MORL4Water: A Modular Multi-Objective Reinforcement Learning Toolkit for Water Resource Management

Zuzanna Osika
Delft University of Technology
Delft, Netherlands
z.osika@tudelft.nl

Roxana Rădulescu
Utrecht University
Utrecht, Netherlands
r.t.radulescu@uu.nl

Jazmin Zatarain-Salazar
Delft University of Technology
Delft, Netherlands
j.zatarain-salazar@tudelft.nl

Frans A. Oliehoek
Delft University of Technology
Delft, Netherlands
f.a.oliehoek@tudelft.nl

Pradeep K. Murukannaiah
Delft University of Technology
Delft, Netherlands
p.k.murukannaiah@tudelft.nl

ABSTRACT

Many real-world decision problems involve conflicting objectives. Multi-objective reinforcement learning (MORL) extends standard RL to optimize multiple objectives simultaneously, producing policy sets that capture different trade-offs. However, MORL research often relies on simplified benchmarks with limited real-world relevance. We present MORL4Water, a modular toolkit for creating realistic MORL environments in water resource management. Built on MO-Gymnasium, MORL4Water enables scenario construction from real data and systematic evaluation of MORL methods. We illustrate its use on the Nile and Susquehanna rivers, benchmarking several MORL algorithms against EMODPS, a domain-specific baseline. Beyond standard performance metrics, we analyze solution sets to reveal differences in exploration, scalability, and trade-off diversity. Our results show that most state-of-the-art MORL algorithms underperform relative to EMODPS, especially in higher-dimensional settings, and highlight the value of solution-set analysis for robust, real-world applications.

CCS CONCEPTS

• **Theory of computation → Sequential decision making**.

KEYWORDS

Multi-objective reinforcement learning; water management; sustainability; simulations; benchmarks

ACM Reference Format:
Zuzanna Osika, Roxana Rădulescu, Jazmin Zatarain-Salazar, Frans A. Oliehoek, and Pradeep K. Murukannaiah. 2026. MORL4Water: A Modular Multi-Objective Reinforcement Learning Toolkit for Water Resource Management. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 10 pages. https://doi.org/10.65109/VSUW5215

1 INTRODUCTION

Many real-world decision problems are inherently multi-objective: they require balancing several, often conflicting goals rather than optimizing a single outcome. Despite this ubiquity, most algorithms for autonomous agents interacting with sequential decision problems still assume a single objective. MORL [10] addresses this limitation by extending the reinforcement learning (RL) framework [35] to learn from vector-valued rewards, with each component representing feedback for a different objective. Rather than producing a single optimal policy, MORL yields sets of policies that represent different trade-offs. Evaluating and reasoning about these solution sets is therefore central to understanding the capabilities of MORL agents and their applicability in complex real-world domains.

The ability to capture and explore trade-offs makes MORL particularly relevant for real-world decision-making, where conflicts among objectives are unavoidable. MORL has been studied in diverse domains, including water management [3, 11], autonomous driving [19], power allocation [22, 39], drone navigation [38], and medical treatment [17, 20], demonstrating its potential in settings where stakeholders must balance multiple competing goals.

MORL is a growing research field with many recent developments in novel algorithms (e.g., [1, 9, 27, 40]). However, the current benchmarking efforts usually involve abstract, toy-like, problems and do not support or demonstrate the potential of MORL algorithms for real world applications. As a result, we do not know whether the state-of-the-art algorithms could solve real-world problems or not, as scalability has been a challenging aspect of MORL, in terms of state and actions spaces, and the number of objectives.

To address this challenge, we introduce MORL4Water[1], a toolkit that enables systematic benchmarking of MORL algorithms in the realistic and high-impact domain of water resource management. Managing water resources is a complex and critical challenge, made increasingly difficult by the ongoing effects of climate change. It can be framed as an RL problem, where decisions must be made at each time step about how much water to release from one or more dams on a water resource such as a river. Effective water management requires balancing multiple, often conflicting, objectives such as hydropower generation, irrigation, or water supply.

Water resource management has been modeled as a sequential multi-objective decision-making problem in various real-world cases, including the Hoa Binh reservoir in northern Vietnam [4], the Lower Susquehanna River in the USA [12], Lake Como in Italy [42], and the Lower Volta River in Ghana [24]. Solutions have employed methods such as dynamic programming, reinforcement learning,

[1] https://github.com/osikazuzanna/morl4water

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus. © 2026 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). https://doi.org/10.65109/VSUW5215

and mathematical optimization (for a comprehensive review, see [16] or [13]). In the domain of water resource planning and management, researchers have developed a specialized algorithm, called evolutionary multi-objective direct policy search (EMODPS) [11], which has demonstrated strong performance in real-world applications [25, 29]. These solutions rely on river system simulations. However, access to such simulations is limited—data and code are not openly available, are written in various programming languages, and the code structures vary between researchers.

In contrast, MORL4Water is an open-source, modular toolkit. It extends the MO-Gymnasium API [8] to provide customizable environments for water resource management. MORL4Water offers water management researchers a structured, reusable framework for building and comparing simulations, and we demonstrate its capabilities with two case studies focused on the Nile and Susquehanna rivers. We benchmark these environments on a subset of the state-of-the-art MORL algorithms, analyzing their performance as well as the quality of the resulting solution sets. While single-objective RL can often be assessed solely through the performance of one policy, MORL requires reasoning about sets of policies and the trade-offs they represent. Yet, most existing work evaluates algorithms through scalar indicators, partly because common benchmarks are toy-like and do not necessitate deeper analysis. By combining standard performance metrics with a solution set analysis, we provide the first benchmarking and direct comparison of general-purpose MORL algorithms with a domain-specific approach in realistic water management contexts. The results highlight both the promise of MORL and its limitations in scalability and exploration, offering valuable insights and establishing a foundation for future studies.

Our contributions are as follows: (1) We develop MORL4Water, a modular MO-Gymnasium environment for creating customizable, real-world water resource management scenarios, demonstrated with two case studies. (2) We benchmark several state-of-the-art MORL algorithms and compare them to EMODPS on these environments. In doing so, we not only evaluate algorithmic performance but also conduct a structured analysis of the resulting solution sets, highlighting both the potential and the limitations of MORL methods for realistic multi-objective decision-making. MORL4Water grounds research in real-world applications, advancing evaluation and highlighting paths to practical impact in sustainability.

2 BACKGROUND: MULTI-OBJECTIVE REINFORCEMENT LEARNING

We formalize a MORL problem as a multi-objective Markov decision process (MOMDP), defined as the tuple $\langle S, A, T, \gamma, \mu, \mathbf{R}\rangle$, where S is the state space, A the action space, $T\colon S \times A \times S \to [0, 1]$ is a probabilistic transition function, $\gamma \in [0, 1)$ is the discount factor, and $\mu\colon S \to [0, 1]$ defines a probability distribution over initial states. The function $\mathbf{R}\colon S \times A \times S \to \mathbb{R}^d$ represents a vector-valued reward function that specifies the immediate reward for each of the $d \geq 2$ objectives [15].

