ORIGINAL ARTICLE

A cross-entropy based direct policy search algorithm for multi-objective energy storage control

Gabriel Matos Cardoso Leite · Carolina Gil Marcelino · Silvia Jiménez-Fernández · Elizabeth Fialho Wanner · Sancho Salcedo-Sanz · Carlos Eduardo Pedreira

Received: 9 November 2024 / Accepted: 9 October 2025

Abstract
Effective control of Energy Storage Systems (ESS) is crucial for the secure and profitable operation of microgrids. In this context, ESSs are essential for enhancing the overall grid resilience, balancing supply, and mitigating voltage and frequency variations. This paper presents a novel neuroevolutionary method, coupling a modified version of the Multi-Objective Evolutionary Policy Search (MEPS) algorithm with the Cross-Entropy method, aimed at optimizing an ESS control problem. The modified MEPS, named Cascade-MEPS, employs a cascade weights mutation operator to refine policies by focusing on the most recent hidden node, ensuring localized and non-disruptive adjustments. The resulting algorithm, referred to as cross-entropy Cascade-MEPS (CE-CMEPS), utilizes the cross-entropy method as a depth initialization strategy, conducting an initial exploration of the weights space to initialize the population prior to Cascade-MEPS execution. Experimental validation on a newly proposed multi-objective ESS control problem demonstrates the efficacy of CE-CMEPS, showcasing performance improvements and reduced variation compared to standalone MEPS. Our results show that CE-CMEPS is an effective ESS discharge controller and a sustainable multi-objective reinforcement learning solution.

Keywords Direct policy search (DPS) · Multi-objective (MO) control · Reinforcement learning · Neural networks architecture · Neuroevolution · Energy management

1 Introduction

In the past decades, the challenge of meeting energy demand and reducing carbon emissions has drawn attention from both academia and industry. Consequently, renewable energy sources (RES), such as wind turbines (WT) and photovoltaic panels (PV), have been playing a major role in the transition to a less-pollutant energy generation, facing not only technological improvements [1] but also cost reductions [2]. To mitigate intermittency and non-continuity production of renewable energy generation, RES are often deployed alongside energy storage systems (ESS) [3, 4] in microgrids (MGs). Microgrids are self-sustaining generation sources that include RES, various energy storage devices, and possibly fossil fuel generation sources such as diesel or gas generators [5, 6].

Published online: 13 February 2026

Efficient energy management is one of the most important factors affecting the quality and reliability of MGs [7]. Among the different RES present in a microgrid, it is worth noting the importance of energy storage management in a secure MG operation [8]. An effective ESS control strategy can lead to energy cost reduction and MG operational stability improvement [9]. The search for optimal ESS control strategies often includes utilizing classical optimization methods [10], heuristic optimization methods [11], and reinforcement learning methods [12], to name a few. Classical and heuristic optimization algorithms have long been recognized as suitable methods to handle ESS control problems. However, these algorithms, as iteration-based solvers, have faced some significant limitations. First, when dealing with large dimensional problems, a large number of iterations is required for population updating or iterative searching. Second, the algorithm may need to be restarted at each iteration and/or when there is a slight change in the problem [13]. On the other hand, reinforcement learning (RL) algorithms, do not need to be restarted at every iteration and are robust to high-dimensional state spaces as well as slight changes in the target problem [14].

In this way, the control of battery ESSs using single-objective reinforcement learning has been widely studied in the last few years. For instance, a model-free algorithm based on a periodic action-value function and deterministic policy gradient is proposed to manage a multi-battery ESS under a residential microgrid [15]. Additionally, in [16], a deep RL solution based on the actor-critic algorithm is presented to address the electricity arbitrage problem in optimal ESS management of a commercial/residential building. Besides actor-critic-based algorithm, the Q-Learning algorithm combined with (deep) neural networks for state-action value function has become a popular RL alternative to energy management problems. In [17], a deep Q-Learning with prioritized replay is employed to manage the scheduling of an ESS in a residential ESS-integrated PV system. A deep Q-Learning algorithm is also utilized in [18], in which it controls the amount of energy bought from the public grid to charge the storage system. Moreover, the authors in [19] employed Q-Learning to search for an optimal ESS charge/discharge strategy in a microgrid, considering residential and commercial load demands.

To handle problems with a large or infinite state space, roughly all RL algorithms utilize the generalization abilities of function approximators in estimating value functions [20]. Feedforward neural networks (NNs) are a particular case of such function approximators that have been successfully employed in combination with reinforcement learning methods to overcome the aforementioned limitations [21, 22]. Specifically, neural networks employing Rectified Linear Units (ReLU) as activation functions have become popular due to their practical performance [23]. Yet, the performance of these networks depends on their complexity, specifically the careful selection of the network's topology and architecture (number of layers, nodes, and connections) as well as the size of the parameters [24]. A poorly chosen network topology may hinder its ability to perform the intended task, even with extensive training. Therefore, many studies have aimed to establish a correlation between neural network complexity and its learning capacity.

In [25], the authors have proven that adapting the NN's topology to the function being approximated leads to a smaller upper bound for learning error, compared to fixing the topology and only adjusting the weights, for the specific case of Sobolev Spaces. Moreover, the authors in [26] have shown that, for a given dense network, there is a sub-network with fewer nodes and connections that, when trained in isolation, achieves comparable performance as the original one. Recently, the importance of constraining the topology of a neural network is reinforced by tightening the learning capability bound of ReLU-based NNs in [27]. The VC-dimension [28] of this class of networks has been proven to have a linear dependency on the number of nodes and connections. Thus, the ultimate performance of a policy parameterized by an ANN depends not only on the weight values but also on the proper selection of the number of nodes and connections in each NN.

Even though RL algorithms usually target problems in which the feedback signal from the environment is scalar, many real-world control problems are inherently too complex and often involve dealing with multiple objectives [29]. Multi-policy multi-objective reinforcement learning (MORL) algorithms have arisen as a viable option

for delivering numerous trade-offs among objectives, offering an effective means to identify superior trade-off solutions [13, 30]. Most of the MORL proposals in the state-of-the-art rely on policy gradient updates [31–34]. However, it is widely known that gradient-based optimization is subject to getting trapped in local minima [35]. Thus, direct policy search (DPS) [36] is emerging as a popular MORL alternative method to alleviate gradient issues. DPS defines the control policy within a given functional parametrization and explores the policy parameters space by searching for the best solution concerning a given set of objectives. Our proposal addresses the following issues that summarize the major contributions of this article:

- We propose a new multi-objective energy storage control problem for a solar-wind microgrid considering three objectives: CO_2 emissions, operational cost, and a penalty for maintaining the battery at a low state of charge (SoC);
- A novel modified version of the multi-objective evolutionary policy search (MEPS) that performs local adjustments to policies by constraining weights mutation;
- A combination of the multi-objective Cross-Entropy (CE) method with the modified MEPS algorithm to control the charge/discharge strategy of an ESS in a microgrid for a working horizon of one week;
- A comparison of the proposed method with standard MEPS, as well as with two MORL algorithms based on actor-critic and Q-learning, and;
- A thorough statistical analysis of the performances of each algorithm in terms of hypervolume indicator. We leverage the robust generalization capabilities of RL, the benefits of gradient-free optimization, and the search efficiency of evolutionary algorithms.

