Technion - Israel Institute of Technology
Faculty of Computer Science

Multi-Objective Approaches in EnergyNet

Students:	Almog Anschel	Rotem Shezaf
IDs:	313123456	314789012
Course:	CLAIR LAB projects	246XXXX
Instructor:	Mr. Itay Segal	Dr. Serah Keren
Semester:	Spring 2025	
Date:	September 4, 2025	

TECHNION

1 Introduction

In standard reinforcement learning, the goal is to find an optimal policy that maximizes the expected return. However, many real-world problems require dealing with multiple conflicting objectives. For example, wind turbines need to maximize power output, but such maximization may lead to larger stress on the turbine components [4]. Each solution creates a trade-off between objectives. To address such problems, we are interested in finding a set of trade-off solutions that represent the multiple objectives [7].

The standard method to address these challenges is to use multi-objective reinforcement learning (MORL) algorithms. In this project, we examined two approaches to MORL algorithms: the utility-based approach and the multi-policy approach, applied to the EnergyNet environment. To represent the utility-based approach, we constructed a multi-objective version of the Soft Actor-Critic (SAC) algorithm (MOSAC). To represent the multi-policy approach, we chose the Hyper-Morl algorithm. The Hyper-Morl algorithm approximates the pareto front by using a hyper-network. We adjust the hyer-morl algorithm to work with 4 objectives. We explored two architectures for the MOSAC algorithm critic network, shared featured network and separate featured network.

The conflicting objectives we selected for the EnergyNet system are:

- Economic profit (energy arbitrage)
- Battery health and lifetime
- Grid support/stability
- Energy autonomy

We found that the Hyper-Morl algorithm converges by the hyper volume metric and reach hyper-volume value off 758.65. We found that the mosac algorithm converges faster than the Hyper-Morl algorithm for the four objective problem, theretofore is preferable when the preference vector is known. Furthermore, we found that the MOSAC algorithm is preferable over the baslines single-objective RL algorithms for the four objective problem, and that the shared-fetures architecture is preferable over the saperated-fetured one for the four objective. Alse, we found, the single objective RL algorithms achieves worst results as the number of objectives increase, and cannot find a good trade off for the four objectives problem.

2 backround

2.1 Multi-Objective Reinforcement Learning

Multi-objective problems are typically modeled as Multi-Objective Markov Decision Processes (MOMDPs) [4], defined by a tuple $\langle S, A, T, \gamma, \mu, \mathbf{R} \rangle$, where:

- S is the state space
- A is the action space
- $T : S \times A \times S \to [0, 1]$ is the probabilistic transition function
- $\gamma \in [0, 1]^d$ is the discount factor vector
- $\mu : S \to [0, 1]$ is the probability distribution over initial states

- $\mathbf{R} : S \times A \times S \to \mathbb{R}^d$ is the multi-dimensional reward function

An MDP is a special case of MOMDP when $d = 1$ [5]. In multi-objective RL, the policy is represented as π_θ where $\theta \in \Theta$ (the policy parameter space) and $\pi_\theta \in \Pi$ (the policy space).

The multi-objective value function is defined as:

$$\mathbf{V}^\pi = \mathbb{E}\left[\sum_{k=0}^{\infty} \gamma_i^k \cdot r_i(s_k, a_k, s_{k+1}) \mid \pi, \mu\right] \tag{1}$$

2.2 Pareto Optimality

2.2.1 Pareto dominance

A policy $\pi \in \Pi$ is Pareto dominate policy $\pi' \in \Pi$ if

$$(\forall i\colon\ V_i^\pi \geq V_i^{\pi'}) \text{ and } (\exists i\colon\ V_i^\pi > V_i^{\pi'})$$

2.2.2 pareto optimality

a policy $\pi^* \in \Pi$ is Pareto optimal if it is not dominated by any other policy $\pi \in \Pi$

2.2.3 pareto front

The Pareto front is The set of all the Pareto optimal policies.

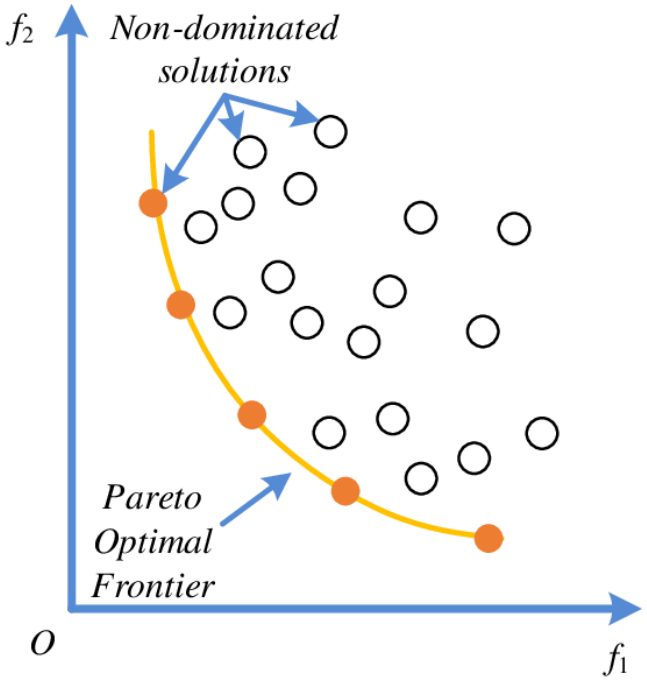

Figure 1: Illustration of Pareto optimality concept showing dominated and non-dominated solutions

2.2.4 hypervolume metric

the hypervolume metric is a metric to value the accuracy of a set of policies as a multy policy solution for a MORL problem. The hyper volume of an approximated set of policies defined as the hyper volume of all the policies that Pareto dominated by a policy in the set in the value function space [4]. The hyper-volume metric is maximized at the pareto front [4]. Therfure, the hyper-volume metric is a standard method to measure the efficiency of a multy policy MORL algorithm.