The vector-valued reward function $\mathbf{R}$ is the main difference from Markov Decision Processes, returning a numeric feedback signal for each composing objective. The agent behaves according to policy $\pi : S \times A \to [0, 1]$. The value function of a policy π in a MOMDP is defined as:

$$\mathbf{V}^{\pi} = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma^k \mathbf{r}_{k+1} \mid \pi, \mu\right], \tag{1}$$

where $\mathbf{r}_{k+1} = \mathbf{R}(s_k, a_k, s_{k+1})$ is the reward received at timestep $k+1$. Since the value function is a vector, $\mathbf{V}^{\pi} \in \mathbb{R}^d$, it can only offer a partial ordering over the policy space. Determining the optimal policy requires additional information on how to prioritise the objectives. We can capture such a trade-off choice using a *utility function*, $u : \mathbb{R}^d \to \mathbb{R}$, that maps the vector to a scalar value. If u is unknown, approaches can adopt the multi-policy paradigm and return an entire solution set.

Solution sets. In the most general case, the **Pareto set** is defined as the optimal solution set, under the minimal assumption that u is monotonically increasing and is based on Pareto dominance. Informally, Pareto dominance ($\succ_P$) introduces a partial ordering over vectors, where one vector is preferred over another when it is at least equal on all objectives and strictly better on at least one. Let Π be a set of policies. The Pareto set $\mathcal{P}(\Pi)$ then contains all pairwise undominated policies, i.e.,

$$\mathcal{P}(\Pi) = \{\pi \in \Pi \mid \nexists\, \pi' \in \Pi : \mathbf{V}^{\pi'} \succ_P \mathbf{V}^{\pi}\} \tag{2}$$

The **Pareto front** (PF), denoted as $\mathcal{F}(\mathcal{P})$, contains the value vectors corresponding to all Pareto optimal policies $\pi \in \mathcal{P}(\Pi)$. If u is a positively-weighted linear sum, then the solution set will be the **convex hull** (CH) of value functions $\mathbf{V}^{\pi}$.

3 MORL4WATER TOOLKIT

We provide an overview of Water Management Systems (WMS) and the two benchmark case studies used in our experiments. Following this brief introduction, we present an overview of the toolkit. For a detailed description, please refer to Appendix A[2]

3.1 Water Resource Management

A WMS encompasses the key components involved in managing a water resource, such as a river. We begin by identifying the core components common to most WMSs, which form the foundation of our proposed toolkit. Figure 1a illustrates the range of components that can be included in a WMS.

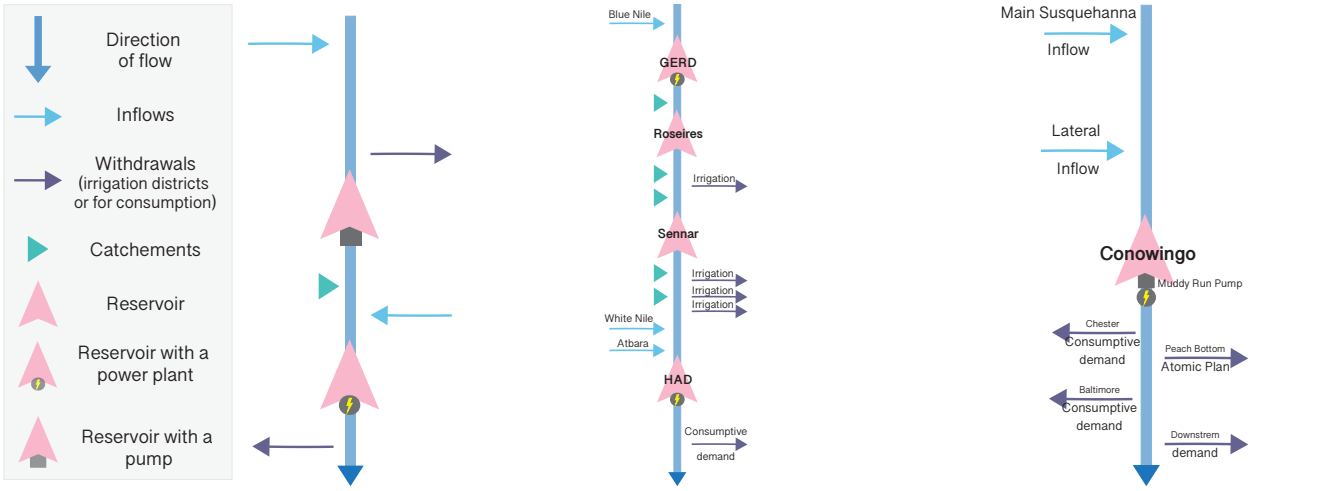
Reservoirs (or dams) are key components of any WMS, serving as control points–water release decisions are made here based on inflow, outflow, storage volume, and physical constraints. They influence the system state by storing and releasing water, while also contributing to water loss through surface evaporation [6].

Inflow refers to the volume of water entering a facility from upstream sources, including contributions from catchments or tributaries, typically estimated using historical river flow data.

Outflow refers to the amount of water leaving the modeled facilities, primarily representing the water released from a reservoir to meet, for instance, specific water demands.

Irrigation Districts are sources of irreversible water consumption, e.g., urban areas, where a portion of the flow is permanently diverted based on water availability and demand.

[2] https://github.com/osikazuzanna/morl4water/blob/main/appendix.pdf



(a) Model of a WMS. (b) Abstract model of Nile WMS. (c) Abstract model of Susquehanna WMS.

Figure 1: Models of water management systems: (a) general WMS, (b) Nile WMS, and (c) Susquehanna WMS.

Power Plants are modelled to calculate the amount of hydroelectric energy generation from dams.

Catchment refers to an area where precipitation is collected by the natural landscape [32]. It represents an inflow of water into the system, contributing to a net gain.

Release Constraints Release constraints capture physical, environmental, and regulatory limitations by enforcing minimum and maximum release bounds, which vary based on the facility's current state.

Time-Dependent Data reflects the system's climate and regional conditions that vary over time. For example, inflows may fluctuate on a monthly basis, while evaporation rates can change daily.

In our toolkit, WMS problems are modeled as MOMDPs, where actions correspond to release decisions from each reservoir, altering the system's state based on the mass-balance equation for each reservoir [13]:

$$v_{t+1} = v_t + q_{t+1} - r_{t+1}, \tag{3}$$

where v_t is the water volume in the reservoir at the start of decision step t, q_{t+1} is the net inflow (i.e.,inflow and direct precipitation minus evaporation and seepage losses)[3] and r_{t+1} is the water released, all occurring between steps t and $t+1$.