The remainder of this article is organized as follows: Sect. 2 elaborates on the detailed modeling of the studied battery energy storage system control. Section 3 presents the proposed coupled multi-objective direct policy search approach. The effectiveness of the method is demonstrated through experiments in Sect. 4. Finally, conclusions are presented in Sect. 5.

2 Problem formulation

This section formally defines the energy storage control problem over a week-long planning horizon. Initially, the configuration of the ESS-integrated solar wind power microgrid system is outlined, followed by a detailed presentation of the constraints and the objective functions. Subsequently, the control problem is formulated as a Markov Decision Process (MDP) [37] and treated from a multi-objective standpoint, considering operational costs, CO_2 emissions, and penalties for surpassing the ESS capacity constraint. In this context, the learning agent functions as an energy management system (EMS), designed to manage the discharge of energy from the ESS and the importation of energy from the public grid.

2.1 Microgrid simulation model

The configuration of the simulated microgrid system is based on the frameworks presented in [38, 39]. This system comprises 200 photovoltaic panels [40], a wind turbine [41], a 140 kW Lithium-ion battery energy storage system (ESS), a DC/AC converter, an electrical load, a main grid connection with real-time pricing (RTP), and an energy management system (EMS). Figure 1 illustrates the system structure. Table 1 details the specifications of the microgrid (MG) project, which is designed for a 24-year operational lifetime.

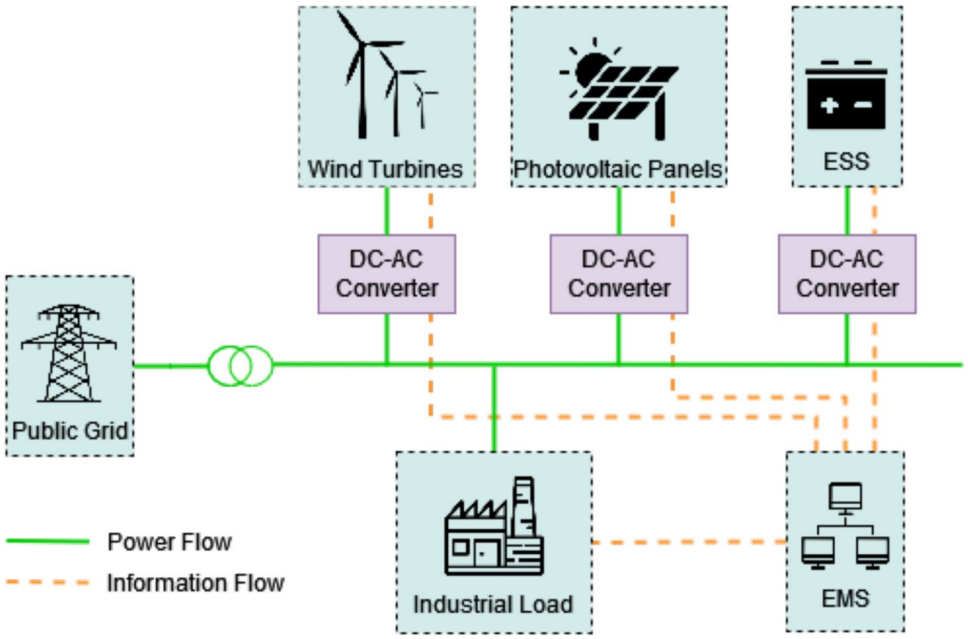


Fig. 1 Simulated microgrid system design. Based on [42]

Table 1 Configuration values for the microgrid project

	DC/AC converter	PV	WT	Battery
Life time (years)	15	24	24	17.5
Efficiency (%)	96	20.4	95	90
Rated power (kW)	–	0.45	100	–
Capacity (kW)	–	–	–	1000
Cycles (un)	–	–	–	8000
Initial cost (€)	-	500.00	1800.00	–
Cost (€/kW)	700.00	–	–	1143.00
Operational cost (€/kW)	–	18	0.36	–

The electrical load consists of the hourly energy demand over a week in 2019 (measured over 168 h) from a set of industrial and commercial buildings in a region of Belgium [43]. Additionally, the MG system utilizes hourly data from 2019 for dynamic energy pricing, wind speed, ambient temperature, and solar radiation from [43]. The following section presents the objective functions as well as the various constraints present in the problem.

2.2 Operating costs and ESS restrictions

The charge and discharge management of the ESS dictates whether energy is being stored or utilized for each time step t. The SoC, at each time step, is calculated according to [42]:

$$SoC(t) = \begin{cases} SoC(t-1) + \frac{P_{ch}(t)\cdot\eta_c}{P_{rated}}, & \text{if charging} \\ SoC(t-1) - \frac{P_{dch}(t)}{P_{rated}\cdot\eta_d}, & \text{if discharging,} \end{cases} \quad (1)$$

in which $\eta_d = 1.0$ and $\eta_c = 0.90$ represent the discharging and charging efficiencies, respectively. P_{rated} refers to the nominal capacity of the ESS. The power requested for charging, $P_{ch}(t)$, is given by [38, 39]:

$$P_{ch}(t) = \min\{(SoC_{max} - SoC(t-1)) \cdot P_{rated}, P_{ch}(t)\}, \tag{2}$$

in which SoC_{max} refers to the maximum allowed SoC. The power discharge request, denoted as $P_{dch}(t)$, is regulated by the learning agent and should not exceed the available energy, $P_{avl}(t) = SoC(t-1) \cdot P_{rated}$, as shown in Eq. (3):

$$P_{dch}(t) = \begin{cases} P_{dch}(t), & P_{dch}(t) <= P_{avl}(t) \\ P_{avl}(t), & \text{otherwise.} \end{cases} \tag{3}$$

Furthermore, the operation takes into account both the utilization and degradation expenses of ESS. The cost of ESS utilization is given by [39, 44]:

$$C_{ESS}(t) = \frac{INV_{ESS}}{L_c \cdot P_{rated} \cdot DoD(t)}, \tag{4}$$

in which the initial investment of ESS and its available cycle lifespan are denoted by L_c and INV_{ESS}, respectively. Consequently, within this modeling framework, the total operational cost is determined according to [44]:

$$C_{total}(t) = IC + C_p + C_{np} + C_{ESS}(t), \tag{5}$$

in which the value $C_{total}(t)$ denotes the total costs of operating the system per time *t*. The initial cost (IC) encompasses 20% of the operation and maintenance costs, a 1.4% rate for inflation, a 6% discount rate, personnel costs, installation expenses, and connection fees. It also includes both non-periodic costs (C_{np}) for the replacement of components such as the ESS [44], and periodic costs (C_p) for the maintenance of solar and wind generation components. Additionally, the constraints on charging and discharging power are specified as follows:

$$Constraints_{ESS} = \begin{cases} 0 \leq P_{ch} \leq P_{ch}^{max}, \\ 0 \leq P_{dch} \leq P_{dch}^{max}, \\ P_{ch} \cdot P_{dch} = 0, \end{cases} \tag{6}$$

where P_{ch}^{max} and P_{dch}^{max} represent the maximum charge and discharge power, respectively. Furthermore, the minimum state of charge (SoC) is defined as $1 - DoD(t)$, in which *DoD* indicates the battery's depth of discharge. The objective function for cost minimization, as formulated by [38, 39], is subsequently presented as follows:

