	0	0	3	**2**	3	Y	Y	150
11	0	1	0	**2**	3	2	Y	N	50
12	0	1	0	2	**2**	2	Y	N	50
13	0	1	0	**1**	2	2	Y	N	90
14	1	0	0	2	**1**	1	Y	N	90
15	1	0	0	**1**	1	1	Y	N	90
16	1	0	0	1	**0**	1	N	N	0
17	0	0	0	**0**	1	3	N	Y	100
18	0	0	0	3	**0**	3	N	Y	100
19	0	0	0	**2**	3	3	Y	Y	150
20	0	1	0	3	**2**	2	Y (N)	N	50 (0)
-	0	1(0)	0	**2**	2	-	-	TOT	1350 (1300)

The RUL values of the parts installed on GTs $g = 1$ and $g = 2$ are reported in the fifth and sixth columns, respectively, where the maintained GT is indicated in bold. For example, the part on the GT undergoing maintenance at $k = 1$, $g = 1$, has $d_1 = 2$ remaining cycles, whereas the GT $g = 2$ has been equipped with a part with one remaining cycle at the last MS.

The next three columns detail the action taken at the k-th MS. For example, at the first MS, the RUL of the part installed on GT $g = 1$ is $r = 2$ (seventh column) and the action taken is a repair (eighth column), with no purchase of new parts (nineth column), i.e., $A_1 = 6$. Finally, the last column reports the maintenance cost, C_k, at the k-th MS, $k = 1, ..., T$.

To go further into the updating dynamics of Table 2, we can see that at the second MS, $w_{2,2} = 1$ because the part removed from GT $g = 1$ is now available at the warehouse for installation on GT $g = 2$. The removed part must be scrapped, as it has no remaining cycles, $d_2 = 0$. This gives a maintenance cost $C_2 = 0$. Notice that the RUL of GT $g = 1$ is not modified at MS $k = 2$, as RUL values are updated at the end of the maintenance cycles, only.

Table 3: RL policy

k	$w_{1,k}$	$w_{2,k}$	$w_{3,k}$	RUL@GT$g = 1$	RUL@GT$g = 2$	RUL installed part	Repair	Purchase	C_k
1	3	1	0	**2**	1	1	Y	N	50
2	2	2	0	1	**0**	2	N	N	0
3	2	1	0	**0**	2	3	N	Y	100
4	2	1	0	3	**1**	3	N	Y	100
5	2	1	0	**2**	3	3	Y	Y	150
6	2	2	0	3	**2**	2	Y	N	50
7	2	2	0	**2**	2	3	Y	Y	150
8	2	3	0	3	**1**	3	Y	Y	190
9	3	3	0	**2**	3	2	Y	N	50
10	3	3	0	2	**2**	2	Y	N	50
11	3	3	0	**1**	2	2	N	N	0
12	3	2	0	2	**1**	1	N	N	0
13	2	2	0	**1**	1	1	N	N	0
14	1	2	0	1	**0**	1	N	N	0
15	0	2	0	**0**	1	2	N	N	0
16	0	1	0	2	**0**	3	N	Y	100
17	0	1	0	**1**	3	3	N	Y	100
18	0	1	0	3	**2**	2	Y	N	50
19	0	1	0	**2**	2	2	Y	N	50
20	0	1	0	2	**1**	2	N	N	0
-	0	0	0	**1**	2	-	-	TOT	1190

Finally, notice that the GT parts are purchased when the warehouse is empty, only. For example, at MS $k = 8$, a part is purchased and installed on GT $g = 2$ with $r = 3$.

The part flow solution given by the application of the MRC rule yields a total maintenance cost of 1350 (in arbitrary units), as reported in the last row of Table 2. This result can be improved by combining MRC with an engineering good-sense rule: eliminate the last repair action at MS $k = 20$, as it is useless to the continuation of the GT operation. This yield a total cost $V = 1300$ (in arbitrary units), in correspondence of an empty warehouse (this solution is indicated in brackets on the last two rows in Table 2).

