# Agent-based modeling and reinforcement learning for optimizing energy systems operation and maintenance: the Pathmind solution

Luca Pinciroli
*Energy Department, Politecnico di Milano, Milan, Italy. E-mail: luca.pinciroli@polimi.it*

Piero Baraldi
*Energy Department, Politecnico di Milano, Milan, Italy. E-mail: piero.baraldi@polimi.it*

Michele Compare
*Aramis S.r.l., Milan, Italy. E-mail: michele.compare@aramis3d.com*

Sahar Esmaeilzadeh
*Pathmind Inc., 1328 Mission Street, San Francisco, CA 94103, USA. E-mail: sahar@pathmind.com*

Mohammed Farhan
*Pathmind Inc., 1328 Mission Street, San Francisco, CA 94103, USA. E-mail: mohammed@pathmind.com*

Brett Göhre
*Pathmind Inc., 1328 Mission Street, San Francisco, CA 94103, USA. E-mail: brett@pathmind.com*

Roberto Grugni
*Engineering Ingegneria Informatica, Milan, Italy E-mail: roberto.grugni@eng.it*

Luigi Manca
*Engineering Ingegneria Informatica, Milan, Italy E-mail: luigi.manca@eng.it*

Enrico Zio
*Energy Department, Politecnico di Milano, Milan, Italy.*
*MINES ParisTech, PSL Research University, CRC, Sophia Antipolis, France.*
*Eminent Scholar, Department of Nuclear Engineering, College of Engineering, Kyung Hee University, Republic of Korea. E-mail: enrico.zio@polimi.it*

The optimization of the Operation and Maintenance (O&M) of energy systems equipped with Prognostics and Health Management (PHM) capabilities can be framed as a sequential decision process, which can be addressed by Reinforcement Learning (RL). However, using RL algorithms requires specific skills, whereas the understanding of the possibly counter-intuitive solutions proposed by RL is not straifhtforward. To sidestep both issues, we use Pathmind, a software tool which enables effectively exploiting the RL capabilities without deep knowledge of machine learning. Pathmind is encoded in the Anylogic environment, which is an Agent-Based simulation software that simplifies the system modeling and allows easily visualizing the effects of the optimized policy. A scaled-down wind farm case study is used to demonstrate the potential of RL in identifying an optimal O&M policy and to show the ease of use of Pathmind and AnyLogic.

*Keywords*: Optimization, Operation and Maintenance, Reinforcement Learning, Pathmind, AnyLogic.

## 1. Introduction

The optimization of Operation and Maintenance (O&M) allows significantly increasing the profit of the plant, reducing its Life Cycle Cost (LCC). As proposed in Pinciroli et al. (2020); Bellani et al. (2019), O&M decision problem can be formalized as a Markov Decision Problem (MDP) over a long-time horizon, whereas Reinforcement Learning (RL) (Sutton and Barto (2018); Arulkumaran et al. (2017)) can be used to find the optimal

Proceedings of the 30th European Safety and Reliability Conference and
the 15th Probabilistic Safety Assessment and Management Conference.
*Edited by* Piero Baraldi, Francesco Di Maio and Enrico Zio
 *Published by* Research Publishing, Singapore
ISBN: 981-973-0000-00-0 :: doi: 10.3850/981-973-0000-00-0_"Pathmind paper"

solution. RL is a ML framework in which a learning agent optimizes its behaviour by means of consecutive trial and error interactions with a white-box model of the system in order to find the optimal policy (Kaelbling et al. (1995); Grondman et al. (2012)), i.e. the function identifying the most suitable action to be taken in each system state in order to maximize a numerical reward.

In principle, tabular dynamic programming algorithms allow finding the exact solution of MDPs (Sutton and Barto (2018)). However, in most cases their computational cost is not compatible with realistic applications to complex systems. Furthermore, the application of RL to industrial systems requires prior knowledge and understanding of RL algorithms technicalities, whereas the the possibly counterintuitive solutions provided by RL can make the understanding of the results very challenging. This lack of confidence in the solution can lead asset managers to distrust the RL solutions.

To overcome these issues, we resort to Pathmind, a web application developed by Pathmind.Inc in AnyLogic environment, which offers a RL state-of-the-art training algorithm and a RL hyperparameters tuning methodology allowing the exploitation of the potentiality of RL while not requiring the user to have a deep knowledge on RL algorithms. On the other hand, AnyLogic offers a modeling and simulation environment, with the possibility of visualizing the effect of the RL policy on the system. This is very helpful in uderstanding the solutions proposed by RL.