The decision interval, defined by the system designer, determines how often release decisions are updated (e.g., hourly, daily, or monthly), while the integration timestep specifies finer intervals for updating the system state to capture dynamic processes like evaporation or precipitation. Since release decisions r_t must comply with physical and normative constraints, they are recalculated over each decision interval using system states integrated at higher temporal resolution.

[3]we model net inflow as a system disturbance (i.e., $q_{t+1} = \epsilon_{t+1}$, which aggregates multiple sources of uncertainty, such as main tributaries, distributed inflow runoff, evaporation, and precipitation over the reservoir

3.2 Case Studies

We introduce two representative case studies (see Figures 1b and 1c for their abstract models) and their associated objectives.

Nile WMS. The Nile river is a crucial water resource in northeast Africa, supporting hydropower, agriculture, and municipal needs across ten countries. Political tensions over water rights have persisted, especially between Egypt, Sudan, and Ethiopia. Egypt, historically the largest user of Nile water, faces challenges from the construction of the Grand Ethiopian Renaissance Dam (GERD), which aims to harness hydropower. While Ethiopia insists that the GERD would not impact downstream flows, Sudan and Egypt see it as a threat to water security, particularly regarding the reservoir filling period. Negotiations remain deadlocked. The river also faces issues related to variable water inflow, frequent droughts and floods, and rising water demand due to population growth.

Previous studies have applied optimization techniques to reservoir operations in the Nile basin, with one of the most cited foundational papers utilizing data from the High Aswan Dam (HAD) in Egypt as a test case [34]. Several other studies have also employed reservoirs in the Nile basin for methodological contributions, but they often lack a deep examination of the real-world implications in the region [30, 33]. However, the number of studies focusing on reservoir modeling with a policy analytics perspective has notably increased since the launch of GERD project in 2011. We build upon Sari [31] and Wheeler et al. [37], regarding the objectives and the data utilized. The former applied EMODPS, while the latter modeled the problem using predefined release rules for different simulations.

Our WMS for Nile environment focuses on three main actors with national governments representing their respective interests.

Ethiopia, a predominantly rural country, faces significant challenges in water management, with only 27% of its population having access to electricity [37]. The country considers effective water management essential for economic growth and contests historical water-sharing treaties, asserting that it was excluded from the

agreements made between Egypt and Sudan. Ethiopia's unilateral construction of GERD is intended to enhance hydropower generation and foster economic development.

Egypt enjoys near-universal access to electricity and clean water, with the highest GDP per capita among Nile Basin countries. It has historically controlled Nile water through the HAD, but suffers water losses due to evaporation. Egypt opposes GERD, fearing it will reduce downstream water flow, threatening food and energy security, though the dam's focus on hydropower offers some relief.

Sudan has a robust agricultural economy supported by substantial irrigation infrastructure. Initially aligned with Egypt in opposing the GERD, Sudan's position has evolved to acknowledge the potential benefits of regulating Blue Nile flows to reduce flooding. Yet, Sudan continues to express concerns on water security.

Accordingly, we consider four objectives as detailed in Table 1.

Objective	Unit
Egypt irrigation deficit (↓)	Demand deficit in cubic meters per sec.
Egypt min HAD level (↑)	Number of months equal or above minimum power generation level
Sudan irrigation deficit (↓)	Demand deficit in cubic meters per sec.
Ethiopia hydropower (↑)	Total hydro-energy generation in MWh

Table 1: Objectives in the Nile WMS (↑ represents maximization and ↓ represents minimization).

Susquehanna WMS. Susquehanna is the longest river in the eastern US, draining nearly 71,000 km^2 and supplying 50% of the freshwater to the Chesapeake Bay. To harness this substantial water flow, the Conowingo dam, one of the largest non-federally regulated hydroelectric dams in US, was built in 1928. This dam plays a crucial role in regulating flow, and serves the diverse stakeholders dependent on its water supply.

Currently, the dam provides water to Chester, PA, and Baltimore, MD, as well as cooling water for the Peach Bottom nuclear power station while adhering to minimum flow requirements established by the Federal Energy Regulatory Commission (FERC) to protect fish resources. The Muddy Run Pumped Storage Facility, built in 1968, enhances power generation by cycling water between its reservoir and the Conowingo reservoir. In average flow conditions, water availability typically meets the needs of hydroelectric operations, water supply, and recreation. However, low flow conditions create difficult trade-offs for Conowingo's operations as they strive to balance water supply for Baltimore and Chester while minimizing impacts on recreational and tourism activities [12].

Our WMS environment of Susquehanna is based on the simulation by Giuliani et al. [12] and Zatarain Salazar et al. [43]. The simulation focuses on identifying an operating policy for the Conowingo dam, while the Muddy Run facility operates under a predetermined weekly rule. This rule outlines a hydropeaking strategy, where turbines function during peak energy hours and pumps operate at night and on weekends when energy prices are lower. The simulation represents the dynamics of the two water reservoirs, guided by the mass balance equations for the water volume in both the Conowingo and Muddy Run reservoirs.

Accordingly, we consider six objectives detailed in Table 2.

Objective	Unit
Recreation (↑)	Number of peak season weekend days at or above the target recreational level of 32.5 m (106.5 ft)
Environmental Shortage (↓)	Average shortage index in regard to the FERC minimum flow requirements
Water supply reliability to Baltimore (↑)	The ratio of water supplied to demand in Baltimore, based on historical data in cubic feet per second, capped at 1 if supply exceeds demand [12]
Water supply reliability to Chester (↑)	The ratio of water supplied to demand in Chester, based on historical data in cubic feet per second, capped at 1 if supply exceeds demand [12]
Water supply reliability to Atomic Plant (↑)	The ratio of water supplied to the cooling demand of the power plant, based on historical data in cubic feet per second, capped at 1 if supply exceeds demand [12]
Hydropower revenue (↑)	Revenue from hydropower at Conowingo (US$/MWh), using a 7-hour moving average of PJM prices [36]

Table 2: Objectives in the Susquehanna WMS.

3.3 MORL4Water Toolkit

Figure 2 shows the main classes in the MORL4Water toolkit. The `WaterManagementSystem` class, which implements the MO-Gymnasium API [9], is the entry-point to the toolkit. It employs additional classes—`Flow`, `Facility`, and `ControlledFacility`—to model different components of a WMS. We provide important details on these classes in this section.

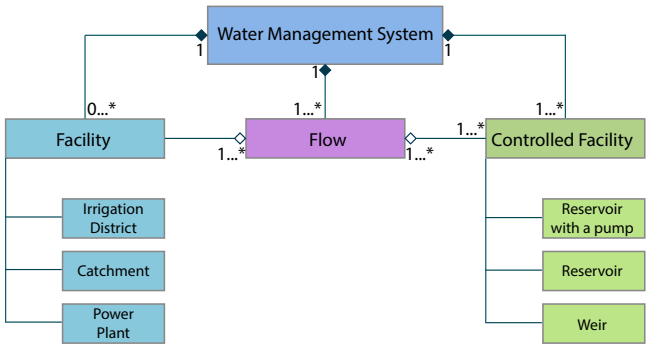


Figure 2: High-level class diagram of MORL4Water toolkit.