The part flow solution provided by MRC is compared to that provided by the SARSA(λ) algorithm, whose setting parameters are reported in Appendix. According to (25), these have been defined based on a series of experiments (see Appendix 1), from which it emerged that λ is the most impacting on convergence. In fact, Figure 2 shows the behavior of the state-action pair value for the first state (i.e., $\mathbf{S_1}=$ (3,1,0,1)) and the corresponding action taken at the beginning of the episode, for three different values of λ. From this Figure, we can see that large values of λ yield fast convergence, whereas setting $\lambda = 0$ leads to not converging within 10^5 episodes. Notice that the final oscillating behaviors of these curves are due to the exploration tasks of RL.

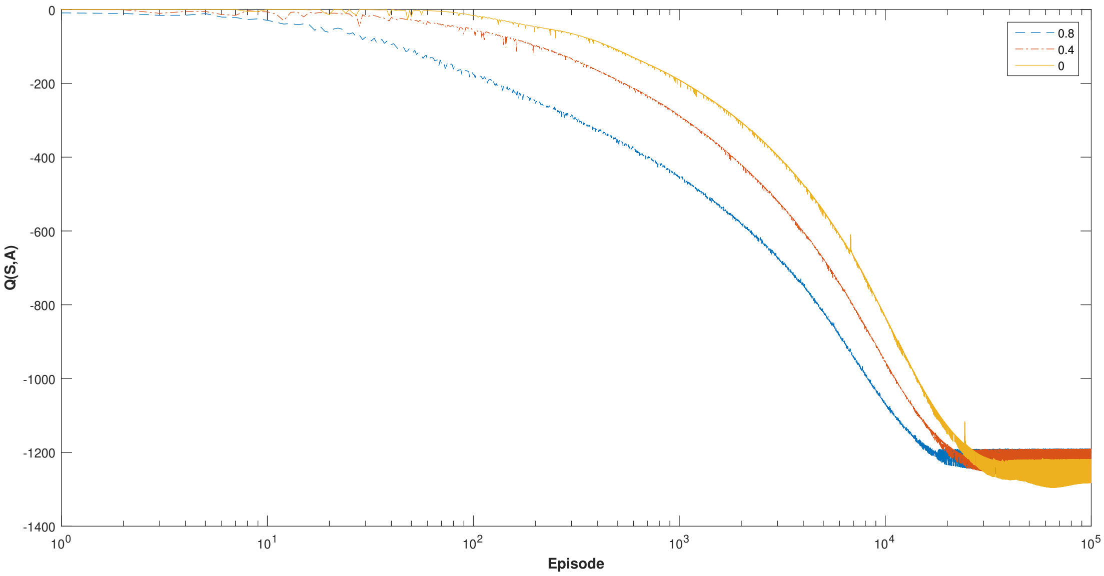

Figure 2: Convergence of $Q(S, A)$ for different values of λ

The RL algorithm converges after around 10^4 episodes (Figure 2), in almost 95 seconds on a 2.20GHz CPU, 4GB RAM computer. The optimal policy found is summarized in Table 3, through the same reading scheme of Table 2 and yields a final cost of 1190, in arbitrary units. This is smaller than that of the MRC policy, i.e., 1300.

Table 4: Comparison between MRC and RL policies reported in Tables 2 and 3

	Number of Purchasing	Repairs of Parts with r=2	Repairs of Parts with r=1	Scrap of Parts with r>0	Scrap of Parts with r=0
RL	7	8	1	6	5
MRC	6	5	5	1	9