In this paper, the optimization of the O&M strategy of a scaled-down wind farm is studied in order to show: $i$) the capabilities of RL to solve the O&M decision problem and $ii$) the benefit of the Pathmind Library in Anylogic to simplify the application of RL to complex optimzation issues.

The structure of the paper is as follows. In Section 2, details about the RL algorithm used in Pathmind are provided. In Section 3, we introduce the case study concerning a wind farm. Results are discussed in Section 4. Finally, conclusions are drawn in Section 5.

## 2. Reinforcement Learning using Pathmind

The general structure of RL is shown in Figure 1. The agent is the decision maker and everything it can interact with becomes the environment. At every decision step, the agent observes the state of the environment and selects an action. Every action an agent takes causes the transition of the environment to a new state, which provides a certain reward. This process repeats trying to maximize the overall reward value (Sutton and Barto (2018)).

When using the PathmindHelper library for AnyLogic, the agent is the neural network trained on the Pathmind web application, whereas the environment is the Anylogic model. The user is required to define five functions in order allow the agent training: $i$) Observation Function, i.e., the function which collects the observations to compose the environment state; $ii$) Reward Variables, i.e., the variables defining the objectives of the training; $iii$) Action Function, i.e., the function that defines the actions that the learning agent can take; $iv$) Action Trigger, i.e., the function that triggers the next action of the learning agent; $v$) Reward Function, i.e., the function that puts together the reward variables in order to provide a total reward to lead the training.

Once these variables are defined, the simulation model can be exported as a standalone Java application from Anylogic and uploaded on the Pathmind web application to perform the training on cloud. On the cloud platform, multiple experiments can be run with different reward functions. These allow exploring the policies that can be obtained with different combinations of the reward variables.

The training algorithm used by Pathmind is the Proximal Policy Optimization (PPO) algorithm which has been shown to provide state-of-the-art performance, despite its ease of implementation and tuning (Schulman et al. (2017)). With respect to the RL hyperparameters optimization, the Population-Based Training (PBT) method pioneered by Google's DeepMind (Jaderberg et al. (2017)), is used by Pathmind. PBT aims at automatically discovering the best set of hyperparameters that allow the learning agent to find the best performing policy as quicly as possible. PBT has been shown to be more time efficient in discovering the best performing policy compared to other techniques (e.g. grid search).

Once the training is concluded, the policy can be imported back into AnyLogic to test its performance.

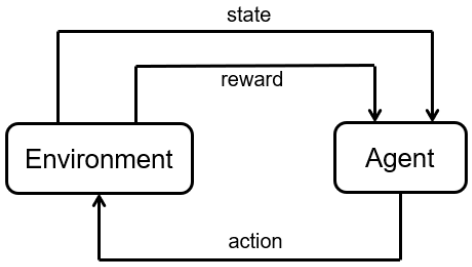


Fig. 1. Schematic representation of reinforcement learning structure.

## 3. Case Study

We consider a wind farm composed of $L = 20$ identical Wind Turbines (WTs) equipped with Prognostics and Health Management (PHM) capabilities, i.e., it is possible to estimate the Remaining Useful Life (RUL) of each component. The failure time, $T$, of each WT is sampled from a exponential distribution with failure rate $\lambda_f = 6.58\ 10^{-3}\frac{1}{d}$ corresponding to the mean value of the failure rates of different WT sub-systems (Ozturk et al. (2018)). The true RUL, $R^*$, at time $t$ is set equal to the $T-t$, whereas the predicted RUL, $R$, is estimated at each time as:

$$R = T - t + \epsilon_R \tag{1}$$

where $\epsilon_R \sim N(0, \sigma_R)$. We assume:

$$\sigma_R = \begin{cases} 50 & \text{if } R^* \geq 300 \\ 30 & \text{if } 100 \leq R^* < 300 \\ 10 & \text{if } 0 < R^* < 100 \end{cases} \tag{2}$$

to take into account the fact the we expect the RUL estimation to be more precise as the current time $t$ approaches the sampled failure time $T$.
The production level, $P$, is assumed be a random variable sampled from different probability density functions according to the true RUL $R^*$, to simulate the dependence of the production level from the degradation level and from the different enviromental conditions affecting each component. In particular, the normalized production level is defined as:

$$P \sim \begin{cases} \mathcal{U}(0,1) & \text{if } R^* \geq 300 \\ \mathcal{U}(0,0.7) & \text{if } 50 \leq R^* < 300 \\ \mathcal{U}(0,0.3) & \text{if } 0 < R^* < 50 \\ 0 \text{ if } R^* = 0 \end{cases} \tag{3}$$

where $\mathcal{U}(a,b)$ identifies a uniform distribution on $[a,b]$. Notice that for each degradation level, the left extreme of the interval $a = 0$. This is to take into account the stochasticity of the wind velocity. We assume to be able to estimate the power production of the next $J = 3$ days. The estimation is affected by uncertainty, described by noise $\epsilon_P \sim N(0, \sigma_P)$ to be added to $P$, with $\sigma_P = 0.05$.
$C = 3$ maintenance crews are available for the wind farm maintenance. Each crew can reach a component and perform either Preventive Maintenance (PM), if the component is not failed, i.e., $R^* > 0$, or corrective maintenance (CM), if the component is failed, i.e., $R^* = 0$.
The maintenance times are sampled from exponential distributions with repair rate $\lambda_{PM} = 0.125\frac{1}{h}$ and $\lambda_{CM} = 0.083\frac{1}{h}$, for PM and CM, respectively, setting $\lambda_{PM}$ $\lambda_{CM}$ equal to the mean value of the repair rates of different WT subsystems (Carroll et al. (2015)).

Each day the $l-th$ turbine provide an income equal to $100 \times P_l$ in arbitrary units. The cost of a PM action is set equal to 600 and the cost of a CM action is set equal to 2500, both in arbitrary units.

### 3.1. *AnyLogic Model*

The wind farm model has been developed in AnyLogic Professional 8.6 software. The model time unit is days and the simulation run time is 1100 days. The model encodes two types of agent: Wind Turbine and Maintenance Crew. The state charts for wind turbines and maintenance crews are shown in Figure 2 and Figure 3, respectively. The wind turbine agent can be in three states: $i$) $Working$, if $R > 0$, $ii$) $Failed$, if $R = 0$ and $iii$) $UnderService$, if the turbine is under maintenance. The maintenance crew agent can be in four states: $i$) $Idle$, if no tasks are assigned and it is waiting at the depot, $ii$) $DrivingtoWork$, when it is reaching the assigned WT to perform maintenance, $iii$) $Working$, when it is performing maintenance. and $iv$) $DrivingHome$ when no tasks are assigned and the agent is driving to the depot.

### 3.2. *Reinforcement Learning Implementation*

We select the wind turbine agent as the learning agent. Since our model is characterized by $L = 20$ identical WTs, we exploit the Multi Agents feature of Pathmind, which allows training multiple agents using a single shared policy. This feature is useful when the policy is unable to distinguish between the individual components that lead to the success of the entire system and when it is

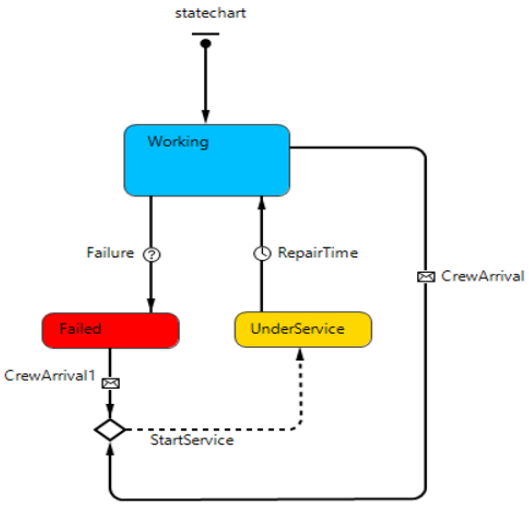


Fig. 2. Wind Turbine state chart.