2.3 The Utility-Based Approach

The simplest approach to multi-objective RL is the utility-based approach [4]. These approaches use a scalarisation function on the value function or Q-function to adapt single-policy algorithms to multi-objective problems.

The main limitations of this approach are:

1. When the utility function is non-linear, the Bellman equation doesn't hold, therfur, there's no garentee for convergence [2]
2. In most cases, the utility function is unknown [4]

Attempts to address these problems include saving the accrued reward (the cumulative reward received until the current time) in the environment state [4], and making the critic learn a multi-dimensional distribution over the reward space [2].

2.4 The Pareto Optimality Approach

When the scalarization function is unknown and objectives are conflicting, there is usually no single optimal solution [7]. Therefore, multi-policy algorithms that learn a set of policies simultaneously are needed. These algorithms learn a set of Pareto-optimal policies, aiming to find the entire Pareto front.

However, methods that learn a finite set of optimal policies, such as Pareto Q-learning [7], cannot learn a continuous Pareto front because a finite number of policies cannot represent the entire Pareto set [6]. While it's possible to create discrete approximations of the Pareto front by training multiple networks with different preferences [3], this method is memory-intensive and provides poor representation when user preferences differ from the learned set [6].

The solution we choose is to use a single network to represent the Pareto front. There are two approaches to approximate the Pareto front with a single network [6]:

1. **Embedding methods**: Add the preferences as input to the network, as shown in the PD-MORL algorithm [1]
2. **Hypernetwork approach**: Use a hypernetwork that determines policy network parameters based on preferences, as shown in Hyper-Morl [6]

The Pareto Set Learning (PSL) algorithm combines both approaches using parameter fusion techniques [5]. The hypernetwork approach experimentally performs better when dealing with a large number of objectives and policy parameters [6], while PD-MORL has theoretical analysis and convergence proofs. [1].

2.5 Hyper-Morl algorithm

An hyper-network is a neural network that generates the parameters of another neural network. Hyper-Morl algorithm use hyper-network from the preference space to the parameter space of the policy network. Hyper-Morl algorithm trains a hypernet that find $\omega \in \Omega$ for each $\theta \in \Theta$ such that $\omega^T J(\theta)$ is maximized [6]. $J(\theta)$ is the policy network, Θ it the policy network parameters space and Ω is the preferance space. Each Optomization step of $\omega^T J(\theta)$ is based on a single objective RL algorithm. The algorithm consist of two stages [6]:

1. **warm up stage**: train the model on a uniform preference vector, $[1/d, ...1/d]$ to get initial policy close to the pareto set.

2. **pareto learning stage**: train the modle on preferance sampled with monte carlo method [6]

The hyper morl-algorithm have two assumptions [6]:

1. the Pareto Front can be approximated as a lower dimension manifold at the preference space

2. the problem is convex. If so, for each policy $J(\theta)$ ($\theta \in \Theta$) in the Pareto front exist $\omega \in \Omega$ that maximizes $\omega^T J(\theta)$. [6]

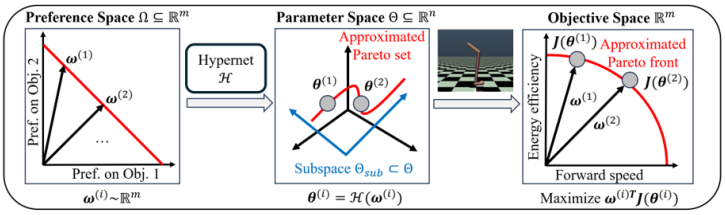

Figure 2: Illustration of the basic idea of the Hyper-Morl algorithm [6]

3 Related Work

3.1 Multi-Objective Reinforcement Learning

Multi-objective reinforcement learning extends traditional RL to handle vector-valued rewards. The field has seen significant advances in recent years, with approaches broadly categorized into utility-based methods and Pareto-based methods.

3.2 Utility-Based Approaches

Utility-based methods scalarize the multi-objective problem using a utility function. These approaches are straightforward to implement but require knowledge of preference weights a priori. Our MOSAC implementation falls into this category, using linear scalarization to combine objectives.

3.3 Pareto-Based Approaches

Pareto-based methods aim to approximate the entire Pareto front, providing a set of optimal trade-off solutions. Recent work by Shu et al. (2023) introduced hypernetwork-based approaches for learning the Pareto set in continuous control tasks. We adapted their methodology for the energy domain.

3.4 MORL in Energy Systems

While reinforcement learning has been widely applied to energy management, multi-objective approaches remain relatively unexplored in this domain. Most existing work focuses on bi-objective problems, typically balancing cost and emissions. Our work extends this to four simultaneous objectives, presenting a more realistic model of grid operation challenges.

4 Problem Formulation

4.1 The Power Control System Environment

The PCS agent operates within the EnergyNet simulation framework, managing battery storage systems in response to grid conditions and price signals. The agent must decide when to charge, discharge, or maintain the battery state while considering multiple performance metrics.

4.2 Multi-Objective Framework

We formulate the problem as a multi-objective Markov Decision Process (MOMDP) with four reward components:

1. **Economic Profit (r_1)**: Revenue from energy arbitrage
2. **Battery Health (r_2)**: Penalty for actions causing battery degradation
3. **Grid Support (r_3)**: Rewards for providing grid stability services
4. **Energy Autonomy (r_4)**: Maintaining energy self-sufficiency

The agent receives a reward vector $\mathbf{r} = [r_1, r_2, r_3, r_4]$ at each timestep instead of a scalar reward.

The challenge lies in approximating the Pareto front of that MOMDP efficiently and providing mechanisms for policy selection based on user preferences.

5 Methodology

Our assumption was that the EnergyNet system would benefit from multi-objective RL because we can extract conflicting objectives from the environment. We chose the following four objectives:

1. Economic profit (energy arbitrage)
2. Battery health and lifetime
3. Grid support/stability
4. Energy autonomy

We hypothesized that economic profit and battery health objectives would conflict, as would grid support and energy autonomy objectives.