$$Cost(t) = C_{total}(t) + P_t^{buy} \cdot Pr_t, \tag{7}$$

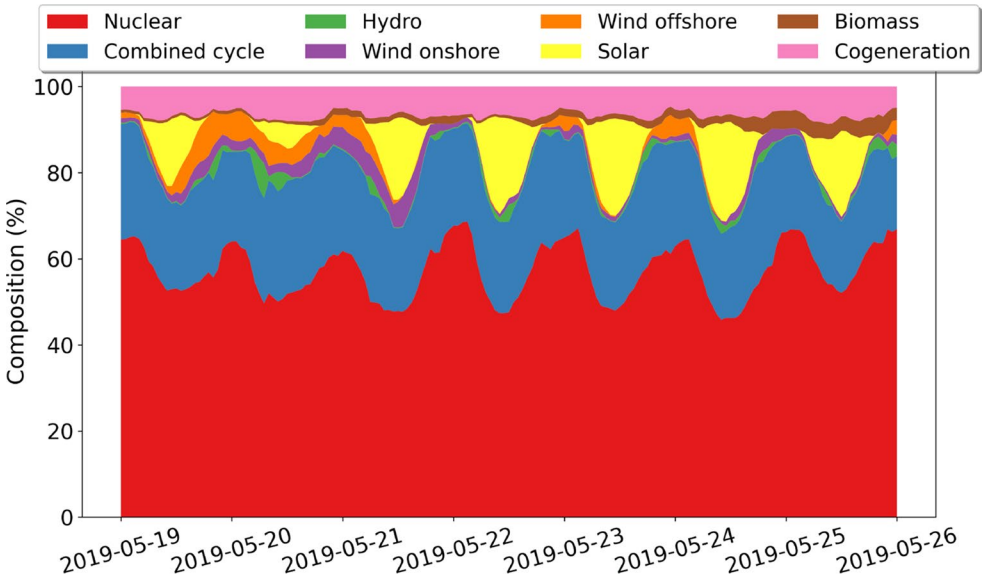


Fig. 2 Hourly dispatchable energy composition in Belgium for a week in 2019. Figure built using data available in [45]

in which the cost of operation is joined with the cost of purchasing public grid energy P_t^{buy} at price Pr_t to supplement the deficient microgrid power.

2.3 CO_2 emissions

To provide an estimate of the amount of grams of CO_2 emissions per kWh for public grid energy purchased and generated from renewable sources, an average of between the minimum and maximum greenhouse gas emission values obtained from [2] are employed as follows:

- Solar Photovoltaic emissions (CO_{2PV}) = 44.15 g CO_2eq./kWh;
- Wind Power onshore emissions ($CO_{2WT_{on}}$) = 11.90 g CO_2eq./kWh;
- Wind Power offshore emissions ($CO_{2WT_{off}}$) = 17.50 g CO_2eq./kWh;
- Nuclear emissions (CO_{2NU}) = 5.75 g CO_2eq./kWh;
- Hydro emissions (CO_{2HY}) = 76.50 g CO_2eq./kWh;
- Cogeneration (CO_{2COG}) and Combined cycle emissions (CO_{2CC}) = 156.00 g CO_2eq./kWh;

Moreover, concerning energy purchased from the public grid, the CO_2eq./kWh is calculated based on the hourly dispatchable energy composition in Belgium, as detailed in [45]. Figure 2 shows the hourly dispatchable energy composition for the considered operation week. It is worth noting that biomass emissions are not considered due to difficulties in estimating its CO_2 emissions.

Hence, the emissions in CO_2eq./kWh [11] for renewable generation and energy purchased from the public grid are:

$$Emission_{WT}(t) = CO2_{WT_{on}} \cdot P_t^{WT} \tag{8}$$

$$Emission_{PV}(t) = CO2_{PV} \cdot P_t^{PV} \tag{9}$$

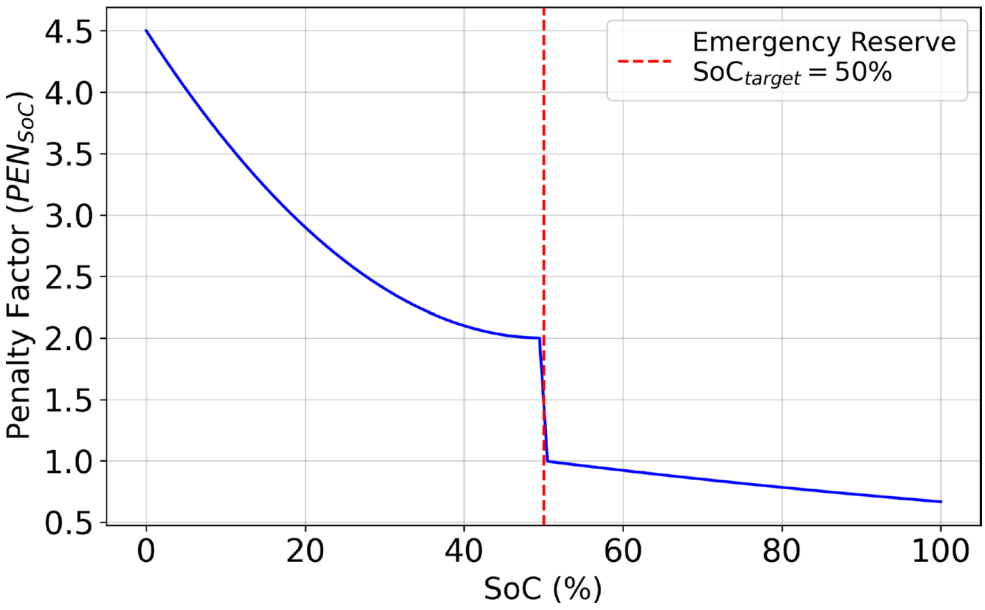


Fig. 3 Penalty factor for different SoC values. Based on [13]

$$Emission_{Buy}(t) = \sum_{RES \in GC_t} CO2_{RES} \cdot Composition(t) \cdot P_t^{Buy}, \tag{10}$$

in which P_t^{PV} and P_t^{WT} denote the amount of energy produced at each time t by photovoltaic panels wind turbines, respectively. The set of renewable energy sources that compose the public grid energy and their corresponding percentage at time t are given by GC_t and *Composition*(t), respectively. The objective function for minimizing CO_2 emissions is derived in accordance with the methodology outlined in [11]:

$$Emission(t) = Emission_{WT}(t) + Emission_{PV}(t) + Emission_{Buy}(t). \tag{11}$$

2.4 ESS state of charge penalty

The proposed model incorporates a penalty function associated with utilizing the energy storage system (ESS) below a 50% state of charge (SoC) threshold, denoted as SoC_{Target}, which serves as an emergency reserve. This penalty discourages excessive depletion of the ESS, thereby reducing reliance on energy purchases from the public grid during renewable energy outages. By enforcing a minimum SoC level, the model ensures that sufficient stored energy remains available to maintain microgrid operation when renewable generation is unavailable.

Introducing a penalty for SoC levels below 50% is further justified by the operational and technical limitations inherent in battery energy storage systems. Deep discharging accelerates capacity degradation and reduces cycle efficiency, particularly in lithium-ion batteries, where it can induce electrode damage and uneven cell aging. Maintaining a higher minimum SoC also preserves reserve capacity, enabling rapid

response to grid contingencies and sudden demand spikes. This operational buffer enhances system resilience and supports grid stability, especially in contexts with high penetration of intermittent renewable sources.