The comparison of Tables 2 and 3 is summarized in Table 4. From this, we can note that MRC requires a number of purchase actions and repairs of parts with $r = 2$ smaller than the corresponding ones of the optimal policy (i.e., columns 2 and 3 of Table 4). This advantage is counter-balanced by the larger number of repair actions of parts with one remaining cycle (5 for MRC vs 1 for RL), whose cost is almost equal to that of purchasing new parts (see Table 1). From this, one can be tempted to conclude that the superior result of new parts RL is due to the scrapping of parts with RUL $r > 0$ (see column 5 in Table 4), which avoids repairing parts with $r = 1$. However, this conclusion is not right. In fact, on the one hand we can notice that the RL optimal policy may require to perform repair actions on parts with $r = 1$. For example, at MS $k = 8$, the part coming from GT $g = 2$ with $r = 1$ is repaired (see Table 3). On the other hand, if we modify the MRC policy with the constraint of always scrapping the parts with $r = 1$, then the total maintenance expenditures are equal to 1250, in arbitrary units (see Table 5), which still is larger than 1190. This proves that scrapping parts with $r = 1$ is not always convenient.
Moreover, the policy of consuming all parts until their RUL $r = 0$ certainly compromises the possibility of reaching the optimal part flow solution: if we apply the RL algorithm in the setting in which $d_k > 0$ and $a_{k,\rho} = 0$, $\rho \in \{0, ..., 3\}$, for $k = 1, ..., T$, then the final maintenance cost is 1290, in arbitrary units (see Table 6), which is smaller than that of the MRC policy (see Table 2), although it is larger than that of RL (see Table 3). From these considerations, it emerges that a simple rule cannot be found to optimally manage the part flow and an optimization algorithm is required to find the most convenient sequence of actions over the service contract duration.

The optimal policy is expected to be dependent on model parameters such as the initial conditions of the warehouse and the length of the time horizon. The initial warehouse composition depends on two factors: number of parts and total number of available cycles. To fairly capture the effect on the total costs of the number of parts initially available, we apply RL and MRC to different compositions of the warehouse, which are such that the sum of $RULs$ available is equal to that in Tables 2 and 3: $\sum_{r=1}^{3} w_{r,1} \cdot r = 5$. Table 7 shows that the final costs are strongly affected by the initial conditions of the warehouse: the larger the number of parts in the warehouse the smaller the costs, which range from 1190, when there are 4 parts in the warehouse, to 1300, when there are 2. The RL policy always outperforms that of MRC. Notice also that rows 2 and 3 refer to

Table 5: MRC policy, scrapping allowed when $RUL = 1$

k	$w_{1,k}$	$w_{2,k}$	$w_{3,k}$	RUL@GT$g = 1$	RUL@GT$g = 2$	RUL installed part	Repair	Purchase	C_k
1	3	1	0	**2**	1	2	Y	N	50
2	3	1	0	2	**0**	2	N	N	0
3	3	0	0	**1**	2	1	N	N	0
4	2	0	0	1	**1**	1	N	N	0
5	1	0	0	**0**	1	1	N	N	0
6	0	0	0	1	**0**	3	N	Y	100
7	0	0	0	**0**	3	3	N	Y	100
8	0	0	0	3	**2**	3	Y	Y	150
9	0	1	0	**2**	3	2	Y	N	50
10	0	1	0	2	**2**	2	Y	N	50
11	0	1	0	**1**	2	2	N	N	0
12	0	0	0	2	**1**	3	N	Y	100
13	0	0	0	**1**	3	3	N	Y	100
14	0	0	0	3	**2**	3	Y	Y	150
15	0	1	0	**2**	3	2	Y	N	50
16	0	1	0	2	**2**	2	Y	N	50
17	0	1	0	**1**	2	2	N	N	0
18	0	0	0	2	**1**	3	N	Y	100
19	0	0	0	**1**	3	3	N	Y	100
20	0	0	0	3	**2**	3	N	Y	100
-	0	0	0	**2**	3	-	-	TOT	1250

settings in which the number of parts initially available is the same (i.e., 3), with the same value of cumulated RUL. Nonetheless, the maintenance costs are different. This tells us that the knowledge about both the total number of parts and the total RUL initially available in the warehouse is not sufficient to derive the optimal maintenance costs, which, indeed, depend on the overall starting warehouse composition.