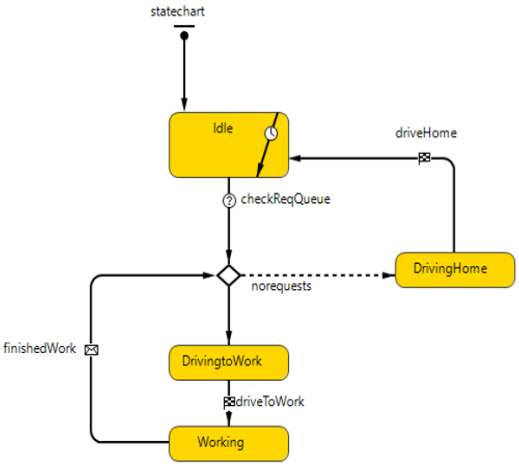


Fig. 3. Maintenance Crew state chart.

difficult for a single-agent policy to separate the performance of each individual component. At each decision instant, i.e. every day, the state of the $l-th$ WT is given as input to the corresponding agent, an action is selected, the reward for the corresponding agent is computed and used to update the RL policy.

### 3.3. *Observation Space*

We define the state $S \in \mathcal{R}^{4+J}$ as a vector containing all the information retriveable from the $l-th$ WT and its environment. In particular, the state contains information on $i$) the current simulation time, $t$, $ii$) the number of WTs asking for maintenance, $iii$) the predicted RUL $R_l$, $iv$) a boolean variable indicating if the $l-th$ WT is working or failed and $v$) the predicted production level, $P_l$, for the following $J$ days.

### 3.4. *Reward Variables*

The following quantities are considered as reward variables for the $l-th$ WT: $i$) the cumulative revenue generated over time $G_l$, $ii$) the cumulative costs over time $X_l$, $iii$) a boolean variable indicating if the $l-th$ WT has failed $f_l$, $iv$) the predicted RUL $R_l$ and $v$) the total distance travelled by the maintenance crews $D$.

### 3.5. *Action Space*

At each decison step, every WT can choose one among two actions: $i$) ask for maintenance or $ii$) wait. If the first action is selected, the corresponding WT is inserted in the list of the WTs requesting maintenance and the first available maintenance crew will be sent to the corresponding location to perform maintenance. If the WT is already in the list the action will be skipped. On the contrary, if the second action is selected, the WT continues working in normal operation.

### 3.6. *Reward Function*

When an action trigger occurs, the learning agents save the values of the reward variables before and after the action is performed. Two values called $before$ and $after$, respectively, are used to build the reward function. The following reward function (RF) is defined to train the agent corresponding to the $l-th$ WT:

$$\begin{aligned} RF &+= after.G_l - before.G_l; \\ RF &-= after.X_l - before.X_l; \\ RF &-= 10\ after.f_l; \\ RF &-= \frac{after.R_l}{100}; \\ RF &-= after.D - before.D; \end{aligned} \tag{4}$$

where the numerical coefficients are used to balance the different objectives.

## 4. Results

The RL optimized policy has been compared to other O&M policies: $i$) a scheduled-maintenance FIFO policy in which maintenance interventions are scheduled at regular intervals with FIFO priority and $ii$) a predictive policy, in which the maintenance interventions are performed only if the turbine RUL estimation, $R$ is smaller than a user-defined threshold. In Table 1, we report the performance of the policies in terms of average profit and average number of maintenance interventions over 1000 test episodes.
The RL optimal policy is able to outperform the scheduled policy, which is considered to be the state-of-the-art for wind farm O&M (Nilsson Westberg and Bertling Tjernberg (2007); Barberá et al. (2013); Asensio et al. (2015); Pattison et al. (2016); Chan and Mo (2017)). The RL policy also overcomes the performance of the Predictive maintenance policy, which is one of the most studied approaches for maintenance of energy systems (de Novaes Pires Leite et al. (2018)). The RL agent is able to reduce the number of maintenanance interventions, reducing the maintenance costs and the production losses.

Table 1. Performance of the tested policies in terms of average profit and average number of maintenance interventions over 500 test episodes.

| Maintenance policy | Average profit | # maintenance |
|---|---|---|
| Scheduled | 674188 ± 12284 | 199.98 ± 0.14 |
| Predictive | 709560 ± 20823 | 203.31 ± 18.32 |
| RL policy | 747647 ± 17455 | 124.06 ± 9.54 |

Furthermore, the RL agent is also able to better select the time to ask for maintenance by exploiting the information about the predicted future power level: the average power level at maintenance is reduced by 16%, from $0.341 \pm 0.015$ (predictive policy) to $0.285 \pm 0.011$ (RL policy).