Consequently, penalizing SoC values below 50% promotes a conservative and sustainable operational strategy, aligning with objectives of long-term battery health preservation and reliable grid support. The corresponding objective function for minimizing the accumulated SoC penalty is defined in Eq. (12) [13]:

$$PEN_{SoC}(t) = \begin{cases} (SoC(t) - SoC_{target})^2 \cdot m_1 + m_2 & \text{if } SoC(t) \leq SoC_{target} \\ exp\left(-\log(m_3) \cdot \frac{SoC(t) - SoC_{target}}{1 - SoC_{target}}\right) & \text{otherwise,} \end{cases} \tag{12}$$

in which $m_1 = 10$, $m_2 = 2$ and $m_3 = 1.5$ are used to control the stringency of M. Figure 3 illustrates the behavior of the penalty factor PEN_{SoC} for different SoC levels [13].

2.5 Storage control Markov decision process

In this problem, energy storage management involves controlling the discharge of the ESS on an hourly basis. The learning agent is tasked with determining the percentage of available ESS energy to be utilized each hour. To address insufficient microgrid power, the system relies on importing from the public grid, necessitating information on both environmental conditions and the current battery state of charge (SoC) across various states. Thus, the operational characteristics are formally represented through a MDP [37], incorporating a state space denoted as *S*, an action space denoted as *A*, and rewards *R*, which are assessed on an hourly basis across one week. The state space *S* is defined by a seven-dimensional vector in $\mathbb{R}^7$, encapsulating state details for each hour *t* [42]:

$$s_t \in S = \left\{ SoC(t), P_t^{load}, Pr_t, P_t^{PV}, P_t^{WT}, \frac{P_t^{buy}}{P_t^{load}}, \frac{P_{dch}(t)}{P_t^{load}} \right\}, \tag{13}$$

in which P_t^{load} denotes the load demand at *t* hour, Pr_t is the to supply the insufficient microgrid power,P_t^{PV}, and P_t^{WT} denote the amount of energy produced at each time *t* by photovoltaic panels wind turbines. The power discharge request, denoted as $P_{dch}(t)$, is regulated by the learning agent and should not exceed the available energy, and P_t^{buy} is the purchasing of public grid energy. Moreover, at each hour, the agent needs to select the percentage of the available ESS energy that will be used. Thus, there are 11 possible actions in action space:

$$a_t \in A = \{0\%, 10\%, 20\%, 30\%, 40\%, 50\%, 60\%, 70\%, 80\%, 90\%, 100\%\}. \tag{14}$$

Upon execution of an action, the dynamics of the energy storage system (ESS) undergo updates in the following manner:

$$SoC(t) = SoC(t-1) - a_t \cdot SoC(t-1) \tag{15}$$

$$P_{dch}(t) = a_t \cdot P_{dch}(t). \tag{16}$$

Note that the ESS is not discharged if the load demand at time t is already met by the renewable energy generation. Thus, an action that leads to discharging the battery in this situation is equivalent to performing no discharge ($a_t = 0\%$). Furthermore, when the energy storage system (ESS) maintains a sufficient state of charge (SoC) to meet the local demand, but the control agent elects to discharge less power than required, the resulting power deficit is compensated by importing energy from the public grid. The first objective is to minimize the system operating cost, denoted as *Cost*(*t*). The second objective is to minimize CO2 emission, represented by emission(t). In addition, the term $PEN_{SoC}(t)$ denotes the penalty function associated with the SoC objective. Finally, the multi-objective reward function is a $\mathbb{R}^3$ vector as follows:

$$\mathrm{R}(s_t, a_t) = \left[Cost(t), Emission(t), PEN_{SoC}(t)\right]. \tag{17}$$

After defining the MDP, reinforcement learning is employed to address the resulting problem. The operational behavior of the learning agent within the designed control problem can be outlined as: (1) Selecting an action a_t; (2) Executing microgrid operations for one hour; (3) Adjusting the current state s_t; (4) Updating costs; (5) Updating emissions; (6) Updating SoC penalty; (7) Repeating steps (1)–(6).

3 Proposed multi-objective algorithm

3.1 MEPS algorithm

The Multi-objective Evolutionary Policy Search (MEPS), introduced in [38], represents a model-free approach designed for estimating action-preference values within MORL environments. MEPS is part of the "actor-only" family of reinforcement learning methods and employs the NEAT (NeuroEvolution of Augmenting Topologies) framework [46] to evolve artificial neural networks (ANNs) for implementing deterministic policies. By evolving network policies, MEPS circumvents the need for gradient updates or the computation of value-function estimations. Moreover, MEPS employs population-based methods to generate a diversified Pareto-optimal set of policies, thereby classifying it as a multi-policy algorithm [30].

Initially, an initial random population P_t ($t = 0$), consisting of s_p ANNs, each equipped with an output node corresponding to every potential action, is generated. The evaluation of individuals is based on a vectorial reward function over a fixed horizon H, each individual's accumulated reward denoted as r_H. The networks in population P_t are then sorted via non-dominated sorting and a density measure, using the accumulated reward. The ANNs used in MEPS are constructed to generate preference values $p(s, a)$ for every possible action a when provided with an input state s_t. Furthermore, to ensure deterministic policy behavior, the agent selects actions greedily.

MEPS utilizes two types of density measures: (1) crowding distance (CD) [47], and (2) hypervolume contribution (HVC) [48]. The former aims to evenly distribute solution points along the Pareto front, ensuring uniform coverage, while the latter is designed to distribute solution points to maximize the hypervolume covered, prioritizing knee points while retaining extremal points.

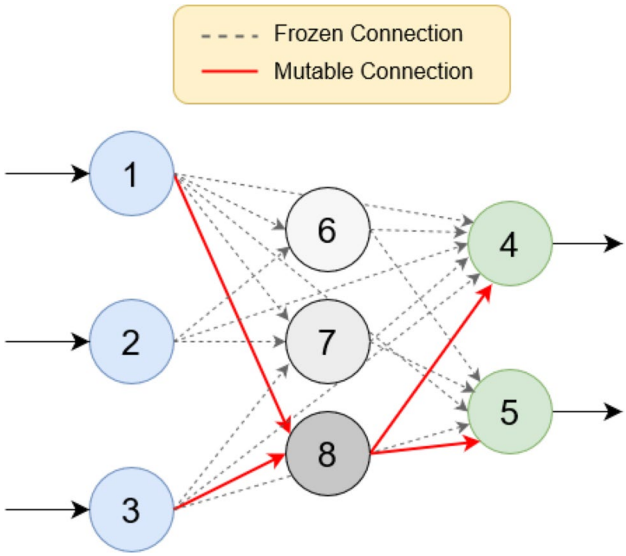


Fig. 4 Illustration of the cascade weight mutation operator behavior. The weights of connections that are not associated with the most recent hidden node are frozen