These results are confirmed when we evaluate the dependence of the costs on the total RUL initially available. Table 8 reports four different initial warehouse compositions, the first three referring to a warehouse with three parts with $r = 1, 2$ and 3, respectively. From Table 8, we can see that the larger the sum of the RUL initially available, the smaller the costs incurred, especially when we consider the warehouse with new parts. Row 4, instead, refers to the warehouse initially composed of four parts with RUL $r = 1$, i.e., one more than those in the first row. The total cost of this scenario, with total RUL initially available equal to 4, is the same as that of scenario 3, with total RUL equal to 9 and also of the scenario in the first row of Table 3, where there are 4 parts with total RUL equal to 5. This confirms that in the settings here considered, the final costs

Table 6: RL policy, scrapping allowed when $RUL = 0$, only

k	$w_{1,k}$	$w_{2,k}$	$w_{3,k}$	RUL@GT$g=1$	RUL@GT$g=2$	RUL Installed Part	Repair	Purchase	C_k
1	3	1	0	**2**	1	3	Y	Y	150
2	3	2	0	3	**0**	3	N	Y	100
3	3	2	0	**2**	3	1	Y	N	50
4	2	3	0	1	**2**	2	Y	N	50
5	2	3	0	**0**	2	2	N	N	0
6	2	2	0	2	**1**	1	Y	N	90
7	2	2	0	**1**	1	3	Y	Y	190
8	3	2	0	3	**0**	2	N	N	0
9	3	1	0	**2**	2	1	Y	N	50
10	2	2	0	1	**1**	1	Y	N	90
11	2	2	0	**0**	1	3	N	Y	100
12	2	2	0	3	**0**	1	N	N	0
13	1	2	0	**2**	1	2	Y	N	50
14	1	2	0	2	**0**	1	N	N	0
15	0	2	0	**1**	1	2	Y	N	90
16	1	1	0	2	**0**	2	N	N	0
17	1	0	0	**1**	2	1	Y	N	90
18	1	0	0	1	**1**	1	Y	N	90
19	1	0	0	**0**	1	1	N	N	0
20	0	0	0	1	**0**	3	N	Y	100
-	0	0	0	**0**	3	-	-	TOT	1290

are more sensitive to the number of parts than to the total RUL.

Finally, Figure 3 shows how the length of the time horizon affects the difference between the maintenance costs of RL and MRC with respect to the four scenarios reported in Table 7. The smallest differences (i.e., 40, in arbitrary units) are achieved when the total number of MSs is such that the parts installed on the GTs following the MRC policy have small RUL at the end of the time horizon, which implies that no parts are left in the warehouse. Indeed, though the MRC policy is quite good for specific values of the contract length, it is always dominated by the RL policy.

Table 7: Effect of the number of parts on costs

Scenario	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	Maintenance Expenditures MRC	Maintenance Expenditures RL	Delta	Number of Parts
1	3	1	0	1300	1190	110	4
2	1	2	0	1390	1280	110	3
3	2	0	1	1350	1240	110	3
4	0	1	1	1440	1300	140	2

Table 8: Effect of the total RUL on costs

Scenario	$w_{1,1}$	$w_{2,1}$	$w_{3,1}$	Maintenance Expenditures	$\sum RUL$
1	3	0	0	1300	3
2	0	3	0	1280	6
3	0	0	3	1190	9
4	4	0	0	1190	4