MORL4Water can be installed using `pip install morl4water`. Each class, as well as the example environment, can then be loaded and executed, as illustrated in the code snippet in Figure 3.

```
import morl4water.examples
from morl4water.core.envs import water_management_system
from morl4water.core.models import reservoir

env = mo_gymnasium.make('susquehanna-v0')

obs, info = env.reset()
for _ in range(1000):
    action = policy(obs)
    obs, rews, term, trunc, info = env.step(action)
```

Figure 3: Example code showing the loading of Reservoir class and running a policy in a WMS.

3.3.1 Water Management System. To set up the `WaterManagementSystem` class, we need to specify its components (from upstream to downstream), the simulation start date, and the time-step size (how often the actions are taken).

- Nile: The starting date is January 1, 2025, and decisions are made monthly over of 20 years. This yields 240 time-steps per episode. The number of time-steps can be fewer if the episode terminates early, e.g., when the water level in a facility drops below 0 or exceeds its maximum capacity.
- Susquehanna: The decisions are made every 4 hours over the course of a year, starting on January 1, 2021. This start date was selected to align with the original simulation [43], where the first day of the year, a Friday, played a critical role because release decisions are influenced by the day of the week. The total number of steps per episode is 2190, but the episode may end earlier depending on the water levels in the facilities.

Actions. Actions are represented as tuples, where the dimension of the tuple refers to the number of reservoirs integrated into the WMS or the number of outflows from each reservoir. Each component of the action tuple is represented as a percentage of water to be released, constrained by the maximum release limit, and is a continuous value within the range $[0, 1]$.

- Nile: The action tuple is four-dimensional, where each dimension represents the percentage of water to be released from one of the following dams: GERD, Roseires, Sennar, and HAD.
- Susquehanna: The release decisions at Conowingo dam are divided into four directions: Baltimore, Chester, the atomic plant, and downstream (for hydroelectricity and further flow of the Susquehanna). Thus, the action tuple for the Susquehanna environment is four-dimensional, where each dimension corresponds to the percentage of water released to each destination.

States. The state is represented as the status of each reservoir in the system with a normalized timestamp and is structured as a tuple, where the dimension of the tuple corresponds to the number of reservoirs plus one. Each component of the tuple takes a continuous value between $[0, 1]$, representing the percentage of the reservoir's storage relative to its maximum capacity. The normalized timestamp (also taking values between $[0, 1]$) is included to capture temporal information, such as the month, hour, or day of the year, allowing the system to account for seasonal and diurnal variations.

- Nile: The state is a 5-dimensional vector, with the components representing the storage of the 4 reservoirs modeled and the month of the year the system is in (normalized).
- Susquehanna: The state is a 2-dimensional vector. The first dimension represents the relative water level in the Conowingo Dam, expressed as a percentage of the maximum water level in the reservoir. The second dimension corresponds to the hour of the day (normalized). This state representation is taken from the original simulation by [12].

Rewards (Objectives). The environment returns a reward vector, whose dimension is determined by the number of objectives. To compute rewards, our toolkit implements various objective functions, reflecting the diversity of objectives commonly found in the literature for different systems. Rewards can be defined for each component of the WMS. For instance, if the goal is to maximize hydropower production, the objective can be specified in the `PowerPlant` class. The toolkit offers different reward functions depending on the goal. Rewards are also normalized by providing the maximum value during facility setup.

- Nile: the reward is four-dimensional (see Table 1
- Susquehanna: the reward is six-dimensional (see Table 2

In MORL, all objectives must be framed as maximization problems. For objectives that need to be minimized, this is achieved by converting their rewards to negative values, effectively transforming them into maximization tasks. Further, since the objective values may be on different scales (e.g., demand deficit vs. number of months), they are normalized to $[-1, 1]$.

3.3.2 Facility and Controlled Facility. The `Facility` and `ControlledFacility` classes represent components that influence the water system and generate rewards. While `Facility` models static elements that passively respond to the environment, `ControlledFacility` allows the agent to take actions that actively affect the system. Our toolkit currently includes three facility types (`Catchment`, `DemandDistrict`, and `PowerPlant`) and two controlled facility types (`Reservoir` and `ReservoirWithPump`). Detailed descriptions are provided in Appendix A[2].

4 EXPERIMENTAL SETUP

We benchmark three MORL algorithms from the MORL-baselines repository [9] on the two WMS case studies built on our MORL4Water toolkit[4]. As baseline we consider EMODPS [12], an algorithm used for approximately a decade in the water management domain for multi-objective water resource planning. The three MORL algorithms were chosen specifically as their current implementation in MORL-Baselines can handle continuous actions and state space as well as more than two objectives. Brief description of the algorithms as well as performance metrics used for evaluation can be found in Subsections 4.1 and 4.2, respectively. We trained each algorithm for 48 hours to enable a fair comparison, given that MORL algorithms are typically trained based on timesteps, while EMODPS is trained based on the number of function evaluations (NFEs). NFEs represent the number of objective function evaluations performed during optimization and are directly related to computational cost, especially in simulation-based methods such as EMODPS. All experiments were conducted using 5 random seeds. Training was performed on a high-performance computing (HPC) system [5], utilizing 48

[4]Benchmark results: https://wandb.ai/osikaz/MORL4Water_training?nw=nwuserosikazuzia

CPU cores, 48 logical processors, and a Tesla V100S-PCIE-32GB GPU. The hyperparameter tuning for the MORL algorithms was conducted using random method, the hyperparameter set-up for each algorithm can be found in Tables 5 and 6 in Appendix B[2].

4.1 Algorithms

We benchmark three multi-policy MORL algorithms, covering both CH and PF-returning methods, and use EMODPS as a baseline.

GPI-LS (Generalised Policy Improvement - Linear Support) [1] is a multi-policy MORL approach that applies the GPI [2] principle (i.e., technique to construct a guaranteed improved policy from a set of existing ones) to multi-objective action-value functions. GPI-LS assumes the utility function as linear and returns a convex hull [15]. GPI-LS learns the convex hull by learning a set of policies at the vertices of the convex hull.

PCN (Pareto Conditioned Network) [26] learns the PF in deterministic MOMDP settings. It trains a single neural network to represent all non-dominated policies. PCN operates by learning to predict the remaining return until the end of the episode, for any state, and selecting the action most likely to achieve it. This transforms the task into a supervised learning problem, avoiding the instability of moving targets. Additionally, using a single network allows PCN to scale efficiently.