5.1 Initial Approach: Utility-Based Methods

Initially, we used the utility-based approach by implementing a multi-objective Soft Actor-Critic (MOSAC) algorithm working with linear utility functions. We Based our implantation on the existing implementation of the stable-baseline SAC algorithm. We experience technical difficulty in the creation of the mosac algorithm, since the stable-basline3 and rl3-zoo cannot handle multi-dimensional critic od reward. For calculating the loss, We used the scalarization function on the critic loss instead of the Q function. We considered expanding the algorithm for non-linear utility functions, But decided that the implementation of The methods we found to handle non-linear utility functions, the accrued reward and distributional learning, is to complicated for implementation.

Since the utility function for EnergyNet is unknown, we decided to search for algorithms that could approximate the Pareto front without requiring explicit preference weights.

5.2 Evolution to Pareto-Based Methods

After failed attempts to create a Pareto-MOSAC algorithm based on Pareto Q-learning principles, we determined that our best approach is to use algorithms that approximate the Pareto front using a single network. We chose this approach because the EnergyNet system is too complex for discrete approximation of the Pareto front using multiple separate policies.

We initially wanted to compare three algorithms:

- Pareto Set Learning (PSL) [5]
- PD-MORL [1]
- Hyper-Morl [6]

Unfortunately, due to time constraints and the late discovery of single-network Pareto approximation methods, we implemented only Hyper-Morl. This decision was based on:

1. Lack of existing code for the PSL algorithm
2. Literature showing PSL's experimental superiority over PD-MORL [5]
3. With four objectives and sufficient policy parameters, Hyper-Morl was well-justified for our use case

5.3 Multi-Objective SAC (MOSAC) Implementation

5.3.1 Architecture Design

MOSAC extends Soft Actor-Critic to handle vector rewards through multi-head critics. We developed two variants:

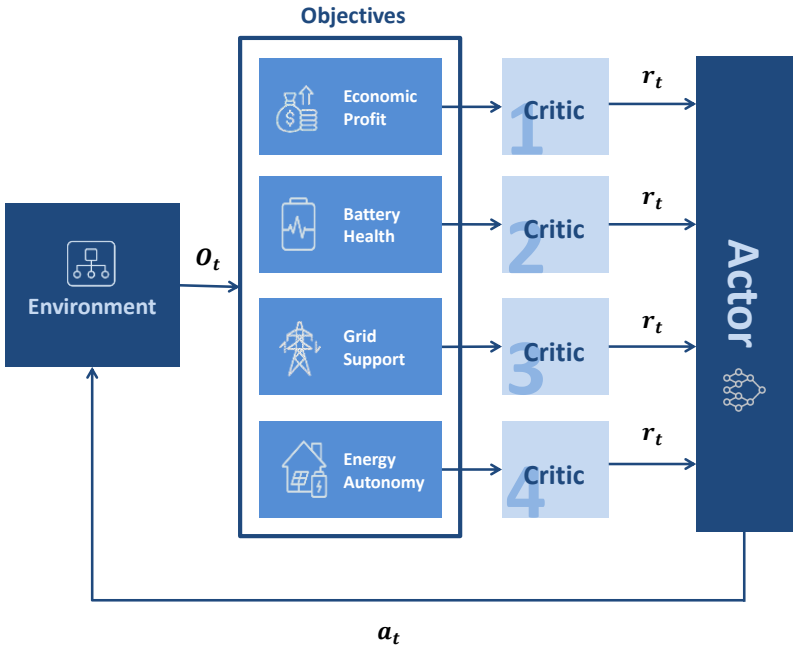

Figure 3: MOSAC architecture showing multi-head critics processing individual objectives

Variant 1: Shared Features

- Single network with multiple output heads
- Shared feature extraction layers
- Joint learning across objectives

Variant 2: Separate Critics

- Four independent critic networks
- No shared parameters
- Independent learning per objective

5.4 Hypernetwork-based MORL Implementation

We used the Hyper-Morl algoritmn with PPO as the algorithm for the single objective RL algorithm.

We adapted the existing Hyper-Morl implementation for the energy net system:

- Extended the algorithm from three to four objectives
- Adjust the rewards normalization to suit the Hyper-Morl algorithm best
- created a wrapper for matching the existing implrmitation detailed of the Hyper-Morl algorithm
- Adjusted hyperparameters for energy domain characteristics

5.5 Reward normalization

To achive the best results from the mosac algorithm and the Hyper-Morl algorithm, we used different reward normalization for each. We chose the min-max normalization for each reward.

6 Experimental Setup

6.1 Environment Configuration

All experiments were conducted using the EnergyNet simulator with the following settings:

- Episode length: 96 timesteps (representing 24 hours with 15-minute intervals)
- Battery capacity: 100 kWh
- Maximum charge/discharge rate: 50 kW
- Price signal: Real historical data from energy markets

6.2 Training Parameters

6.2.1 MOSAC Configuration

Parameter	Value
Learning rate	3e-4
Discount factor (γ)	0.99
Buffer size	1000
Batch size	64
Training frequency	1
Total timesteps	500,000

Table 1: MOSAC hyperparameters

Objective	min	max
Economic profit	-50	50
Battery health and lifetime	-2	1
Grid support/stability	-1	1
Energy autonomy	0	1

Table 2: Min Max normalization parameters for MOSAC

6.2.2 Hyper-Morl Configuration

Parameter	Value
Learning rate	5e-4
Discount factor (γ)	0.95
Total timesteps	30,000,000
Number of processes	4
Warmup steps	2048

Table 3: Hyper-Morl hyperparameters

Objective	min	max
Economic profit	-10	10
Battery health and lifetime	-5	5
Grid support/stability	-0.02	0.02
Energy autonomy	-1	1

Table 4: Min Max normalization parameters for Hyper-Morl

6.3 Evaluation Metrics

We evaluated our algorithms using:

- Scalarized reward for comparison with baselines
- Hypervolume indicator for Pareto front quality
- Individual objective performance
- Pareto front coverage and diversity

7 Results and Analysis

7.1 Baseline Comparison

We first evaluated standard RL algorithms with our scalarization wrapper to establish baselines. we examined the performns of 3 single objective RL algorithms on the problem with our scalrized wrapper: TD3, SAC and PPO. the learning curve is shown at figure 3, and the achived mean reward for the three algorithms is shown at table 3.