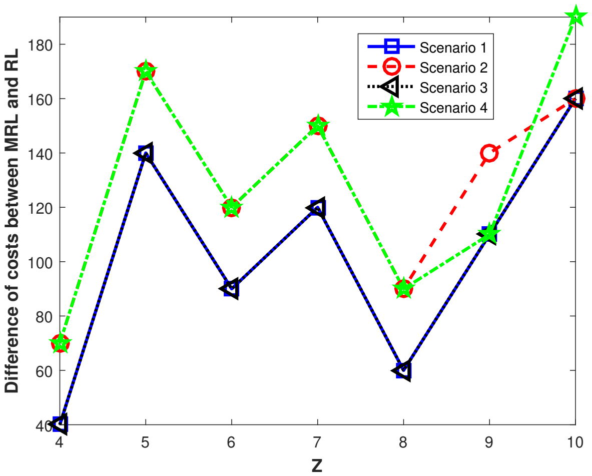

Figure 3: Difference of maintenance expenditures between MRC and RL for 4 different scenarios

5 Conclusions

This work offers a formalization of the GT PFM in the Oil & Gas industry. The problem is of crucial importance for the profitability and the reliability of the GT plants. GT PFM has been formulated as a SDP and RL has been used for the solution.

Other optimization algorithms such as evolutionary algorithms (e.g., Genetic Algorithms, Differential Evolution, etc...), dynamic programming or linear programming algorithms, could be adopted for the specific setting considered in this work. However, the choice of model-free RL has a twofold justification. On one side, RL algorithms allow encoding the aleatory uncertainties, e.g., in the failure times of the GT parts or in the non-negligible duration of the inspection cycle, more easily than the other algorithms. On the other side, although here not considered, the complexity of the real industrial applications requires SDP to encode many additional GT operational aspects, such as the possibility of inspecting the parts without performing maintenance (i.e., condition-based maintenance), the different duration of the maintenance intervals for parts of different technologies, the constraints on the shareability of the parts on GTs with different operation temperatures, etc. Accounting for these GT operation features requires encoding constraints about the actions that can be taken in each state, which are really difficult to set in both evolutionary and linear programming frameworks. Another characteristic of real applications is the non-negligible duration of the maintenance interventions for parts of different technologies, which invalidates the assumption that the parts removed from the GTs are readily available for the next MS. These features of real practice lead to the fact that Eqs. (4) and (5) can no longer be used for updating the MS dynamics, and more complex equations should be found for the specific application. On the contrary, RL, being a model-free method which does not require the knowledge of the updating dynamics, allows easily encoding the additional features of the specific real applications, as it acts on the simulation of the decision process and, thus, selects actions from those feasible, only. This makes RL easily integrable with part flow simulators, as long as action-value approximations techniques are used to handle the large dimensionality of the action-state space. This issue will be tackled in future works.

The results of a case study inspired by a real industrial application show that RL finds PFM policies with maintenance costs smaller than those derived from experience-based rules (i.e, MRC). Moreover, the case study shows that the optimal maintenance policy found by RL strongly depends on the initial situation of the warehouse and the length of service contract, which makes not possible to identify a set of general rules.

References

[1] Yang K, He C, Huang Q, Huang ZY, Wang C, Wang Q, et al. Very high cycle fatigue behaviors of a turbine engine blade alloy at various stress ratios. International Journal of Fatigue. 2016;.

[2] Peters J, Ritchie R. Influence of foreign-object damage on crack initiation and early crack growth during high-cycle fatigue of Ti–6Al–4V. Engineering Fracture Mechanics. 2000;67(3):193–207.

[3] Boyce B, Ritchie R. Effect of load ratio and maximum stress intensity on the fatigue threshold in Ti–6Al–4V. Engineering Fracture Mechanics. 2001;68(2):129–147.

[4] Bodrov A, Stalder J. An analysis of axial compressor fouling and a blade cleaning method. 1998;.

[5] Kurz R, Brun K. Fouling mechanisms in axial compressors. Journal of Engineering for Gas Turbines and Power. 2012;134(3):032401.

[6] Morini M, Pinelli M, Spina P, Venturini M. Influence of blade deterioration on compressor and turbine performance. Journal of engineering for gas turbines and power. 2010;132(3):032401.

