Systems & Control Transactions

Research Article - Peer Reviewed Conference Proceeding
ESCAPE 35 - European Symposium on Computer Aided Process Engineering
Ghent, Belgium. 6-9 July 2025
Jan F.M. Van Impe, Grégoire Léonard, Satyajeet S. Bhonsale,
Monika E. Polańska, Filip Logist (Eds.)

PSE PRESS

MORL4PC: Multi-Objective Reinforcement Learning for Process Control

Niki Kotecha[a], Max Bloor[a], Calvin Tsay[b], and Antonio del Rio Chanona[a]*

[a] Department of Chemical Engineering, Sargent Centre for Process Systems Engineering, Imperial College London, SW7 2AZ, United Kingdom
[b] Department of Computing, Imperial College London, SW7 2AZ, United Kingdom
* Corresponding Author: a.del-rio-chanona@imperial.ac.uk.

ABSTRACT

In chemical process control, decision-making often involves balancing multiple conflicting objectives, such as maximizing production, minimizing energy consumption, and ensuring process safety. Traditional approaches for multi-objective optimization, such as linear programming and evolutionary algorithms, have proven effective but struggle to adapt in real-time to the dynamic and nonlinear nature of chemical processes. In this paper, we propose a framework that combines Reinforcement Learning (RL) with Multi-Objective Evolutionary Algorithms (MOEAs) to address these challenges. Specifically, we utilize MOEAs, such as NSGA-II, to optimize the parameter space of policy neural networks, resulting in a Pareto front of policies. This Pareto front provides a diverse set of policies that enable operators to dynamically switch control strategies based on real-time system conditions and prioritized objectives. Our proposed methodology is applied to a Controlled Stirred Tank Reactor (CSTR) case study, focusing on balancing the objectives of minimizing set point error and operational costs. The case studies highlight the adaptability of the method in dynamic environments, including scenarios involving fouling, showcasing its resilience and flexibility in complex, nonlinear systems. This work demonstrates the potential of combining RL and MOEAs to advance decision-making in chemical process control, providing a robust and adaptable framework for navigating environments with changing objectives. The methodology can also be extended to other industrial applications with similarly challenging multi-objective requirements.

Keywords: Process Control, Machine Learning, Industry 4.0, Reinforcement Learning

INTRODUCTION

In recent years, the field of Process Systems Engineering (PSE) has experienced a significant shift towards advanced optimization techniques that can handle the inherent complexities of modern systems, whether in process control, sustainable supply chains, or other industrial operations. A key challenge in these systems is optimizing multiple, often conflicting objectives —such as minimizing operational costs, ensuring process safety, improving product quality, and reducing environmental impact. This challenge has been amplified by a growing emphasis on sustainability and efficiency in industrial operations which highlights the need to develop decision-making frameworks that can handle dynamic environments with shifting priorities.

In industrial process control, balancing multiple objectives, such as optimizing productivity, reducing energy consumption, minimizing environmental impact, whilst maintaining safety is crucial. However, traditional control methods struggle to handle these competing goals, especially when unexpected disruptions occur, leading to emergency shutdowns and costly downtime.

Traditionally, industrial control problems have been addressed using model-based techniques, such as proportional-integral-derivative (PID) controllers, model predictive control (MPC), and multi-objective optimization (MOO) methods. These methods rely on mathematical models of system dynamics and predefined performance criteria to achieve optimal control. While effective in many industrial settings, they often struggle to manage the complexities of nonlinear, high-dimensional, and

dynamically changing environments.

For instance, PID controllers are limited in their ability to handle multi-variable interactions and time-varying dynamics. MPC, although more robust, can become computationally expensive and less effective when confronted with nonlinear stochastic systems or unexpected disturbances. Similarly, traditional MOO techniques rely on explicit trade-offs between objectives and are often static. These approaches can be inflexible, failing to adapt to evolving priorities in real-world decision-making where conflicting or shifting objectives are common.

Reinforcement Learning (RL) has emerged as a promising alternative to overcome these limitations. As a data-driven, model-free approach, RL enables agents to learn optimal control policies by interacting with the environment, making it well-suited for dynamic and nonlinear stochastic systems [1]. By framing control tasks as sequential decision-making problems, RL offers a natural framework for optimizing processes over time. This adaptability makes it particularly effective for applications such as chemical reactors, power grids, and supply chains, where system behavior can be complex and difficult to model explicitly.