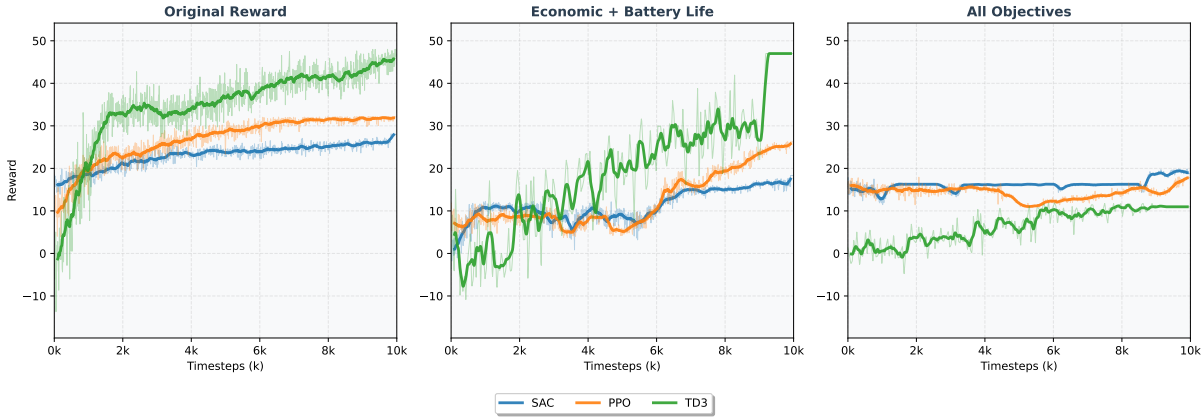

Figure 4: Performance comparison of baseline algorithms on different objective combinations

Algorithm	Original Reward	Economic + Battery	All Objectives
SAC	23.10	11.25	16.22
PPO	26.92	12.48	14.30
TD3	34.63	19.40	6.58
MOSAC (Shared)	7	18.4	**28.00**
MOSAC (Separate)	12.4	11.7	18.50

Table 5: Mean rewards over 3 random seeds

7.2 MOSAC Performance

7.2.1 Shared Critics Variant

The shared critics variant achieved the highest scalarized reward of 28 after 500,000 timesteps. The shared feature extraction appears to enable better coordination between objectives.

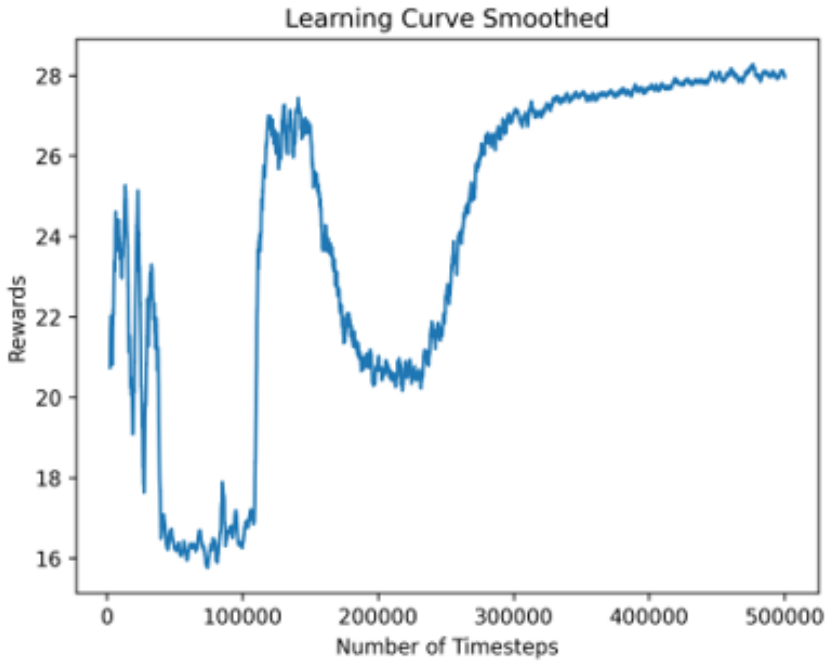

Figure 5: Learning curve for MOSAC with shared critics

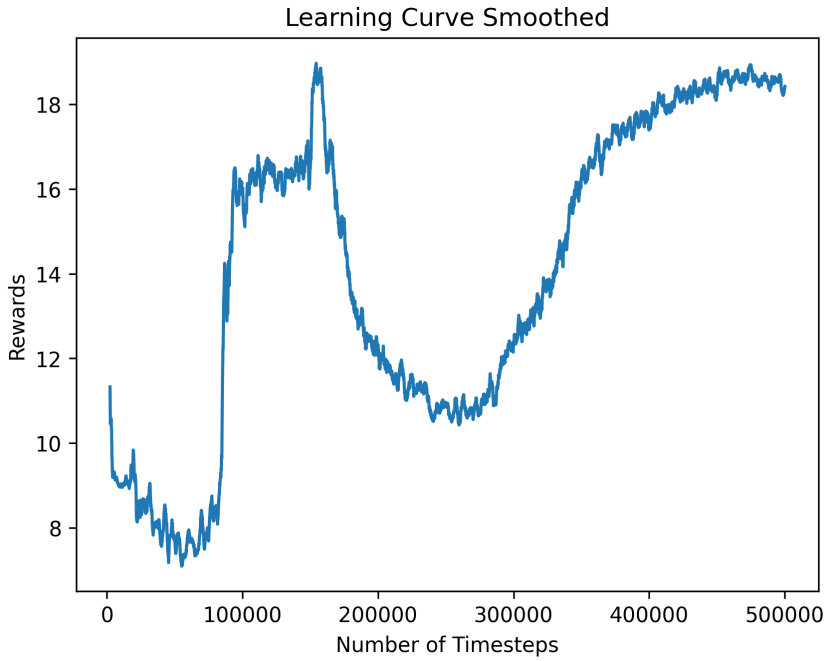

Figure 6: Learning curve for MOSAC with shared critics for Economic + Battery Life objectives

7.2.2 Separate Critics Variant

The separate critics variant converged faster (100,000 steps) but achieved lower final performance (18.5 reward). This suggests a trade-off between convergence speed and final performance.