[7] Eliaz N, Shemesh G, Latanision R. Hot corrosion in gas turbine components. Engineering failure analysis. 2002;9(1):31–43.

[8] Goward G. Progress in coatings for gas turbine airfoils. Surface and Coatings Technology. 1998;108:73–79.

[9] Compare M, Martini F, Mattafirri S, Carlevaro F, Zio E. Semi-Markov model for the oxidation degradation mechanism in gas turbine nozzles. IEEE Transactions on Reliability. 2016;65(2):574–581.

[10] Balevic D, Hartman S, Youmans R. Heavy-duty gas turbine operating and maintenance considerations. GE Energy, Atlanta, GA. 2010;.

[11] Ng I, Parry G, Wild P, McFarlane D, Tasker P. Complex Engineering Service Systems: Concepts and Research. Springer, London; 2011.

[12] Wang W. A model for maintenance service contract design, negotiation and optimization. European Journal of Operational Research. 2010;201(1):239–246.

[13] Murthy D, Asgharizadeh E. Optimal decision making in a maintenance service operation. European Journal of Operational Research. 1999;116(2):259–273.

[14] Bollapragada S, Gupta A, Lawsirirat C. Managing a portfolio of long term service agreements. European journal of operational research. 2007;182(3):1399–1411.

[15] Hu Q, Boylan JE, Chen H, Labib A. OR in spare parts management: a review. European Journal of Operational Research. 2017;.

[16] Sutton RS, Barto AG. Introduction to reinforcement learning. vol. 135. MIT Press Cambridge; 1998.

[17] Lieckens KT, Colen PJ, Lambrecht MR. Network and contract optimization for maintenance services with remanufacturing. Computers & Operations Research. 2015;54:232–244.

[18] Kurz J. Capacity planning for a maintenance service provider with advanced information. European Journal of Operational Research. 2016;251(2):466–477.

[19] Jin T, Tian Z, Xie M. A game-theoretical approach for optimizing maintenance, spares and service capacity in performance contracting. International Journal of Production Economics. 2015;161:31–43.

[20] Keizer MCO, Teunter RH, Veldman J. Joint condition-based maintenance and inventory optimization for systems with multiple components. European Journal of Operational Research. 2017;257(1):209–222.

[21] Van Horenbeek A, Buré J, Cattrysse D, Pintelon L, Vansteenwegen P. Joint maintenance and inventory optimization systems: A review. International Journal of Production Economics. 2013;143(2):499–508.

[22] Jaturonnatee J, Murthy D, Boondiskulchok R. Optimal preventive maintenance of leased equipment with corrective minimal repairs. European Journal of Operational Research. 2006;174(1):201–215.

[23] Compare M, Bellani L, Cobelli E, Zio E. Reinforcement learning-based flow management of gas turbine parts under stochastic failures. The International Journal of Advanced Manufacturing Technology. 2018;99(9-12):2981–2992.

[24] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: A survey. Journal of artificial intelligence research. 1996;4:237–285.

[25] Szepesvári C. Algorithms for reinforcement learning. Synthesis lectures on artificial intelligence and machine learning. 2010;4(1):1–103.

[26] Kuznetsova E, Li YF, Ruiz C, Zio E, Ault G, Bell K. Reinforcement learning for microgrid energy management. Energy. 2013;59:133–146.

[27] Rahimiyan M, Mashhadi HR. An adaptive Q-learning algorithm developed for agent-based computational modeling of electricity market. IEEE Transactions on Systems, Man, and Cybernetics Part C, Applications and Reviews. 2010;40(5):547.

[28] Barde SR, Yacout S, Shin H. Optimal preventive maintenance policy based on reinforcement learning of a fleet of military trucks. Journal of Intelligent Manufacturing. 2016;p. 1–15.

[29] Aissani N, Beldjilali B, Trentesaux D. Dynamic scheduling of maintenance tasks in the petroleum industry: A reinforcement approach. Engineering Applications of Artificial Intelligence. 2009;22(7):1089–1103.