Despite its advantages, standard RL algorithms are typically designed to optimize a single objective, which limits their effectiveness in scenarios requiring trade-offs among multiple competing goals. This limitation has driven the development of multi-objective reinforcement learning (MORL), an extension of RL tailored to handle multiple objectives simultaneously.

MORL methods can be broadly categorized into scalarization-based, Pareto-based, and decomposition-based approaches. Scalarization converts multiple objectives into a single reward function, often requiring prior knowledge of preference weights [2]. Pareto-based methods, such as Pareto Q-learning [3], maintain a set of non-dominated solutions, by learning a set of Pareto-optimal policies, enabling agents to explore multiple trade-offs without explicit weighting. Decomposition-based approaches, including MORL/D [4], break multi-objective problems into smaller subproblems and leverage reinforcement learning to solve them iteratively. Recent advances, such as deep reinforcement learning for multi-objective optimization (DRL-MOA) have further improved the scalability and generalization of MORL methods.

Multi-Objective Evolutionary Algorithms (MOEAs), for instance, offer a robust way to search across a broad solution space without requiring differentiable objective functions. Their ability to maintain a population of solutions, each addressing different trade-offs between objectives, makes them a natural fit for tackling multi-objective problems in dynamic environments. In this paper, we combine the adaptive decision-making capabilities of RL with MOEAs' ability to balance diversity and proximity to the Pareto front. Specifically, we use MOEAs to search the parameter space of neural network policies, resulting in a Pareto front of policies. This equips the decision-maker with a swarm of policies, allowing for dynamic switching between policies based on the current system objectives and dynamics. This strategy ensures flexibility in real-time decision-making under uncertainty, allowing the system to adapt quickly to changing conditions without the need to evolve policies continuously.

The rest of the paper is organized as follows: the preliminaries section provides the background on reinforcement learning and evolutionary strategies, the methodology section describes the proposed multi-objective evolutionary algorithm-based reinforcement learning framework in more detail. The case study section discusses the simulation and experimental results and finally, we summarize the paper and provide an outlook for future work.

PRELIMINARIES

Introduction to Reinforcement Learning

In single agent reinforcement learning, the agent aims to learn an optimal policy by interacting with the environment and learning through trial-and-error. In RL, the agent observes the current state $x_t \in X \subseteq \mathbb{R}^{n_x}$, chooses an action $u \in U \subseteq \mathbb{R}^{n_u}$ with probability given by the policy $\pi(u|x)$ and transitions into the next state, $x_{t+1} \in X \subseteq R^{n_x}$ with probability given by the state transition probability function $P(x_{t+1}|x_t,u_t)$ and receives a reward $r_{t+1} = \mathcal{R}(x_t,u_t,x_{t+1}) \in \mathbb{R}$. The agent finds an optimal policy π^* by maximizing the expected sum of rewards over a time horizon defined as:

$$J(\pi) = \mathbb{E}_{(x_t,u_t)\sim\pi}\left[\sum_{t=0}^{T-1}\gamma^t r_{t+1}(x_t,\, u_t = \pi(x_t))|\right]$$

$$\pi^* \in \arg\max_{\pi} J(\pi)$$

Where $\gamma \in [0,1]$. In practice, the policy is parameterized such that $\pi^* \approx \pi^*(u|x;\theta)$, where $\theta \in \Theta \subseteq \mathbb{R}^{n_\theta}$ represents the weights of the neural network and Θ is the parameter space. The aim is to optimize parameters θ to maximize the cumulative rewards, learning an effective mapping from states x to actions u.

To achieve this, policy-gradient methods optimize parameters θ by estimating gradients of the expected reward with respect to parameters θ. This approach contrasts to value-based methods, as they directly update the policy, making it suitable for continuous action spaces and high dimensions. Algorithms like REINFORCE, Proximal Policy Optimization (PPO), and Trust Region Policy Optimization (TRPO) are popular choices here, as they leverage policy gradients and surrogate objectives to improve stability and performance [5,6]. However, policy gradient methods rely on sampling rollouts to

estimate gradients. These samples can exhibit large variance due to the stochastic nature of the environment and policy, leading to noisy gradient estimates.