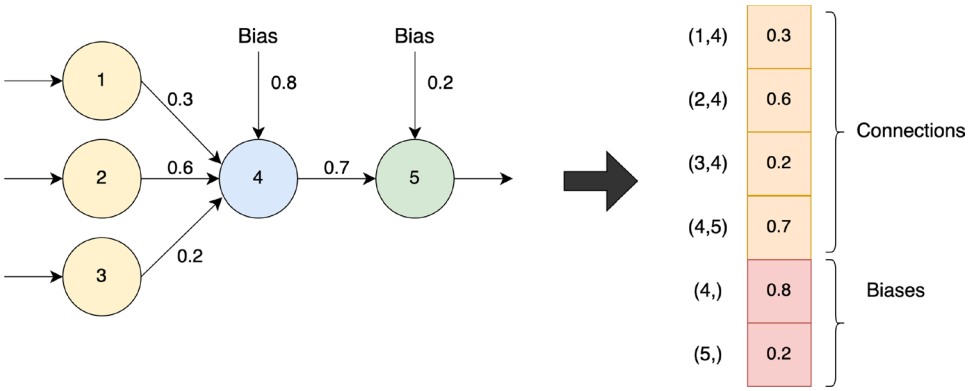


Fig. 5 Illustration of the encoding of MEPS individuals into real-valued vectors

During the evolutionary process, a set of parents U_t comprising s_p individuals is selected randomly from the population P_t using crowding binary tournament selection [47]. A subsequent offspring population Λ_t is produced by duplicating individuals from U_t and implementing two distinct forms of mutations: structural and parametric. Structural mutations occur probabilistically and involve (1) connecting two previously unlinked nodes (p_{ac}), and (2) introducing a new hidden node into the network architecture (p_{an}). It is worth noting that the individuals in MEPS are ReLU-based feedforward ANNs, which inherently do not incorporate recurrent connections. Parametric mutation comprises adjusting weights and biases of connections by the addition of noise characterized by a Gaussian distribution $\mathcal{N}(0, \sigma)$ with σ as parameter.

The next generation population P_{t+1} consists of the survivors chosen from the population $R_t = P_t \cup \Lambda_t$ with a total size of $2s_p$. MEPS employs two different approaches for survival selection. The first involves sorting the population R_t via non-dominated sorting into fronts or ranks, and thereafter iteratively selecting individuals for the next generation based on their front, similar to the NSGA-II approach [47]. If a front exceeds the available slots in the generation, only the best individuals from lower-density regions are chosen, as determined by the selected density measure. The second approach involves adaptively limiting the number of survivors that are selected from each of the fronts. In the context of multi-objective optimization, during the initial generations, less favorable non-dominated individuals might be prioritized over individuals from alternative fronts [49].

Table 2 Parameters description

Parameter	Description
n_p	Population size
S	Density measure function
$\phi(x)$	Activation function
$\phi_o(x)$	Output activation function
ψ	Initial fraction selected from first front
t_{max}	Total generations
t_r	End generation number of heavy tail survivor selection
α	Heavy tail selection parameter
n_i	Number of input nodes
n_h	Number of initial hidden nodes
n_o	Number of output nodes
p_{ac}	Add connection mutation probability
p_{dc}	Delete connection mutation probability
p_{an}	Add node mutation probability
p_{dn}	Delete node mutation probability
σ	Parametrical mutation standard deviation
HT	Indicate the use or not of the heavy tail survivor selection
h	Length of the episode to evaluate the agent

Similarly, from a neuroevolutionary point of view, discarded potential solutions may be indicative of novel topologies that are prematurely extinct. To prevent this problem, MEPS uses a selection method based on a heavy-tailed Pareto distribution [50] that enables some individuals from higher ranks to survive. In this selection method, the maximum survivors for each *i*-th front is determined according to Eq. (18)

$$n_{F_i} = \begin{cases} \frac{\beta/i^{\beta+1}}{\sum_{i>1}^{K} \beta/i^{\beta+1}} \cdot \left(s_p - \lceil s_p \cdot ratio \rceil\right), & i > 1 \\ \lceil s_p \cdot ratio \rceil, & i = 1, \end{cases} \tag{18}$$

in which *ratio* and *K* indicate the fraction selected from the first front and the number of non-dominated fronts, respectively. The parameter β defines the distribution's tail. To avoid losing promising topologies and converging to an incomplete Pareto front prematurely, *ratio* is incrementally increased with the number of generations, as described by Eq. (19)

$$ratio = \begin{cases} 1, & t_r < t < t_{max} \\ \psi + t_r \cdot \frac{(1-\psi)}{t}, & \text{otherwise}, \end{cases} \tag{19}$$

where ψ lies in the (0, 1) interval and represents the beginning proportion of non-dominated individuals that will be drawn This proportion is is gradually increased over the course of generations to reduce the exploration of solutions from higher ranks. Additionally, after a predefined number of generations t_r, the heavy-tailed selection mechanism is substituted by the first approach. Importantly, if there are fewer individuals in a front than its maximum allowed survivors, the remaining slots are cyclically allocated to the next front until all slots are filled. Furthermore, to retain the best individuals discovered across generations, MEPS updates a memory of size s_p with the next population P_{t+1}.

3.2 Cross-entropy based cascade MEPS

In this work, we propose both a modified MEPS version and a novel coupled algorithm from the combination of the Cross-Entropy method and the aforementioned modified MEPS version. First, the proposed modified MEPS version, dubbed Cascade-MEPS (CMEPS), is inspired by the Cascade-NEAT algorithm presented in [51]. The Cascade-NEAT algorithm employs a cascade mutation operator that alters both the "Add node" and "Add connection" mutation operators. This change constrains not only recent nodes to be connected to all previously existing nodes but also weight mutations to occur only on connections associated with the most recent hidden node. The Cascade-NEAT algorithm demonstrated superior performance in single-objective RL problems, where the optimal actions for one state often vary abruptly compared to neighboring states, resulting in high variation and discontinuity in the best solutions. Previous neuroevolution research suggests that a strategy to address such problems is to implement local and non-disruptive adjustments to policies [52].

The cascade mutation operator, as adopted in our work, is inspired by the Cascade-NEAT architecture proposed by Kohl and Miikkulainen [51]. The central idea is to restrict weight modifications to the most recently added node in the network, rather than applying changes across the entire network. This design stems from the insight that, in highly discontinuous decision spaces, large-scale mutations can disrupt the finely tuned behavior of existing network components. By allowing only the newest node to be mutated, the network incrementally refines its decision boundaries in a localized and stable manner. The rationale is to gradually refine new contributions to the model without inadvertently altering previously established structures that may already be performing well. Given the significant variation in ESS control due to various factors each hour, we propose a novel version of MEPS. This version implements a modified cascade mutation operator that constrains the weights mutation operator to make localized adjustments to each policy, specifically targeting connections associated with the most recent hidden node, as depicted in Fig. 4.