CAPQL (Concave-Augmented Pareto Q-learning) [21] is a multi-policy MORL approach that assumes linear utility functions, but proposes to augment the immediate scalarised reward with a concave term, the entropy of the policy. This overcomes the limitations of linear utility and return the PF. CAPQL follows the soft actor critic framework [14], with the newly proposed learning objective, and by additionally conditioning the policy and Q-networks on the linear weights.

EMODPS. (Evolutionary Multi-Objective Direct Policy Search) is a simulation-based optimization framework for designing control policies in complex water management systems. It combines direct policy search, global function approximators (e.g., radial basis functions), and multi-objective evolutionary algorithms to address challenges such as high dimensionality, nonlinear dynamics, and competing objectives [11, 28]. By directly searching the policy space and avoiding discretization, EMODPS efficiently explores Pareto-optimal policies. To optimize policy parameters, we use ε-NSGAII [18], which maintains solution diversity through ε-dominance and adaptive population sizing.

4.2 Performance Metrics

Unlike single-objective RL, where one optimal solution exists, MORL generates a set of solutions, making evaluation challenging. We employ four widely recognized metrics for evaluation.

Hypervolume (↑) [45] represents the region or (hyper-)volume between the points in the solution set and a reference point. The reference point indicates the lower bound for each objective. Solutions with high hypervolume values are preferred. A solution set can be assessed by comparing its hypervolume with that of competing algorithms or the true Pareto front, if known.

Cardinality (↗) indicates the size of the solution set. Solution sets with higher cardinality offer decision-makers more options to choose from. However, a very high cardinality without proper diversity (i.e., if many solutions are clustered closely together) may not be beneficial. Therefore, cardinality is typically assessed with other metrics, such as sparsity, to ensure that the set of solutions is both large and well-distributed.

Sparsity (↓) [40] refers to how spread out or clustered the solutions are in a solution set. Ideally, a well-distributed solution set should cover the objective space evenly, providing decision-makers with diverse trade-offs between objectives indicating low sparsity.

Expected Utility Metric (↑) [44] When the utility function is linear, the expected utility over a distribution of reward weights can be assessed using the Expected Utility Metric (EUM). The EUM quantifies the average utility that a user would obtain from a given set of solutions, assuming a prior distribution over user utility functions.

While other metrics exist, they often rely on a known Pareto front. In real-world problems, with continuous action and state spaces, obtaining a reference Pareto front is, typically, not possible.

5 RESULTS

As shown in Figure 4, EMODPS achieves favorable convergence in mean hypervolume in both environments, indicating that solution set quality improves over time.

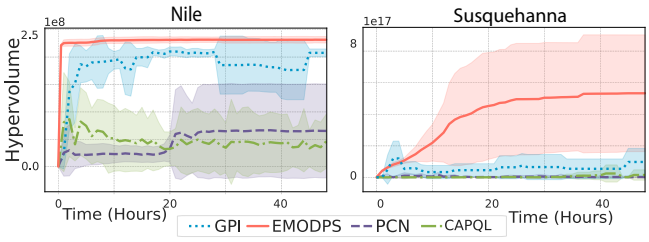


Figure 4: Hypervolume over time

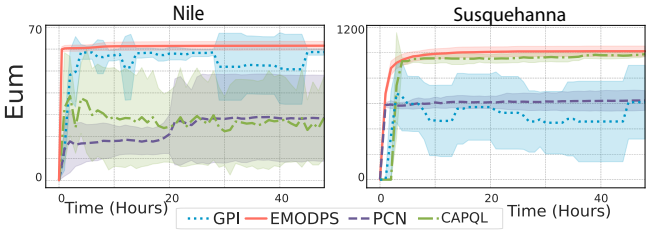


Figure 5: Expected Utility over time

In the Nile environment, EMODPS reaches high performance within the first hour and then stabilizes, with only minor gains in the remaining 47 hours. The large hypervolume values are expected due to its exponential scaling with the number of objectives (in the order of $n_{\text{timesteps}}^{n_{\text{obj}}}$). Among MORL algorithms, GPI-LS matches EMODPS in hypervolume within the first few hours, underscoring the importance of efficient prioritization when building Pareto

fronts and selecting experiences. PCN shows improvement only after 20 hours, and CAPQL fails to converge, possibly due to its reliance on stochastic policies. Hypervolume and expected utility trends (Figure 5) align closely for Nile: hypervolume grows by expanding coverage, and expected utility improves as new solutions better match user preferences. EMODPS demonstrates the most stable performance across seeds, as shown by the shaded area on the graph which represents standard deviation.

For Susquehanna, EMODPS exhibits significantly higher instability in hypervolume compared to the other MORL algorithms. Nevertheless, none of the MORL algorithms match EMODPS in terms of overall performance. Interestingly, when comparing the EUM, CapQL achieves results similar to those of EMODPS, despite performing poorly in hypervolume. This discrepancy may be attributed to CAPQL's potentially biased utility, where one objective is heavily favored over others, causing hypervolume improvements to no longer correlate with actual utility gains. Similarly, as shown in Table 3, the average sparsity of both algorithms is comparable; however, the significantly lower cardinality observed in CAPQL indicates that it focuses exploration on specific regions of the solution set that yield higher expected utility values. This highlights the importance of analyzing multiple performance metrics, as they emphasize different aspects of solution quality. Hypervolume remains difficult to interpret, with large gains sometimes resulting from irrelevant extreme solutions rather than meaningful improvements. In contrast, metrics like expected utility, which directly assess an agent's ability to maximize user satisfaction, provide more insights into algorithms' training.

	Algorithm	Hypervolume (% Baseline) ↑	Expected Utility ↑	Cardinality ↑	Sparsity ↓
Nile	EMODPS (Baseline)	2.3E+08 (100%)	61.6	845	1.2
	PCN	6.5E+07 (28%)	28.4	52	60.6
	GPI-LS	2.1E+08 (90%)	58.7	28	568.7
	CapQL	4.2E+07 (20%)	28.5	31	262.1
Susquehn.	EMODPS (Baseline)	5.33E+17 (100%)	1012.15	969	1671.6
	PCN	1.3E+15 (0.2%)	623.7	58	8159.2
	GPI-LS	9.74E+16 (18%)	609.9	90	15315.3
	CapQL	1.5E+16 (3%)	989.3	68	1672.6

Table 3: Comparison of MORL algorithms and EMODPS.