An alternative approach is using Evolutionary Strategies (ES). ES methods optimize parameters θ by exploring the parameter space Θ through simulating an evolutionary process. Rather than relying on gradient-based updates, ES methods employ population-based search techniques to evolve policies. In each iteration, a population of candidate policies is evaluated based on a reward metric, with the most successful candidates selected to "reproduce" via mutation and recombination, guiding the search toward high-performing areas of the parameter space.

Evolutionary strategies for reinforcement learning (ES-RL) have shown promising results in optimizing policy parameters due to their lack of need for backpropagation. Instead of relying on gradient information, ES-RL evaluate policy performance by sampling multiple trajectories and directly update the parameters towards those that lead to higher performance. This method enhances scalability in distributed settings as its easier to parallelize. There's also fewer hyperparameters to tune compared to gradient-based methods and they are less likely to get stuck in local optima as they are population-based methods so search "globally" rather than relying on stochastic estimates of gradients to optimize the parameters. Although ES-RL has demonstrated competitive performance, as highlighted in OpenAI's work [7], where it showed comparable results to other policy gradient methods on environments like MuJoCo and Atari, it is important to note that ES-RL does not universally outperform policy gradient methods. Rather, ES-RL is particularly advantageous in specific contexts, especially when extended to multi-objective evolutionary reinforcement learning (MOEA-RL). The population-based search techniques of ES-RL can be leveraged to handle multiple, conflicting objectives in complex, dynamic environments providing a framework for optimization problems where traditional RL methods may face limitations.

METHODOLOGY

This section presents the proposed multi-objective reinforcement learning framework which integrates traditional reinforcement learning with multi-objective evolutionary strategies.

MORSE - Multi-Objective Reinforcement learning via Strategy Evolution

Due to the aforementioned benefits of evolutionary strategies, the proposed methodology leverages multi-objective evolutionary strategies (MOEA) for multi-objective reinforcement learning (MORL). In this work, we leverage NSGAA-II [8] to directly optimize the parameter space of the policies, building a Pareto set of policies rather than a Pareto set of solutions, as is common in traditional multi-objective optimization methods. This approach provides the decision maker with a diverse set of adaptable and dynamic policies, facilitating rapid decision-making in complex environments. The proposed methodology is shown in Algorithm 1 and Figure 1.

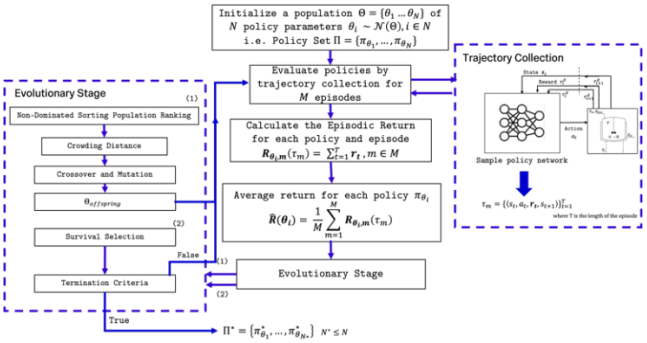

Figure 1. Schematic overview of our MORSE framework.

This results in a Pareto front of policies rather than a single policy. This allows decision-makers to have the flexibility to switch between policies and choose one that gives the best trade-off according to the needs in real-time. This is valuable in dynamic environments where priorities may shift due to external disruptions to the system [3,5,9].

Multi-Objective Markov Decision Process

The sequential decision-making problem is formulated as a Multi-Objective Markov Decision Process, which can be defined as a tuple $\langle X, U, \mathcal{T}, \gamma, \mu, \mathbf{R} \rangle$ where X is the state space, U is the action space, $\mathcal{T}$ is the probability transition function where $\mathcal{T} : X \times U \times X \to [0,1]$, $\mu : X \to [0,1]$ is a probability distribution over initial states and $R : X \times U \times X \to R^d$ is a vector valued reward function where $d \geq 2$ is the number of objectives. The vector-valued reward function R is one of the differences between single-objective RL and multi-objective RL. Finally, a policy $\pi : X \to U \in \Pi$, maps states to actions where Π is a set of all the possible policies. Another notable difference between multi-objective and single-objective MDPs is the vector valued value function, $V^\pi \in R^d$ which is conditioned on the number of objectives and is defined as $V^\pi(x) = \mathbb{E}_\pi \sum_{t=0}^{T-1} \gamma^t r_{t+1}$ where $r_{t+1} = R(x_t, u_t, x_{t+1})$.