## 5. Conclusions

In this paper, we have shown that the application of RL to the optimization of the O&M of industrial systems equipped with PHM capabilities yields significant savings. In particular, we have tested the applicability of RL to the optimization of O&M of a scaled-down case study concerning a wind farm. The results have shown that the policy found by RL overcomes those of state-of-the-art approaches, as it finds more effective planning of the maintenance interventions.
We rely on a simulation model developed in AnyLogic, which is used by the the Pathmind Library to train the learning agents. This allows approaching RL without requiring specific knowledge on RL algorithms. AnyLogic allows to easily compare the performance of the RL policy to more complex heuristics by running experiments and visualizing the results.

## Acknowledgement

The RL training was faciliteted by Pathmind. Special thanks to Engineering Ingegneria Informatica for AnyLogic Professional software license.

## References

Arulkumaran, K., M. P. Deisenroth, M. Brundage, and A. A. Bharath (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine 34*(6), 26–38.

Asensio, S., J. M. Pinar Pérez, and F. P. García Márquez (2015, 01). *Economic Viability Study for Offshore Wind Turbines Maintenance Management*, Volume 362, pp. 235–244.

Barberá, L., A. Guerrero, A. Crespo Marquez, V. Gonzalez-Prida, A. J. Guillén Lopez, J. F. Gomez Fernandez, and A. Sola (2013, 01). State of the art of maintenance applied to wind turbines. Volume 33, pp. 931–936.

Bellani, L., M. Compare, P. Baraldi, and E. Zio (2019). Towards developing a novel framework for practical phm: A sequential decision problem solved by reinforcement learning and artificial neural networks. *Int. J. Prognostics Health Manag.*.

Carroll, J., A. Mcdonald, and D. Mcmillan (2015). Failure rate, repair time and unscheduled o&m cost analysis of offshore wind turbines. *Wind Energy 19*.

Chan, D. and J. Mo (2017). Life cycle reliability and maintenance analyses of wind turbines. *Energy Procedia 110*, 328 – 333. 1st International Conference on Energy and Power, ICEP2016, 14-16 December 2016, RMIT University, Melbourne, Australia.

de Novaes Pires Leite, G., A. M. Araújo, and P. A. C. Rosas (2018). Prognostic techniques applied to maintenance of wind turbines: a concise and specific review. *Renewable and Sustainable Energy Reviews 81*, 1917 – 1925.

Grondman, I., L. Busoniu, G. A. D. Lopes, and R. Babuska (2012). A survey of actor-critic reinforcement learning: Standard and natural policy gradients. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) 42*(6), 1291–1307.

Jaderberg, M., V. Dalibard, S. Osindero, W. M. Czarnecki, J. Donahue, A. Razavi, O. Vinyals, T. Green, I. Dunning, K. Simonyan, et al. (2017). Population based training of neural networks. *arXiv preprint arXiv:1711.09846*.

Kaelbling, L. P., M. L. Littman, and A. W. Moore (1995). An introduction to reinforcement learning. In L. Steels (Ed.), *The Biology and Technology of Intelligent Autonomous Agents*, Berlin, Heidelberg, pp. 90–127. Springer Berlin Heidelberg.

Nilsson Westberg, J. and L. Bertling Tjernberg (2007, 04). Maintenance management of wind power systems using condition monitoring systems—life cycle cost analysis for two case studies. *Energy Conversion, IEEE Transactions on 22*, 223 – 229.

Ozturk, S., V. Fthenakis, and S. Faulstich (2018). Failure modes, effects and criticality analysis for wind turbines considering climatic regions and comparing geared and direct drive wind turbines. *Energies 11*(9), 2317.

Pattison, D., M. D. S. Garcia, W. Xie, F. Quail, M. Revie, R. Whitfield, and I. J. Irvine (2016). Intelligent integrated maintenance for wind power generation. *Wind Energy 19*, 547–562.

Pinciroli, L., P. Baraldi, G. Ballabio, C. Compare, and E. Zio (2020). Deep reinforcement learning for optimizing operation and maintenance of energy systems equipped with phm capabilities. In *Proceedings of the 30th European Safety and Reliability Conference and the 15th Probabilistic Safety Assessment and Management Conference, 2020*.

Schulman, J., F. Wolski, P. Dhariwal, A. Radford, and O. Klimov (2017). Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Sutton, R. S. and A. G. Barto (2018). *Reinforcement Learning: An Introduction*.