Table 3 summarizes the final mean performance metrics[5]. For MORL algorithms, hypervolume is reported as a percentage relative to the EMODPS baseline. Overall, MORL algorithms underperform compared to EMODPS, which is tailored for water management problems. An exception is GPI-LS on the 4-objective Nile case, achieving comparable hypervolume and EUM despite assuming a linear utility function and exploring only the convex Pareto front, while EMODPS covers the full front with more solutions. In the 6-objective Susquehanna case, the MORL algorithms significantly underperform, highlighting their poor scalability. Notably, this is the first complex 6-objective benchmark; previous testing for 6-objective problems was limited to simple grid world problems (e.g. [41]). Regarding cardinality, EMODPS produces substantially more solutions for both environments. For Nile, GPI-LS has fewer solutions and higher sparsity than PCN. Despite fewer solutions, GPI-LS covers more diverse and spread-out trade-offs, as indicated by its high hypervolume. In contrast, PCN, which finds more solutions, may focus on specific regions of the objective space. CAPQL exhibits lower sparsity compared to GPI-LS for both cases. This can indicate that CapQL's solution sets may be concentrated in specific regions of the objective space.

[5]Due to space constraints, we report standard deviations of the results in the Appendix B[2].

6 SOLUTION SET ANALYSIS

We analyze solution sets using parallel coordinate plots [23], where objectives are on the x-axis and normalized values (higher is better) on the y-axis. Each polyline represents a solution, with an ideal line at 1.0 across all objectives. Non-dominated solutions from multiple seeds are merged and Pareto-filtered, with the final number of non-dominated solutions indicated above each plot.

Figure 6 shows merged solution sets for the Nile WMS (EH: Ethiopia hydropower, SD: Sudan deficit, ED: Egypt deficit, HAD: Egypt Min HAD). EMODPS exhibits consistent exploration across runs, with similar merged and per-seed solution sizes (Table 3). In contrast, MORL algorithms produce much larger merged sets, indicating that different seeds explore distinct objective-space regions. CAPQL underperforms because its solutions cluster at extremes, while PCN struggles on the HAD objective. GPI-LS closely matches EMODPS trade-offs but with sparser solutions, achieving similar coverage with fewer policies, which may be advantageous in real-world settings as it can reduce cognitive load for decision-makers (compared to 796 policies from EMODPS).

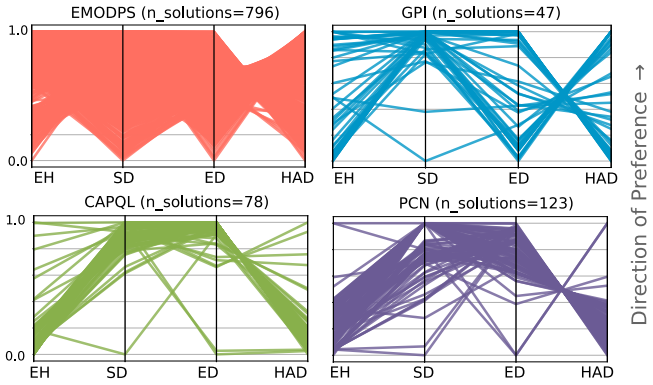


Figure 6: Parallel coordinate plots for the Nile WMS.

Figure 7 shows merged solution sets for the Susquehanna WMS (Rec: recreation, ER: hydropower revenue, Bal: Baltimore supply, Ato: atomic plant, Che: Chester, Env: environmental shortage). Merged sets are larger than per-seed averages—about twice for EMODPS and GPI, and four times for PCN and CAPQL—highlighting the challenge of consistent exploration as objectives increase. PCN and CAPQL identify few high-quality solutions for some objectives (Rec/Env for PCN, Rec for CAPQL), which explains their poor performance. CAPQL performs well on all objectives except REC, which explains its high expected-utility metric and a lower hypervolume. GPI-LS resembles EMODPS in trade-off balance but fails

to explore high values for ER, Bal, Ato, and Che, yielding lower performance than on the Nile.

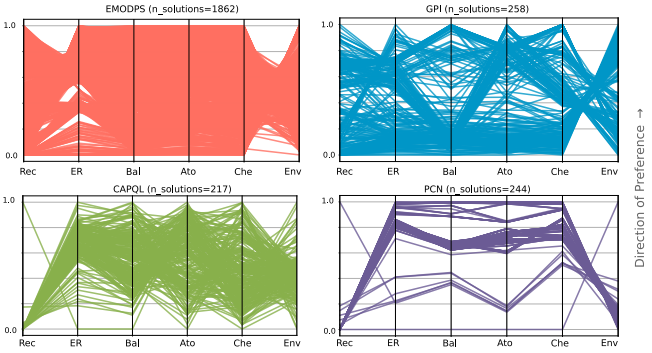


Figure 7: Parallel coordinate plots for the Susquehanna WMS.

Combining solutions from multiple seeds reveals fuller algorithm behavior and limitations in consistent exploration. Across both Nile and Susquehanna, patterns are clear: PCN shows weak coverage, CAPQL explores narrow trade-offs, and GPI-LS balances objectives well, approaching EMODPS's exploration with fewer solutions.

We study the distribution of solutions across different random seeds further to explore insights on the exploration capabilities of MORL algorithms versus EMODPS. Figure 8 presents the solution sets for each algorithm across two selected seeds (showing all seeds would lead to excessive visual clutter) for the Nile case (a similar analysis for the Susquehanna environment is in Appendix C[2]). EMODPS and GPI-LS show consistent solution distributions across seeds, with only minor differences. CapQL displays moderate variation, while PCN exhibits significant divergence, with seeds exploring distinct and opposing regions of the trade-off space.

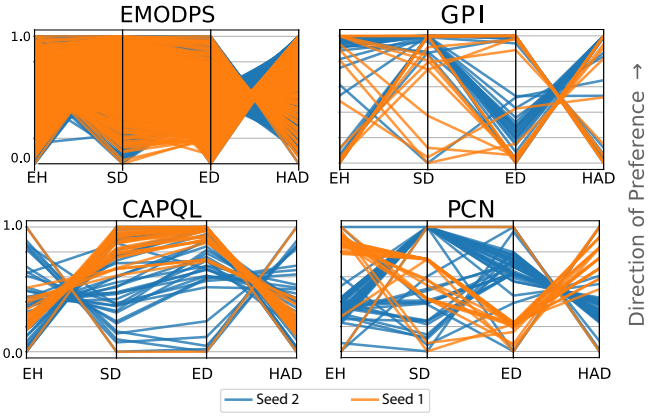


Figure 8: Solution sets (Nile WMS) for two seeds.

To quantify the differences between solution sets produced by different seeds, we calculate the pairwise average minimal distance between each set of solutions per seed. That is, for each pair of seeds, we measure the average distance from each point in one set to its nearest neighbor in the other set, using the Euclidean distance. Then, we report the mean of these distances in Table 4.

These distances indicate the average spread between the set of solutions produced by different seeds for each algorithm. A lower mean distance indicates that the solutions are more tightly grouped between the seeds, reflecting greater consistency. Among the algorithms, EMODPS achieves the lowest mean distance (0.1275), followed by GPI-LS (0.1388), suggesting higher robustness across runs. In contrast, PCN exhibits the highest mean distance (0.2527), indicating greater variability between seeds.