A note that due to the multi-objective nature of the definition, it is possible to encounter a situation where for objectives j and k where $j, k \in \{1, 2, \dots, d\}$ and policies π and π' where $\pi, \pi' \in$, both of the following inequalities can hold true:

$$V_j^\pi(x) > V_j^{\pi'}(x) \text{ Policy } \pi \text{ is better for objective } j$$

$$V_k^\pi(x) < V_k^{\pi'}(x) \text{ Policy } \pi' \text{ is better for objective } k$$

This indicates that while policy π outperforms policy π' in

objective j, it underperforms in terms of objective k. This illustrates the concept of trade-offs in multi-objective reinforcement learning, where optimizing for one objective can lead to suboptimal performance in another. Therefore, the agent must make decisions based on the relative importance of each objective.

Computational Case Study

The modelled system is a Controlled Stirred Tank Reactor (CSTR), a complex and nonlinear dynamical system which can be modelled using two ordinary differential equations. A schematic is also shown in Figure 2.

$$\frac{dC_A}{dt} = \frac{q}{V}\left(C_{A_f} - C_A\right) - kC_A e^{\frac{-E_A}{RT}}$$

$$\frac{dT}{dt} = \frac{q}{V}\left(T_f - T\right) - \frac{\Delta H_R}{\rho C_p} kC_A e^{\frac{-E_A}{RT}} + \frac{UA}{\rho C_p V}\left(T_c - T\right)$$

The equations above describe a CSTR where an irreversible chemical reaction occurs which converts species A to B with cooling provided by water flowing through a coiled tube

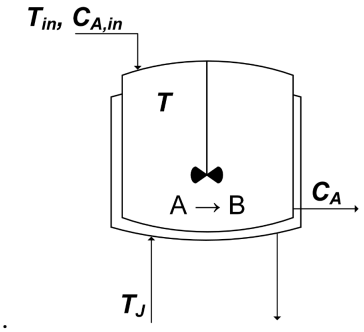

Figure 2. Schematic overview of Controlled Stirred Tank Reactor (CSTR)

The state variables are the concentration of species A, C_A and the reactor temperature T, which constitute the state representation in our RL framework. At any given time step, the state can be expressed as a vector $x_t = [C_A, T]$, which provides the agent with the necessary information about the system's current conditions. The state variables evolve dynamically according to the system's governing equations.

Our system features one input control variable, the cooling water temperature T_c, modelled as a continuous variable.

Due to the non-linear dynamics, evident in the exponential term which includes the reactor temperature T, strong coupling between state variables, and potential for instability, the control of the CSTR presents significant challenges [10]. These difficulties are further exacerbated in systems with multiple conflicting objectives and when objectives change over time depending on the dynamics of the system. For instance, optimizing reactor performance might involve simultaneously maintaining a desired product concentration, minimizing energy consumption, and ensuring safe operating conditions, which often conflict with one another.

Algorithm 1 MORSE

Input: Number of policies N, Maximum generations G, Evaluation episodes E
Output: Pareto front set of policies $\mathcal{F}_{\text{Pareto}}$

Step 1: Initialization
Generate a population of policies $\Pi = \{\pi_1, \pi_2, \ldots, \pi_N\}$, where each policy π_i is parameterized by θ_i
Step 2: Policy Evaluation
for each policy $\pi_i \in \Pi$ **do**
  for each episode $e = 1, 2, \ldots, E$ **do**
    Initialize state s_0 from the environment
    for each time step $t = 0, 1, \ldots, T$ **do**
      Select action $a_t \sim \pi_{\theta_i}(s_t)$ based on the policy
      Execute action a_t, observe reward r_t, next state s_{t+1}
      Accumulate discounted reward for each objective
    end for
  end for
  Compute average objective values over E episodes
end for
Step 3: Non-dominated Sorting
Sort policies into fronts based on dominance relationships: $\mathcal{F}_1, \mathcal{F}_2, \ldots$
Step 4: Crowding Distance Calculation
for each front $\mathcal{F}_j$ **do**
  Compute crowding distance $d(\pi_i)$ for each policy π_i
end for
Step 5: Selection and Reproduction
Binary tournament selection, then crossover & mutation:
$\theta_{\text{offspring}} = \text{Crossover}(\theta_i, \theta_j) + \text{Mutation}(\theta_k)$
Step 6: Survival Selection
Combine parent population $\mathcal{P}$ and offspring $\mathcal{P}_{\text{offspring}}$. Select top N policies based on non-domination rank and crowding distance:
$\mathcal{P}' = \text{Top-}N(\mathcal{P} \cup \mathcal{P}_{\text{offspring}})$
Step 7: Termination Criteria
if Termination criteria met **then**
  break
end if
Step 8: Pareto Front Identification
Identify non-dominated solutions in the final population:
return Pareto front set $\mathcal{F}_{\text{Pareto}}$; $\Pi = \{\pi_1^*, \pi_2^*, \ldots, \pi_{n*}^*\}$