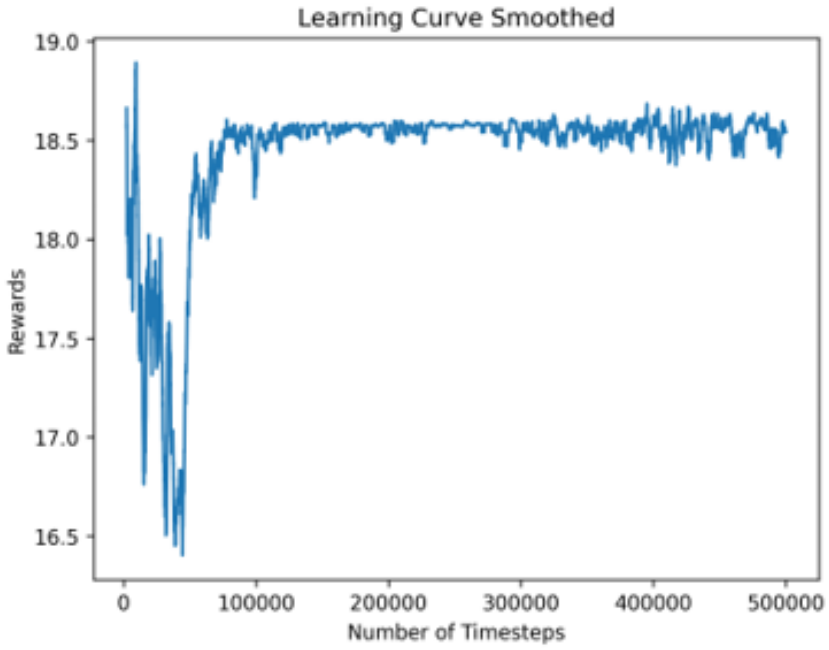

Figure 7: Learning curve for MOSAC with separate critics

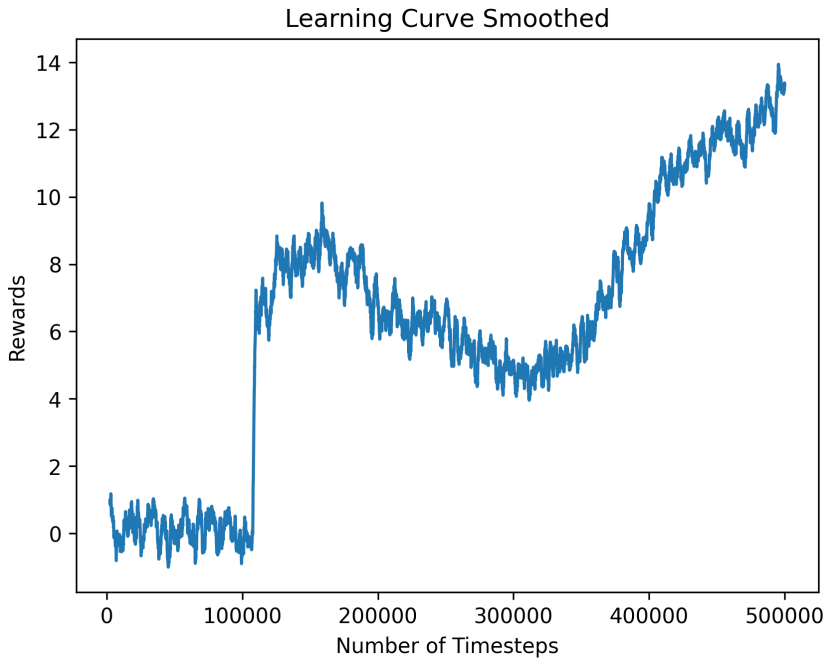

Figure 8: Learning curve for MOSAC with separate critics for Economic + Battery Life objectives

7.3 Hyper-Morl Results

7.4 Hyper-Morl Results

7.4.1 Pareto Front Approximation

The hypernetwork successfully approximated the Pareto front in the four-dimensional objective space. Figure 9 shows the projection onto the first two objectives.