Cross-entropy method (CE) is an evolutionary algorithm initially proposed by [53] for estimating probabilities of rare events in complex stochastic networks, and later extended to solve optimization problems. CE is a Monte Carlo technique for sampling and optimization that can be applied to combinatorial and continuous problems. It

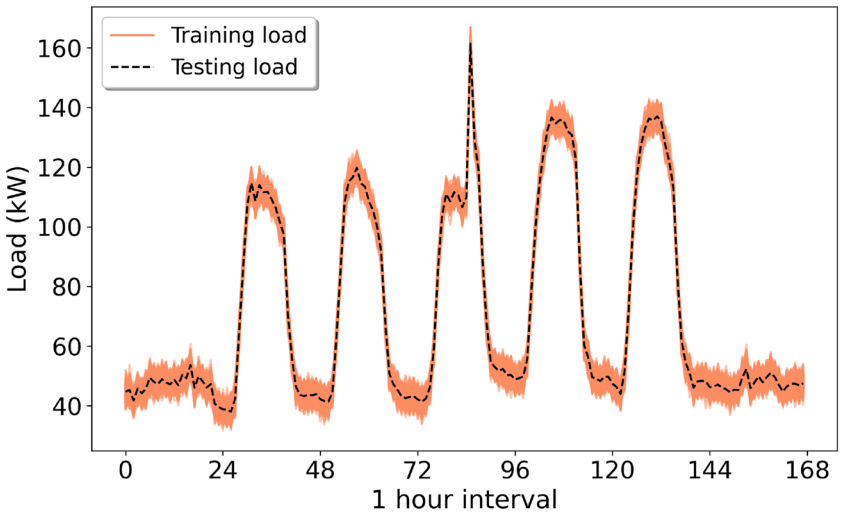
Table 3 Parameter initialization values used in the ESS-integrated solar wind power microgrid problem

Description	CE-CMEPS	MEPS	MPSAC	MODQN
s_p	50			
Initial *ratio*	0.5		–	–
t_r	250		–	–
β	1.0		–	–
p_{ac}	0.2		–	–
p_{an}	0.2		–	–
σ	0.5		1.0	–
Learning rate	–	–	0.001	0.001
Gamma	–	–	0.99	0.99
Initial epsilon	–	-	–	0.1
Epsilon decay	–	-	–	$3.7 \cdot 10^{-5}$
Smoothing coefficient	0.5	–	–	–
Generations	50 + 450	500	250 + 250	500
H	168			

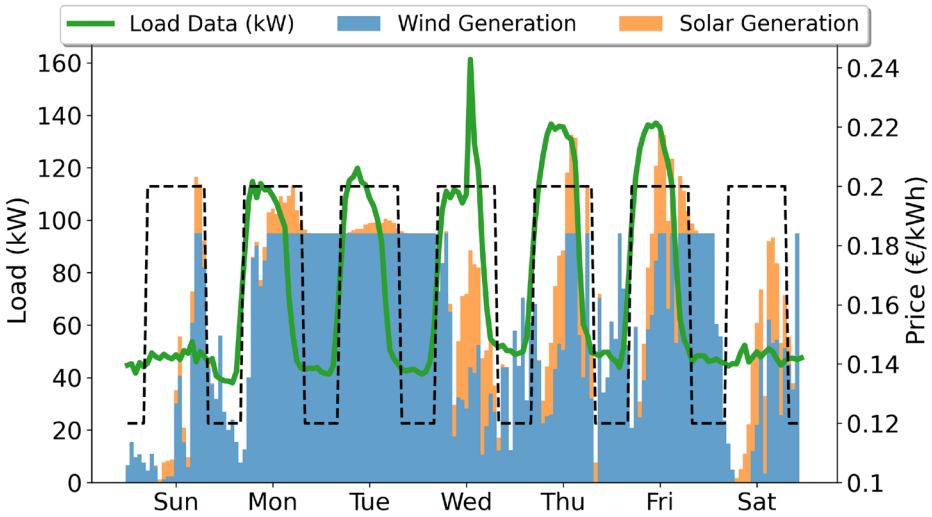
is distinct from the classical cross-entropy definition in information theory.

Here, a multi-objective CE version based on [54] acts as an initial depth search operator and optimizes the initial population weights using CE to find a promising basin of attraction to initialize MEPS' population. In

Fig. 6 Generation of training and testing load scenarios



(a) The black line delineates the test scenario, whereas the green area illustrates the spectrum of noise incorporated into the test scenario to generate the training scenarios.



(b) Detailing of the test scenario.

the first step, CE is initialized with a population of ANNs with random weights encoded as illustrated in Fig. 5. Subsequently, CMEPS is utilized to evolve the ANN topology starting with weights obtained from the first step to estimate action-preference values in MORL. The motivation for this proposal is based on the promising results of both CE as initial depth search shown in [6, 55], and a two-step evolution approach for single-objective policy search proposed in [56].

The multi-objective CE method for optimization employed in this work can be summarized in Algorithm 1 [54]. Algorithm 2 shows the pseudocode for the proposed approach. All parameters are explained in Table 2. The time complexity of the Algorithm 2 is divided into two parts: (1) CE complexity and (2) CMEPS complexity. According to [55], the CE time complexity is of order $O(s_p^3)$. Since the cascade weights mutation operator does not change the time complexity of MEPS standard weights mutation operator, the time complexity for CMEPS is

 https://doi.org/10.1007/s00521-025-11785-3

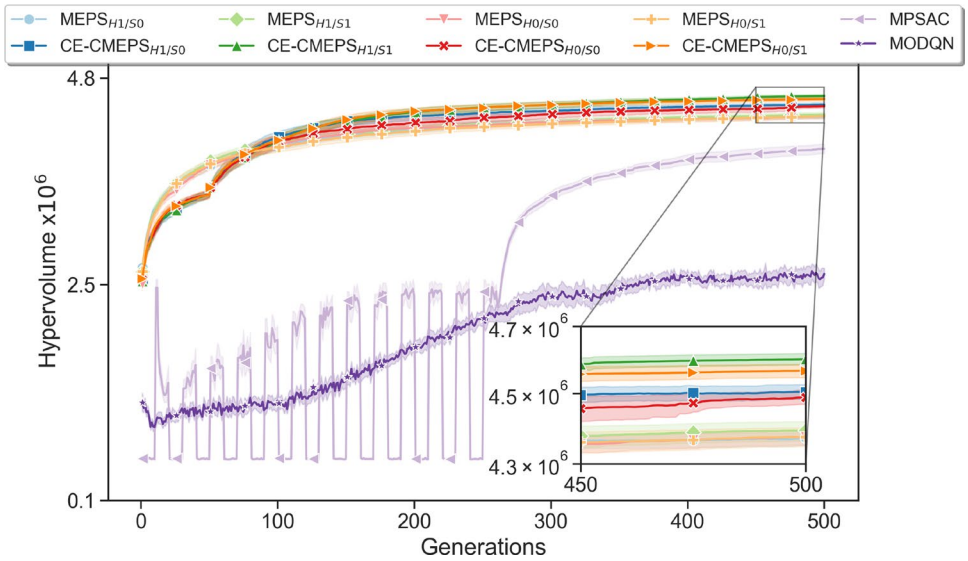


Fig. 7 Mean HV values across 20 runs for each method during the training phase

Table 4 Evaluation of each algorithm's performance in terms of hypervolume within the test load demand scenario

The results in bold mean that after the hypothesis test was performed, the result presented by CE-CMEPS h1/s1 is robust compared to the other methods

	Mean	Std	Worst	Median	Best
MEPS$_{H1/S0}$	4.372E6	0.466E5	4.234E6	4.380E6	4.426E6
CE-CMEPS$_{H1/S0}$	4.504E6	0.514E5	4.415E6	4.503E6	4.597E6
MEPS$_{H1/S1}$	4.393E6	0.656E5	4.194E6	4.410E6	4.463E6
CE-CMEPS$_{\boldsymbol{H1/S1}}$	**4.601E6**	**0.383E5**	**4.548E6**	**4.596E6**	**4.698E6**
MEPS$_{H0/S0}$	4.375E6	0.501E5	4.221E6	4.394E6	4.422E6
CE-CMEPS$_{H0/S0}$	4.488E6	0.468E5	4.424E6	4.479E6	4.624E6
MEPS$_{H0/S1}$	4.375E6	0.616E5	4.277E6	4.390E6	4.462E6
CE-CMEPS$_{H0/S1}$	4.566E6	0.460E5	4.510E6	4.556E6	4.690E6
MPSAC	4.012E6	1.223E5	3.759E6	4.012E6	4.278E6
MODQN	2.619E6	1.438E5	2.401E6	2.614E6	3.062E6

the same as in standard MEPS, $O(t_{max} \cdot m \cdot s_p^2)$ for the version that employs crowding distance and $O(s_p^3 \cdot m^2)$ for the counterpart that employs hypervolume contribution.