In this study, our CSTR model integrates two cumulative objective functions throughout the time horizon: Minimize the set point error and minimize the costs (composed of heat costs and emergency shutdown costs). The reward function therefore is a vector-valued function at each time step:

$$\boldsymbol{R}_t = \left[\left(C_A(t) - C_A^*(t)\right)^2, \alpha\left(T_c - T(t)\right)^2 + \beta\mathbb{I}(shutdown)\right]$$

The first term represents the squared error between the actual concentration of species A and the desired set point concentration, $C_A^*(t)$. The second term captures the heat costs related to the deviation between the cooling temperature T_c and reactor temperature $T(t)$ where α is a constant; the term $\beta\mathbb{I}(shutdown)$ accounts for emergency shutdown costs where $\mathbb{I}(shutdown)$ is an indicator function that takes value 1 if an emergency shutdown takes place and 0 otherwise. The cost coefficient β reflects the severity of this event.

The goal is to ascertain the optimal control action during each time period t spanning over a total of T time periods within a discrete-time setup. The environment is implemented by adapting PC-Gym [10].

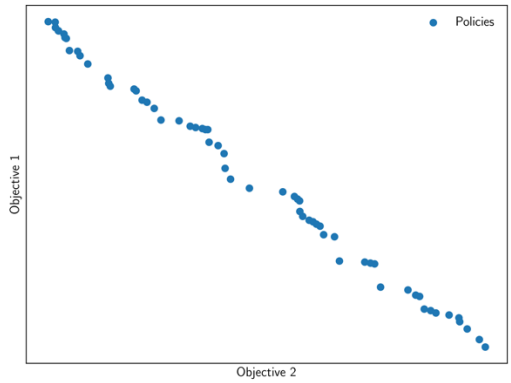

Figure 3. The resulting Pareto front showing the set of non-dominated policies, illustrating the trade-off between two competing objectives.

SCENARIOS

In this section, we examine the adaptability of our methodology through several case studies. Using our approach, we derive a Pareto set of policies, as shown in Figure 3, that optimally balance competing objectives. When a disruption affects the system, this Pareto set allows for swift policy switching, enabling us to select a policy that best meets the real-time needs and constraints. By dynamically adjusting to changing conditions, we demonstrate how our method offers resilience and flexibility in complex environments.

Scenario 1: Fouling

Fouling is a common issue in chemical processes, particularly in CSTRs, where unwanted material accumulates on the reactor walls or heat exchanger surfaces. This buildup can significantly reduce heat transfer efficiency, alter reaction rates, and increase operating costs due to the need for additional energy to maintain the desired temperature.

In this case study, we simulate fouling by introducing a time-dependent reduction in the heat transfer coefficient. As fouling progresses, maintaining optimal product concentrations and minimizing energy costs become increasingly challenging. Using our methodology, we analyze how the Pareto set of policies adapts to these conditions. For instance:

- Policies that prioritize minimizing set-point error may involve increasing the heat input to compensate for the reduced heat transfer efficiency.
- Policies that that prioritize cost minimization may tolerate larger deviations from the desired set-point to avoid excessive energy consumption.

The results shown in Figure 4 demonstrate that the proposed approach can dynamically adjust control actions to balance competing objectives under fouling conditions, ensuring safe and efficient reactor operation. The dynamic policy significantly outperforms the constant policy in minimizing set point error (Objective 1), showcasing its ability to respond quickly to disturbances and maintain the system closer to the desired operating range. However, the difference in costs (Objective 2) between the dynamic and constant policies is relatively small, indicating that both approaches manage costs effectively under normal fouling conditions.

This small difference in costs suggests that cost optimization may not be as sensitive to dynamic control under routine disruptions. However, costs could become a much more prominent factor in scenarios where fouling leads to severe disruptions, such as an emergency shutdown or significant system failure.