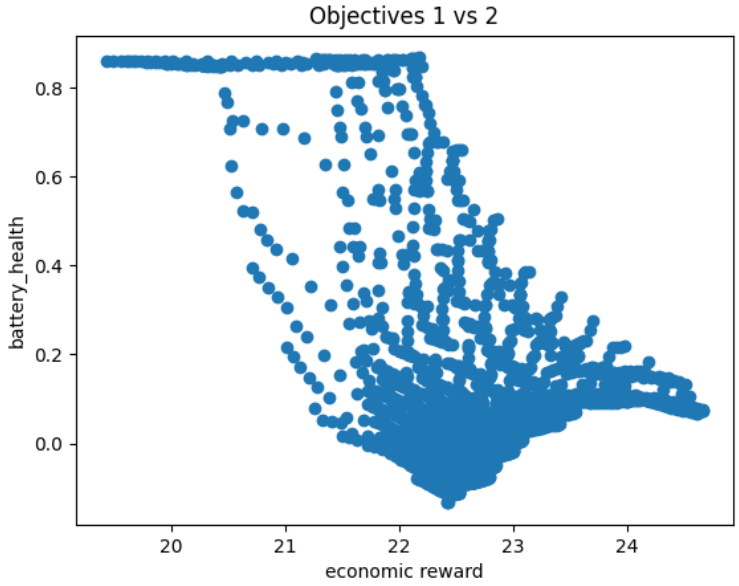

Figure 9: Pareto front projection: Economic reward vs Battery health

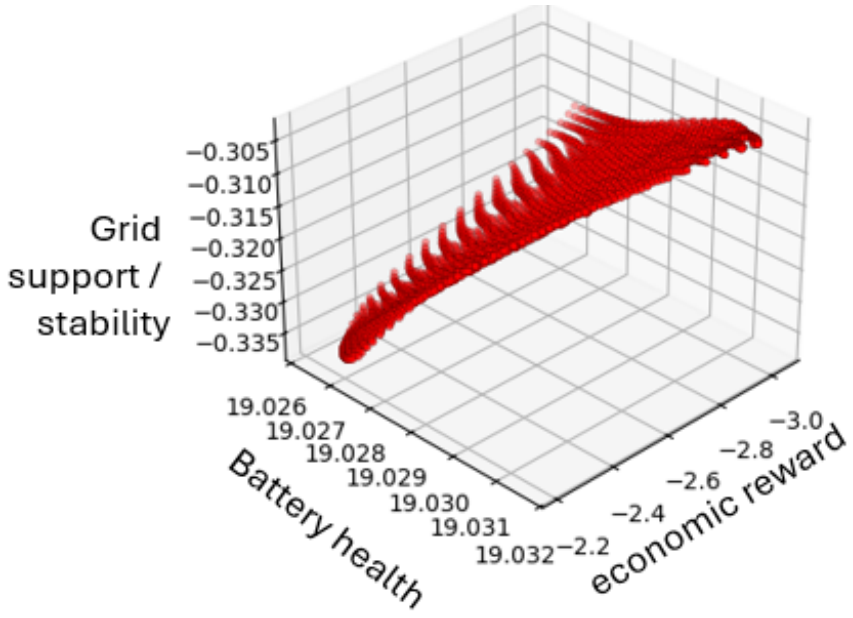

Figure 10: 3D projection of the Pareto front (first three objectives)

7.4.2 Hypervolume Analysis

Since our agent's didn't perfumed selling or buing actions, the autonomy reward was constant. Therefore, our hyper valume metric was 0. Therfure, to check the preformence of the Hyper-Morl algorithm we used the hyper volume metric on the first two objectives. The three-dimensional hypervolume (calculated on the first three objectives) converged to 758.65, indicating good coverage of the objective space.

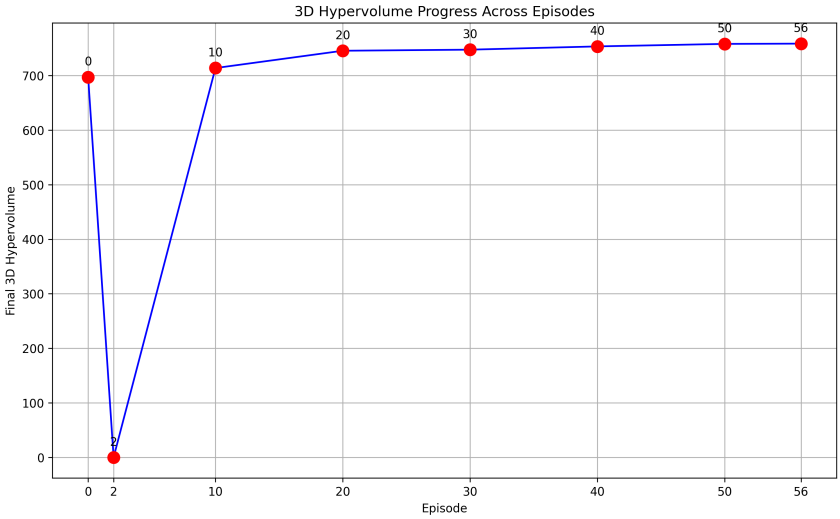

Figure 11: 3D hypervolume convergence over training episodes

7.5 Principal Component Analysis

We performed PCA analysis on the Pareto front to visualize the 4 dimensions pareto front. Further more, we used the pcs analysis to understand the relationships between objectives.