Algorithm 1 Multi-Objective CE Method for optimization

Input: Initial mean vector μ_0 and variance vector σ_0^2, sample size N, elite size N_e, smoothing coefficient α, maximum number of generations T, number of variables d
Output: A_t elites

$P_0 \leftarrow$ sample N individuals from multivariate $\mathcal{N}(\mu_0, \sigma_0^2)$;
Evaluate and assign fitness to each individual in P_0;
$t \leftarrow 1$;
while $t < T$ **do**
  $A_t \leftarrow$ Select the N_e individuals from P_{t-1} based on non-dominance sorting and crowding distance [47];
  for $i \leftarrow 1$ *to* d **do**
    $\mu_t^i \leftarrow \sum_{x_j \in |A_t|} x_j^i / |A_t|$;
    $(\sigma_t^i)^2 \leftarrow \sum_{x_j \in |A_t|} (x_j^i - \mu_t^i)^2 / |A_t|$;
  end
  $\mu_t \leftarrow \alpha\mu_t + (1-\alpha)\mu_{t-1}$;
  $\sigma_t \leftarrow \alpha\sigma_t + (1-\alpha)\sigma_{t-1}$ $\Lambda_t \leftarrow$ sample $|P_{t-1}| - |A_t|$ individuals from multivariate $\mathcal{N}(\mu_t, \sigma_t^2)$;
  Evaluate and assign fitness to each individual in Λ_t;
  $P_t \leftarrow \Lambda_t \cup A_t$;
  $t \leftarrow t + 1$;
end

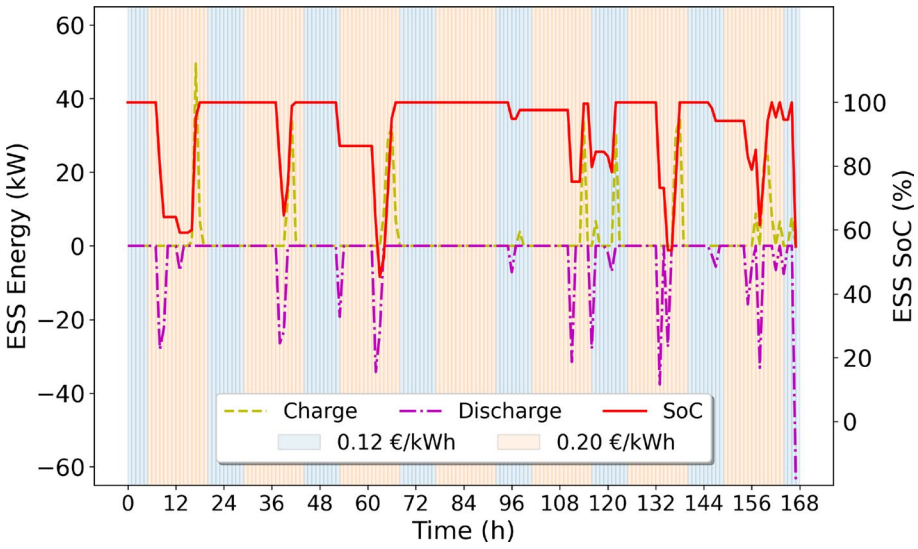


Fig. 8 Behavior of the CE-CMEPS$_{H1/S1}$ solution with best ranking after ranking the combined Pareto set of solutions using the Modified TOPSIS with equal importance for every objective

Algorithm 2 Cross-Entropy based Cascade Multi-Objective Evolutionary Policy Search (CE-CMEPS)

Input: Smoothing coefficient α, s_p, ψ, t_{max}, t_r, t_{CE}, β, σ, p_{ac}, p_{an}, n_i, n_o, n_h, S, HT, H
Output: Memory M

Initialize population P_t with fully connected ANNs containing n_i input nodes, n_h hidden nodes and n_o output nodes;
$t \leftarrow 0$;
Evaluate each individual of P_t for an episode of length h;
Encode each individual of P_t as real-valued vector;
$\mu_0 \leftarrow$ sample mean of P_t real-valued vectors;
$\sigma_0^2 \leftarrow$ sample variance of P_t real-valued vectors;
$d \leftarrow$ number of elements in the real-valued vectors;
$N \leftarrow 2 \cdot s_p$;
$N_e \leftarrow s_p$;
Run Run Algorithm 1for t_{CE} generations evaluating each individual of P_t for an episode of length H;
$t \leftarrow t + t_{CE}$;
Update population P_t with the individuals output from CE;
while $t < t_{max}$ **do**
    Run MEPS[38] main loop using the cascade weights mutation, evaluating each individual of P_t for an episode of length H;
    Update memory M following memory update procedure;
    $t \leftarrow t + 1$;
end

4 Experiments and results

This section presents and analyzes the outcomes of CE-CMEPS in the context of the proposed ESS control problem. The effectiveness of the coupled approach is evaluated against standard MEPS and two standard MORL algorithms: Multi-Policy Soft Actor-Critic (MPSAC) and Multi-Objective Deep Q Networks (MODQN). MPSAC integrates soft-actor critic methods with multi-objective CMA-ES [57], while MODQN utilizes the MORL framework for Deep RL as introduced in [32]. The MODQN architecture comprises two fully connected layers with 64 neurons each, whereas MPSAC employs ANNs with a single hidden layer containing 64 neurons. Both CE-CMEPS and standard MEPS began with an ANN population initialized without hidden layers. For all algorithms, ReLU activation functions [58] were used in the neurons. The input and output layer configurations were the same across all four methods, with 7 and 11 neurons, respectively. The remaining hyperparameters are detailed in Table 3 with CE parameters chosen empirically and MEPS parameters extracted from [59].

Regarding computational complexity, CE-CMEPS incurs an additional cost of $O(s_p^3)$ due to the integration of the Cross-Entropy (CE) method, as compared to CMEPS and MEPS. However, in our experimental setup, we selected parameter values to balance this cost. Specifically, we set $s_p = 50$ and $t_{\max} = 450$ for CE-CMEPS, and $t_{\max} = 500$ for MEPS, assuming a constant m. Under these conditions, both algorithms yield the same numerical time complexity of $O(m \cdot 1,250,000)$. This ensures a fair comparison between methods while keeping computational effort consistent across experiments.