Algorithm	Mean (Std. Dev.) Distance
CapQL	0.1950 (± 0.07)
PCN	0.2527 (± 0.10)
GPI-LS	0.1388 (± 0.03)
EMODPS	0.1275 (± 0.06)

Table 4: Distance between per-seed solution sets (Nile WMS).

7 CONCLUSIONS

Our work highlights the importance of grounding MORL research in realistic, high-impact application domains. Although there is an increasing number of MORL algorithms, most evaluations rely on abstract benchmarks with limited practical relevance. To address this limitation, we developed MORL4Water, a reusable toolkit for studying MORL in complex water management tasks. Further, we demonstrated the complexity of evaluating MORL methods through quantitative measures as well as qualitative solution-set analysis.

Using case studies on the Nile and Susquehanna river systems, we revealed both the promise and limitations of state-of-the-art algorithms. On the four-objective Nile case, GPI-LS matched the performance of EMODPS while offering fewer solutions, reducing decision complexity without sacrificing trade-off coverage; PCN and CapQL failed to achieve consistent or broad Pareto coverage. In contrast, all MORL methods underperformed EMODPS on the six-objective Susquehanna case, underscoring scalability challenges. Thus, although current MORL algorithms may not yet replace specialized approaches, our results provide valuable insights and a foundation for improving and innovating MORL methods.

The two case studies demonstrate the feasibility and usefulness of our framework, but they are not sufficient to generalize our findings across all water management contexts. However, building detailed real-world water system models is a time-consuming task, and it requires both extensive domain knowledge and access to high-quality data. Our contribution lies in streamlining this process through a reusable pipeline, which lowers the barrier for developing further simulations and also enables future analyses of how policies may shift under scenarios such as climate change.

Finally, our benchmarking covers only a subset of methods from the MORL-baselines framework [9], focusing on established representatives to ensure consistent implementation—a critical consideration given RL's sensitivity to implementation details [7]. Our goal is not to provide a definitive ranking, but to illustrate realistic benchmarking, highlight challenges like scalability and exploration, and show the value of analyzing full solution sets rather than relying solely on scalar metrics.

REFERENCES

[1] Lucas N. Alegre, Ana L. C. Bazzan, Diederik M. Roijers, Ann Nowé, and Bruno C. da Silva. 2023. Sample-Efficient Multi-Objective Learning via Generalized Policy Improvement Prioritization. In *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems* (London, United Kingdom) *(AAMAS '23)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 2003–2012.

[2] André Barreto, Shaobo Hou, Diana Borsa, David Silver, and Doina Precup. 2020. Fast reinforcement learning with generalized policy updates. *Proceedings of the National Academy of Sciences* 117, 48 (2020), 30079–30087.

[3] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. 2012. Tree-based Fitted Q-iteration for Multi-Objective Markov Decision problems. In *The 2012 International Joint Conference on Neural Networks (IJCNN)*. 1–8. https://doi.org/10.1109/IJCNN.2012.6252759

[4] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. 2012. Tree-based fitted Q-iteration for multi-objective Markov decision problems. In *The 2012 international joint conference on neural networks (IJCNN)*. IEEE, 1–8.

[5] Delft High Performance Computing Centre (DHPC). 2024. DelftBlue Supercomputer (Phase 2). https://www.tudelft.nl/dhpc/ark:/44463/DelftBluePhase2.

[6] Emad Elba, Brigitte Urban, Bernd Ettmer, Dalia Farghaly, et al. 2017. Mitigating the impact of climate change by reducing evaporation losses: sediment removal from the High Aswan Dam reservoir. *American Journal of Climate Change* 6, 02 (2017), 230.

[7] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Firdaus Janoos, Larry Rudolph, and Aleksander Mądry. 2019. Implementation Matters in Deep RL: A Case Study on PPO and TRPO. In *International Conference on Learning Representations (ICLR)*. https://openreview.net/forum?id=r1etN1rtPB

[8] Florian Felten, Lucas N. Alegre, Ann Nowe, Ana Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. 2023. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (Eds.), Vol. 36. Curran Associates, Inc., 23671–23700. https://proceedings.neurips.cc/paper_files/paper/2023/file/4aa8891583f07ae200ba07843954caeb-Paper-Datasets_and_Benchmarks.pdf

[9] Florian Felten, Lucas N. Alegre, Ann Nowé, Ana L. C. Bazzan, El Ghazali Talbi, Grégoire Danoy, and Bruno C. da Silva. 2023. A Toolkit for Reliable Benchmarking and Research in Multi-Objective Reinforcement Learning. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS 2023)*.

[10] Zoltán Gábor, Zsolt Kalmár, and Csaba Szepesvári. 1998. Multi-criteria reinforcement learning.. In *ICML*, Vol. 98. 197–205.

[11] Matteo Giuliani, Andrea Castelletti, Francesca Pianosi, Emanuele Mason, and Patrick M Reed. 2016. Curses, tradeoffs, and scalable management: Advancing evolutionary multiobjective direct policy search to improve water reservoir operations. *Journal of Water Resources Planning and Management* 142, 2 (2016), 04015050.

[12] Matteo Giuliani, Jonathan D Herman, Andrea Castelletti, and P Reed. 2014. Many-objective reservoir policy identification and refinement to reduce policy inertia and myopia in water management. *Water resources research* 50, 4 (2014), 3355–3377.

[13] M Giuliani, JR Lamontagne, PM Reed, and A Castelletti. 2021. A state-of-the-art review of optimal reservoir control for managing conflicting demands in a changing world. *Water Resources Research* 57, 12 (2021), e2021WR029927.

[14] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. 2018. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *International conference on machine learning*. PMLR, 1861–1870.

[15] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, Johan Källström, Matthew Macfarlane, Mathieu Reymond, Timothy Verstraeten, Luisa M Zintgraf, Richard Dazeley, Fredrik Heintz, et al. 2022. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems* 36, 1 (2022), 26.

[16] Jonathan D Herman, Julianne D Quinn, Scott Steinschneider, Matteo Giuliani, and Sarah Fletcher. 2020. Climate adaptation as a control problem: Review and perspectives on dynamic water resources planning under uncertainty. *Water Resources Research* 56, 2 (2020), e24389.

[17] Ammar Jalalimanesh, Hamidreza Shahabi Haghighi, Abbas Ahmadi, Hossein Hejazian, and Madjid Soltani. 2017. Multi-objective optimization of radiotherapy: distributed Q-learning and agent-based simulation. *Journal of Experimental & Theoretical artificial intelligence* 29, 5 (2017), 1071–1086.

[18] Joshua B Kollat and Patrick M Reed. 2005. The value of online adaptive search: a performance comparison of NSGAII, ϵ-NSGAII and ϵMOEA. In *International conference on evolutionary multi-criterion optimization*. Springer, 386–398.