[30] Pontrandolfo P, Gosavi A, Okogbaa OG, Das TK. Global supply chain management: a reinforcement learning approach. International Journal of Production Research. 2002;40(6):1299–1317.

[31] Giannoccaro I, Pontrandolfo P. Inventory management in supply chains: a reinforcement learning approach. International Journal of Production Economics. 2002;78(2):153–161.

[32] Kim C, Jun J, Baek J, Smith R, Kim Y. Adaptive inventory control models for supply chain management. The International Journal of Advanced Manufacturing Technology. 2005;26(9-10):1184–1192.

[33] Liu Q, Dong M, Lv W, Ye C. Manufacturing system maintenance based on dynamic programming model with prognostics information. Journal of Intelligent Manufacturing. 2017;p. 1–19.

[34] Li Z, Ding Z, Wang M. Optimal Bidding and Operation of a Power Plant with Solvent-Based Carbon Capture under a CO 2 Allowance Market: A Solution with a Reinforcement Learning-Based Sarsa Temporal-Difference Algorithm. Engineering. 2017;3(2):257–265.

[35] Whitehead SD, Lin LJ. Reinforcement learning of non-Markov decision processes. Artificial Intelligence. 1995;73(1-2):271–306.

[36] Wang YH, Li THS, Lin CJ. Backward Q-learning: The combination of Sarsa algorithm and Q-learning. Engineering Applications of Artificial Intelligence. 2013;26(9):2184–2193.

[37] Guo M, Liu Y, Malec J. A new Q-learning algorithm based on the metropolis criterion. IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics). 2004;34(5):2140–2143.

[38] Hwang KS, Tan SW, Chen CC. Cooperative strategy based on adaptive Q-learning for robot soccer systems. IEEE Transactions on Fuzzy Systems. 2004;12(4):569–576.

6 APPENDIX

The SARSA(λ) algorithm steps (16) can be summarized as follows:

```
-Initialize all $Q_\pi(\mathbf{S}_k, A_k) = 0, \forall k$
while $n < N$ do
    -$k \longleftarrow 1$
    -Define the initial conditions as $d_1$ and $\mathbf{S}_1$
    -Apply the $\epsilon$-greedy policy to get $A_1$
    while $k < T$ do
        -Execute $A_k$, receive $-C_{k+1}$ and observe $d_{k+1}$ and $\mathbf{S}_{k+1}$
        -Apply the $\epsilon$-greedy policy to get $A_{k+1}$
        -Update all $Q_\pi(\mathbf{S}_z, A_z)$ visited within the $k$-th MS according to Eq. (10)
        -$k \longleftarrow k + 1$
    end
    -Update $\alpha_n$ and $\epsilon_n$
end
- Find optimal policy
```

To find a good compromise between exploration and exploitation, we gradually drop down the exploration (37), i.e. we use:

$$\epsilon_n = \epsilon_0 \cdot (N_{\epsilon_0} + 1)/(N_{\epsilon_0} + n) \tag{11}$$

where $\epsilon_0 \in [0, 1]$ is the initial value, ϵ_n is ϵ at the n-th episode and N_{ϵ_0} is the episode at which the value of ϵ_n is almost halved.

Yet, as explained in (38), the larger the value of α_n, the faster the agent learns and, thus, the larger the probability of converging to a sub-optimal solution. To avoid this drawback α has been set as:

$$\alpha_n = \alpha_0 \cdot (N_{\alpha_0} + 1)/(N_{\alpha_0} + n) \tag{12}$$

where α_0 is the initial learning rate and N_{α_0} and α_n are the analogous of N_{ϵ_0} and ϵ_n.

Finally, Table 9 reports the values of the parameters used in this work.

Table 9: RL Parameters

ϵ_0	α_0	γ	λ	N	N_{α_0}	N_{ϵ_0}
0.1	0.1	1	0.8	1e5	1e4	1e3