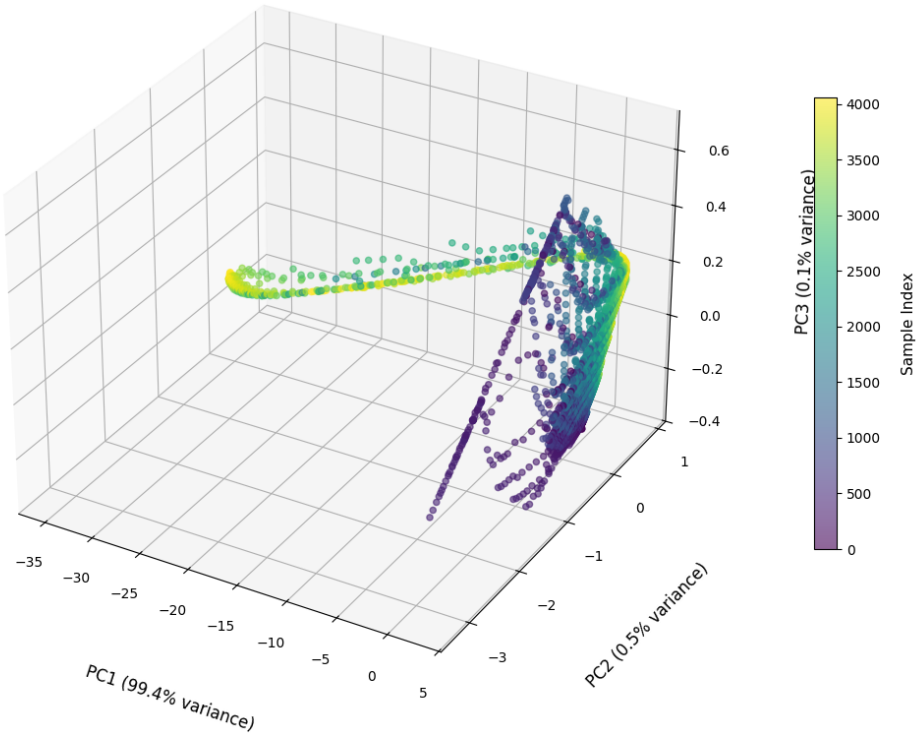

Figure 12: PCA projection of the Pareto front

Component	PC1	PC2	PC3	PC4
Explained Variance	0.9945	0.0050	0.0006	0.0000
Economic Reward	-0.0576	0.9683	0.2431	0.0000
Battery Health	-0.0009	-0.2436	0.9699	0.0000
Grid Support	0.9983	0.0557	0.0149	0.0000
Energy Autonomy	0.0000	0.0000	0.0000	1.0000

Table 6: Principal component loadings and explained variance

8 Future Work

8.1 Pareto Set Learning (PSL)

We plan to implement PSL as an alternative hyper-network approach. PSL uses parameter fusion between the hypernetwork and policy network, Potentially benefit from the advantage of both embodying and hyper-network approaches, Leading for better stability and faster convergence.

Key advantages of PSL:

- No warmup stage required (unlike Hyper-Morl)
- The Parameter may provides better stability [5]

- has guarantee of convergence, similar to PD-MORL [5]
- may have similar benefis to Hyper-Morl algorithm, as to the use of hyper-network

8.2 Multi-Agent Extension

Extending the framework to handle both ISO and PCS agents simultaneously would provide a more realistic simulation of grid operations. This presents additional challenges:

- Coordination between multiple objectives across agents
- Emergent behaviors from multi-agent interactions
- Scalability to larger grid systems

8.3 Improving Action Utilization

Investigating why agents fail to perform buy/sell actions effectively:

- Reward shaping to encourage diverse action usage
- Action space modifications
- Exploration strategies tailored for the energy domain

8.4 Dynamic Preference Elicitation

Developing methods for real-time preference adjustment:

- Interactive interfaces for grid operators
- Adaptive preference learning from operator feedback
- Context-aware preference switching based on grid conditions

9 discussion

The utility-based approach and the multy policy approach cannot be directly compared, as the multy objective approach measured with the hyper-volume metric and the utility based approach measured with the scalarized reward. Furthermore, we had different reward normalization for each algorithm because they both where sensitive for the normalization and required diffrent normalization for finding a good traidoff between the objectives. Therfure, we have a direct performance comparison only for the utility based approach. But, as the MOSAC algorithm converge over 500,000 steps and the hyper MORL algorithm over 30 millions, worm-up stage not included, we claims that the Hyper-Morl algorithm might not be required if the energy net system have known preference weights.

9.1 Utility-Based Approach

9.2 Baslined comparsion

We found that from the single objective RL algorithms, TD3 algorithms have the best result for single objective and for two objectives, and PPO has the second best. All the baseline single RL algorithms failed in finding sufficient tradeoff for the 4 objective problem, as can be shown from the flat learning curve in the figure.

9.3 Comparison between MOSAC and sac

The most reliable baseline for our mosac algorithm is the single objective sac. The shared featured mosac algorithm has significant improvement over the sac algorithm for two and four objectives. The separate features mosac algorithm achieves only insignificant improvement over the sac algorithm for two objectives, and a mild improvement with the four objective problem.

Comparsion between MOSAC and the other baselines A for the other baseline algorithms, Four objectives MOSAC algorithm clearly outperformed all the single objective RL, but for the two objectives problem that wasn't the case. We believe that one of the reasons TD3 outperformed shares-featured MOSAC for the two objective problem, and TD3 and PPO outperformed sac for the two objective problem, is that TD3 and PPO are better algorithm then sac for that problem.

Comparison between shared features and separate features network For the four-objective problem and the two-objective problem, the shared features network outperformed the separate features network. But, For the four objective problem the separate features network converged much faster. We believe that the reason is that the shared-featured architecture creates more dependency between the objectives, therfure able to create a better tradeoff between the conflicting objectives. Because the separated features network run four single objectives critics simultaneity, it converges faster for the four objective problem.

Failures of MOSAC on single objectives MOSAC algorithms received poor results for the single objective problem. We believe that the main reason MOSAC algorithm have bad result on the single objective problem, and one of the reasons the algorithm has medicore performance on the two objective problem, is the method we choose for the loss calculation. while calculating the critic losses, we applied the MSE loss before the scalarization and not afterwards. Therfure, we might actually don't have garentee of convergence, As opposed to what we though. Another assumption, might be that the system was sensitive do to the training of pcs only, without the iso agent. unfortunately, We didn't use a model of the iso, only trained the pcs agent. If we trained then simultaneously we might get better results. Moreover, We choose a reward normalization the was very well suited for the four objectives problem, but might be less so for the two objectives and the one objective problems. To sum up, We probably should have calculate the MSE loss on the scalarized critic for better results.

9.4 Pareto-Based Approach

We cannot perform result comprassion for the Pareto-based approach, since we only implemented Hyper-Morl algorithm. We will analyze the results recived from the running of the Hyper-Morl algorithm instead.

We received that the 3D hyper volume metric converges to 758.65 as shown at figure 8. Therfure, we belive that the Hyper-Morl algorithm creates a good approximation to the pareto front. As can be shown at figure 7, the 3D projection of the approximated Pareto front is indeed a manifold in two dimentional space, therfure satisfises one of the assumption of the Hyper-Morl algorithm.

We assume that consumption and production action didn't performed because we didn't train the iso agent separately.