Figure 4. Execution profiles of the fouling case study, showing objective 1 (set point error) and objective 2 (costs) with shaded variance.

CONCLUSIONS AND FUTURE WORK

In conclusion, we have proposed a methodology which leverages reinforcement learning (RL) and multi-objective evolutionary algorithms (MOEAs) to address the challenges of controlling a CSTR with competing objectives. By leveraging a Pareto-based approach, by using MOEAs to search the parameter space of the neural network policies, we successfully derive a set of non-dominated policies that balance objectives such as minimizing set-point error and operational costs. The case studies highlight the adaptability of the method in dynamic environments, including scenarios involving fouling showcasing its resilience and flexibility in complex

systems.

Future work will focus on improving the efficiency of the evolutionary search process by integrating domain-specific knowledge and operator expertise. Additionally, we will investigate collaborative decision-making strategies across interconnected processes by extending this work to multi-unit reactor systems. We also plan to adapt our method to handle partially observable environments, which are often encountered in industrial settings with incomplete measurements. Finally, the incorporation of additional objectives such as emissions control, stability and long-term maintenance costs, will further enhance its relevance to real-world applications. Source code for this study is available on GitHub.

REFERENCES

1. Bloor, M.; Ahmed, A.; Kotecha, N.; Tsay, C.; Mercangöz, M.; Del Rio-Chanona, A. Hierarchical Reinforcement Learning for Plantwide Control. In Computer Aided Chemical Engineering; Elsevier, 2024; Vol. 53, pp 1639–1644. https://doi.org/10.1016/B978-0-443-28824-1.50274-X
2. Van Moffaert, K.; Drugan, M. M.; Nowe, A. Scalarized Multi-Objective Reinforcement Learning: Novel Design Techniques. In 2013 IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning (ADPRL); IEEE: Singapore, Singapore, 2013; pp 191–199. https://doi.org/10.1109/ADPRL.2013.6615007
3. Van Moffaert, K.; Nowé, A. Multi-Objective Reinforcement Learning Using Sets of Pareto Dominating Policies. J. Mach. Learn. Res. 2014, 14 (1), 3483–3512.
4. Felten, F.; Talbi, E.-G.; Danoy, G. Multi-Objective Reinforcement Learning Based on Decomposition: A Taxonomy and Framework. J. Artif. Intell. Res. 2024, 79, 679–723. https://doi.org/10.1613/jair.1.15702
5. Hayes, C. F.; Radulescu, R.; Bargiacchi, E.; Källström, J.; Macfarlane, M.; Reymond, M.; Verstraeten, T.; Zintgraf, L. M.; Dazeley, R.; Heintz, F.; Howley, E.; Irissappane, A. A.; Mannion, P.; Nowé, A.; Ramos, G.; Restelli, M.; Vamplew, P.; Roijers, D. M. A Practical Guide to Multi-Objective Reinforcement Learning and Planning. Auton. Agents Multi-Agent Syst. 2022, 36 (1), 26. https://doi.org/10.1007/s10458-022-09552-y
6. Kotecha, N.; Chanona, A. del R. Leveraging Graph Neural Networks and Multi-Agent Reinforcement Learning for Inventory Control in Supply Chains. arXiv October 24, 2024. https://doi.org/10.48550/arXiv.2410.18631
7. Salimans, T.; Ho, J.; Chen, X.; Sidor, S.; Sutskever, I. Evolution Strategies as a Scalable Alternative to Reinforcement Learning. arXiv September 7, 2017. https://doi.org/10.48550/arXiv.1703.03864
8. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Trans. Evol. Comput. 2002, 6 (2), 182–197. https://doi.org/10.1109/4235.996017
9. Qiu, Y.; Kotecha, N.; Del Rio Chanona, A. Leveraging Reinforcement Learning and Evolutionary Strategies for Dynamic Multi Objective Decision Making in Supply Chain Management. IFAC-Pap. 2024, 58 (14), 598–603. https://doi.org/10.1016/j.ifacol.2024.08.402
10. Bloor, M.; Torraca, J.; Sandoval, I. O.; Ahmed, A.; White, M.; Mercangöz, M.; Tsay, C.; Chanona, E. A. D. R.; Mowbray, M. PC-Gym: Benchmark Environments For Process Control Problems. arXiv December 5, 2024. https://doi.org/10.48550/arXiv.2410.22093