[19] Changjian Li and Krzysztof Czarnecki. 2019. Urban Driving with Multi-Objective Deep Reinforcement Learning. In *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems* (Montreal QC, Canada) *(AAMAS '19)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 359–367.

[20] Daniel Lizotte, Michael Bowling, and Susan Murphy. 2010. Efficient Reinforcement Learning with Multiple Reward Functions for Randomized Controlled Trial Analysis. *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, 695–702.

[21] Haoye Lu, Daniel Herman, and Yaoliang Yu. 2023. Multi-Objective Reinforcement Learning: Convexity, Stationarity and Pareto Optimality. In *The Eleventh International Conference on Learning Representations*. https://openreview.net/forum?id=TjEzIsyEsQ6

[22] Youngwoo Oh, Arif Ullah, and Wooyeol Choi. 2023. Multi-Objective Reinforcement Learning for Power Allocation in Massive MIMO Networks: A Solution to Spectral and Energy Trade-Offs. *IEEE Access* (2023).

[23] Zuzanna Osika, Jazmin Zatarain Salazar, Diederik M. Roijers, Frans A. Oliehoek, and Pradeep K. Murukannaiah. 2023. What lies beyond the pareto front? a survey on decision-support methods for multi-objective optimization. In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence* (Macao, P.R.China) *(IJCAI '23)*. Article 755, 9 pages. https://doi.org/10.24963/ijcai.2023/755

[24] Afua Owusu, Jazmin Zatarain Salazar, Marloes Mul, Pieter van der Zaag, and Jill Slinger. 2022. Quantifying the trade-offs in re-operating dams for the environment in the Lower Volta River. *Hydrology and Earth System Sciences Discussions* 2022 (2022), 1–27.

[25] Julianne D Quinn, Patrick M Reed, Matteo Giuliani, and Andrea Castelletti. 2019. What is controlling our control rules? Opening the black box of multireservoir operating policies using time-varying sensitivity analysis. *Water Resources Research* 55, 7 (2019), 5962–5984.

[26] Mathieu Reymond, Eugenio Bargiacchi, and Ann Nowe. 2022. Pareto Conditioned Networks. In *The 21st International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 1110–1118. https://aamas2022-conference.auckland.ac.nz

[27] Willem Röpke, Mathieu Reymond, Patrick Mannion, Diederik M Roijers, Ann Nowé, and Roxana Rădulescu. 2025. Divide and Conquer: Provably Unveiling the Pareto Front with Multi-Objective Reinforcement Learning. In *Proceedings of the 24th International Conference on Autonomous Agents and Multiagent Systems*. 1774–1783.

[28] Jazmin Zatarain Salazar, Jan H Kwakkel, and Mark Witvliet. 2024. Evaluating the choice of radial basis functions in multiobjective optimal control applications. *Environmental Modelling & Software* 171 (2024), 105889.

[29] Jazmin Zatarain Salazar, Patrick M Reed, Jonathan D Herman, Matteo Giuliani, and Andrea Castelletti. 2016. A diagnostic assessment of evolutionary algorithms for multi-objective surface water reservoir control. *Advances in water resources* 92 (2016), 172–185.

[30] Matteo Sangiorgio and Giorgio Guariso. 2018. NN-based implicit stochastic optimization of multi-reservoir systems management. *Water* 10, 3 (2018), 303.

[31] Yasin Sari. 2022. *Exploring trade-offs in reservoir operations through many objective optimisation: Case of Nile river basin.* Master's thesis. TU Delft.

[32] R. Soncini-Sessa and E. Weber. 2007. Chapter 5 Modelling the components. In *Integrated and Participatory Water Resources Management: Theory*, R. Soncini-Sessa (Ed.). Developments in Integrated Environmental Assessment, Vol. 1. Elsevier, 137–184. https://doi.org/10.1016/S1574-101X(07)01105-2

[33] Athanasia-Tatiana Stamou and Peter Rutschmann. 2018. Pareto optimization of water resources using the nexus approach. *Water resources management* 32 (2018), 5053–5065.

[34] Jery R Stedinger, Bola F Sule, and Daniel P Loucks. 1984. Stochastic dynamic programming models for reservoir operation optimization. *Water resources research* 20, 11 (1984), 1499–1505.

[35] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press.

[36] United States Securities and Exchange Commission. 2011. Exxon Mobil Corporation - 2010 Financial & Operating Review. https://www.sec.gov/Archives/edgar/data/22606/000119312511099635/dex991.htm. Accessed: 2024-10-12.

[37] Kevin G Wheeler, Jim W Hall, Gamal M Abdo, Simon J Dadson, Joseph R Kasprzyk, Rebecca Smith, and Edith A Zagona. 2018. Exploring cooperative transboundary river management strategies for the Eastern Nile Basin. *Water Resources Research* 54, 11 (2018), 9224–9254.

[38] Jiahao Wu, Yang Ye, and Jing Du. 2024. Multi-objective reinforcement learning for autonomous drone navigation in urban areas with wind zones. *Automation in Construction* 158 (2024), 105253.

[39] Zheng Xiong, Biao Luo, Bing-Chuan Wang, Xiaodong Xu, and Tingwen Huang. 2023. Multi-Objective Battery Charging Strategy Based on Deep Reinforcement Learning. *IEEE Transactions on Transportation Electrification* (2023).

[40] Jie Xu, Yunsheng Tian, Pingchuan Ma, Daniela Rus, Shinjiro Sueda, and Wojciech Matusik. 2020. Prediction-guided multi-objective reinforcement learning for continuous robot control. In *International conference on machine learning*. PMLR, 10607–10616.

[41] Runzhe Yang, Xingyuan Sun, and Karthik Narasimhan. 2019. A generalized algorithm for multi-objective reinforcement learning and policy adaptation. *Advances in neural information processing systems* 32 (2019).

[42] Marta Zaniolo, Matteo Giuliani, and Andrea Castelletti. 2021. Neuro-evolutionary direct policy search for multiobjective optimal control. *IEEE Transactions on*

Neural Networks and Learning Systems 33, 10 (2021), 5926–5938.

[43] Jazmin Zatarain Salazar, Jan H. Kwakkel, and Mark Witvliet. 2024. Evaluating the choice of radial basis functions in multiobjective optimal control applications. *Environmental Modelling & Software* 171 (2024), 105889. https://doi.org/10.1016/j.envsoft.2023.105889

[44] Luisa M Zintgraf, Timon V Kanters, Diederik M Roijers, Frans Oliehoek, and Philipp Beau. 2015. Quality assessment of MORL algorithms: A utility-based approach. In *Benelearn 2015: proceedings of the 24th annual machine learning conference of Belgium and the Netherlands*.

[45] Eckart Zitzler, Lothar Thiele, Marco Laumanns, Carlos M Fonseca, and Viviane Grunert Da Fonseca. 2003. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on evolutionary computation* 7, 2 (2003), 117–132.