The first PCA component vastly representing the Grid Support reward with Grid Support value of 0.9983 (table 4). The second one representing Battery Health, and the third one Battery Health. Therfue, we can conclude Grid Support is the most important reward at the construction of the Pareto front.

The second PCA component has Battery Health value of -0.2436, and the first PCA component has Economic Reward of 0.2431. All the other calls at table 4 that aren't matching the main objective of the relevant PCA component are order of magnitude lower, we can conclude that Economic Reward and Battery Health are the main conflicting objectives. We found small overlap between the representing PCA components of the Economic reward and the Grid support. That suggest a mild conflict between the Grid support and the Economic reward objectives. since action and production action didn't performed, the Energy autonomy reward is orthogonal to other objectives as can be shown at the fourth line of table 4.

10 Conclusion

In this project, we explored two approaches for multi-objective RL in the EnergyNet environment: the utility-based approach and the Pareto optimality approach.

10.1 Summary of Findings

10.1.1 Utility-Based Approach

For the utility-based approach, we examined a multi-objective Soft Actor-Critic algorithm with two variants: shared features and separate features. From the result comparison, we conclude that:

- The shared features version of MOSAC achieved the best performance (reward of 28) for the four objectives problem
- MOSAC outperformed all single-objective RL baseline algorithms (SAC, PPO, TD3)
- MOSAC is the best algorithm for the four objective problem when the scalarization function is known and linear
- For MOSAC algorithm to have convergence garentee and improved performance we should calculate the loss of the scalarized Q-function, and not use scalarization on the Q-function loss.

10.1.2 Pareto-Based Approach

For the Pareto optimality approach using Hyper-Morl:

- Successfully approximated the Pareto front when preferences are unknown
- The hypervolume metric converged, indicating good coverage of the objective space
- Provided a continuous approximation of trade-offs between objectives

10.2 Key Insights

From the the result comperassion done in the discussion, We believe that the shared features architecture is the best for handling the utily base approach, but might work better with PPO or TD3.

From the Principal Component Analysis of the Hyper-Morl results, we concluded that:

- The most conflicting objectives are economic profit (energy arbitrage) and battery health
- Grid support dominates the first principal component, suggesting it's the most influential objective
- Energy autonomy appears orthogonal to other objectives

10.3 Limitations and Challenges

Despite promising results, several challenges remain:

- Limited exploration of alternative Pareto-based methods due to time constraints
- Hyper-Morl required significantly more training time than MOSAC
- No formal convergence guarantees exist for the hypernetwork approach
- Limited exploration of alternative Pareto-based methods due to time constraints
- MOSAC algorithm might have better result on the two objectives problem and have convergence guarantee if we Calculated the loss of the scalarized critic instead of the scalarised critic loss

10.4 Recommendations

Based on our findings:

1. **When to use MOSAC**: When scalarization weights are known and linear utility is acceptable
2. **When to use Hyper-Morl**: When the full Pareto front is needed or preferences may change
3. **Future work**: Implement PSL for comparison and investigate action utilization issues

10.5 Impact

This work demonstrates the feasibility and benefits of multi-objective reinforcement learning for power control systems. The ability to balance multiple competing objectives while providing transparent trade-off visualization represents a significant advancement for smart grid optimization. Our results provide a foundation for developing more sophisticated energy management systems capable of adapting to complex, real-world constraints.

References

[1] Toygun Basaklar, Suat Gumussoy, and Umit Ogras. Pd-morl: Preference-driven multi-objective reinforcement learning algorithm. In *International Conference on Learning Representations*, 2023.

[2] Andrea Castelletti, Francesca Pianosi, and Marcello Restelli. A multiobjective reinforcement learning approach to water resources systems operation: Pareto frontier approximation in a single run. *Water Resources Research*, 49(6):3476–3486, 2013.

[3] Donghui Chen, Yiping Wang, and Wei Gao. Combining a gradient-based method and an evolution strategy for multi-objective reinforcement learning. *Applied Intelligence*, 50:3301–3317, 2020.

[4] Conor F Hayes, Roxana Rădulescu, Eugenio Bargiacchi, et al. A practical guide to multi-objective reinforcement learning and planning. *Autonomous Agents and Multi-Agent Systems*, 36(1):26, 2022.

[5] Erkai Liu, Yao-Cheng Wu, Xin Huang, Chao Gao, Run-Jie Wang, Ke Xue, and Chao Qian. Pareto set learning for multi-objective reinforcement learning. *National Key Laboratory for Novel Software Technology, Nanjing University*, 2025.

[6] Tianxin Shu, Ke Shang, Chen Gong, Yang Nan, and Hisao Ishibuchi. Learning pareto set for multi-objective continuous robot control. In *Proceedings of the 33rd International Joint Conference on Artificial Intelligence*, pages 4920–4928. IJCAI, 2024.

[7] Kristof Van Moffaert and Ann Nowé. Multi-objective reinforcement learning using sets of pareto dominating policies. *Journal of Machine Learning Research*, 15(1):3483–3512, 2014.

A Implementation Details

A.1 Environment Wrapper

The multi-objective environment wrapper extends the base EnergyNet environment to provide vector rewards. Key modifications include:

- Decomposition of the original reward into four components
- Normalization of individual objectives to similar scales
- Support for different scalarization methods

A.2 Network Architectures

A.2.1 MOSAC Networks

- Actor: 3 hidden layers (256, 256, 128) with ReLU activation
- Critics: 2 hidden layers (256, 128) per critic head
- Output: Continuous action space with tanh activation

A.2.2 Hypernetwork

- Hypernetwork: 2 hidden layers (128, 256)
- Policy network: 2 hidden layers (64, 64)
- Preference embedding: Linear transformation of preference vector

A.3 Computational Resources

All experiments were conducted on a remote Linux server with:

- GPU: NVIDIA V100 (32GB)
- CPU: Intel Xeon Gold 6248 (40 cores)
- RAM: 384GB
- Training time: MOSAC (2-4 hours), Hyper-Morl (48-72 hours)