To assess the capacity of each algorithm in generalizing to unseen scenarios, we used various load scenarios for training and testing. Figure 6a shows the test scenario as a dashed line, and the orange region represents the extent of noise included in the process of generating training scenarios. For each algorithm, the ESS started at 100% SoC, and 20 independent runs were performed. The performance of each execution was determined by the average reward obtained from five randomly sampled load scenarios from the green area. The public grid energy purchasing price is composed of day and night prices of 0.12€/kWh and 0.20€/kWh, respectively. Day prices apply from 5 a.m. to 8 p.m. and night prices apply from 8 p.m. to 5 a.m. [43]. After training, each algorithm was tested on the test scenario detailed in Fig. 6b.

The hypervolume indicator (HV) is employed as a measure to assess the quality of the solution set obtained by each method [60]. The HV measure is chosen because it allows for the comparison of different algorithms using a single value, without requiring knowledge of the true Pareto front or its approximation. It is obtained using the accumulated rewards from multiple policies after the predefined number of generations and a reference point. From a decision-maker standpoint, a high HV value translates into better trade-off solutions. The reference point used in this work is equal to [37280, 340, 560].

Figure 7 illustrates the hypervolume of policies averaged across 20 runs for each algorithm. Both MPSAC and MODQN exhibit increasing hypervolume values with each generation during training. However, their performances are consistently below those of MEPS and CE-CMEPS, with MODQN displaying the poorest training performance. Notably, the hypervolume values of CE-CMEPS do not show as much of an increase in the first 50 generations when the CE algorithm is active. However, upon completion of the CE algorithm and the initialization of weights for CMEPS, it takes less than 50 generations for the CE-CMEPS variants to surpass the hypervolume values of standard MEPS variants. This pattern suggests that the CE method can identify a more suitable basin of attraction for CMEPS to operate within.

Following training, the performance of each algorithm was assessed on an unseen test scenario. Table 4 provides details on the average HV values obtained in this test scenario. Similar to their training performance, both MPSAC and MODQN not only exhibited the poorest performance but also demonstrated the highest standard deviations, with MODQN yielding the lowest HV values. Among the MEPS versions, H0/S0 and H1/S0 showed comparable results, with a slight advantage for CE-CMEPS. CE-CMEPS$_{H0/S1}$ and CE-CMEPS$_{H1/S1}$ versions achieved the best performances in terms of HV values. Notably, all CE-CMEPS versions improved upon the performance of their standard MEPS counterparts in terms of HV, with H1/S1 and H0/S1 attaining the highest HV values. Furthermore, apart from CE-CMEPS$_{H1/S0}$, the other CE-CMEPS versions also exhibited reduced standard deviations, maintaining consistent performance across multiple executions. Finally, the CE-CMEPS$_{H1/S1}$ version not only achieved the highest average HV value but also exhibited the lowest standard deviation.

The mean and standard deviation of hypervolume values may serve as introductory indicators but it might not sufficiently capture the nuances of the obtained results. Hence, a statistical protocol, based on methodologies outlined in [6, 61], was adopted. The Kruskal-Wallis test [62] was utilized to identify potential differences among the mean objective function values using 20 runs for each algorithm. Accordingly, a p-value < 0.05 indicates (with a significance level of 95%) the occurrence of differences among the means. Subsequently, the Wilcoxon signed-rank test [63] using the Holm-Bonferroni correction [64] was conducted for pairwise analysis to detect differences between the analyzed samples. The presence of p-values less than 0.05 in the comparisons indicates rejection of the null hypothesis with 95%

significance in all cases. As corroborated by the statistical test results, not only does CE-CMEPS$_{H1/S1}$ exhibit superior performance, but CE-CMEPS also outperforms standard MEPS in the proposed multi-objective ESS control problem.

Subsequently, we analyzed the behavior of the CE-CMEPS$_{H1/S1}$ solution with best ranking after ranking the combined Pareto set of solutions using the Modified Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [65] with equal importance for every objective. Figure 8 shows that the proposed method effectively learned to discharge the ESS at moments when both the public grid electricity price is higher and the amount of renewable energy in its composition is reduced. This solution leads to a total weekly cost of 37161.02€, 313.70kg of CO_2, and an accumulated SoC penalty of 120.19, whereas the best TOPSIS ranking solution from MPSAC yields a total weekly cost of 37178.48€, 317.50kg of CO_2, and an accumulated SoC penalty of 122.00. Therefore, CE-CMEPS$_{H1/S1}$ offers an efficient solution for transitioning to clean energy that not only reduces costs and increases battery life but also saves nearly 4 kg of CO_2 per week compared to an actor-critic-based solution. Additionally, in terms of final network size, CE-CMEPS solution was composed of 59 hidden nodes and 204 connections, while MPSAC's solution consisted in 64 hidden nodes and 1152 connections. Such parameter reduction makes CE-CMEPS a suitable solution for use as a controller in devices with low computing power.

5 Conclusion

This paper proposed a modified version of MEPS and its coupling with the multi-objective Cross-Entropy (CE) method. In this coupled approach, CE acted as a depth initialization strategy performing an initial search in the weights space to initialize the population. The modified version of MEPS, namely Cascade-MEPS (CMEPS), was then initialized using the obtained set of weights. The CMEPS employed a cascade weights mutation operator that restricts mutation to the weights associated with the most recent hidden node, to make local and non-disruptive adjustments to policies. The proposed approach was validated on a newly proposed multi-objective energy storage system (ESS) control problem. The proposed coupled algorithm, namely CE-CMEPS, not only improved the performance of MEPS but also reduced the performance variation. Despite the advantages and enhancements offered by CE-CMEPS, the integration of CE and CMEPS introduces a few additional hyperparameters and adds another layer of complexity to MEPS. Users are required to specify both the number of generations for each algorithm and the CE smoothing coefficient value. Moreover, the initial population is initialized with the same topology, implying that every weight-encoded vector in CE exists in the same dimension. It is also worth noting that, although the integration of the Cross-Entropy (CE) method in CE-CMEPS introduces an additional computational cost, this is not a limiting factor in our application. Since the energy planning problem considered here spans a one-week horizon and does not require real-time decision-making, the longer training time is acceptable. Moreover, the testing phase remains computationally efficient, ensuring the practical applicability of the proposed approach. We suggest future research to explore the utilization of CE for chromosomes of varying lengths. Overall, CE-CMEPS has demonstrated efficacy as an ESS discharge controller and represents a promising solution for sustainability and energy conservation. It outperforms traditional deep-learning MORL methods and yields solutions suitable for integration into embedded control systems.

Acknowledgments This work was partially funded by the Brazilian research agencies: FAPERJ- Fundação Carlos Chagas Filho de Amparo à Pesquisa do Estado do Rio de Janeiro (Grant Numbers E-26/200.840/2021, E-26/210.057/2024, and E-26/210.549/2024), CNPq-National Council for Scientific and Technological Development (Grant Numbers 306258/2019-6, 403964/2023-7, and 309342/2023-6), Ph.D. Scholarship from CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES-PROEX), and support from CAPES (Grant number: 23038.006308/2021-70). UFRJ ALV Grant. This research has also been partially supported by projects PID2023-150663NB-C21 and TED2021-131777B-C22, funded by the Spanish Ministry of Science and Innovation (MICINN); and TEC-2024/ECO-287 funded by Comunidad de Madrid (R&D activities programme "Tecnologías 2024"). The authors thank UFRJ, UAH, and Aston University for the infrastructure

used to conduct this work.

Funding The Article Processing Charge (APC) for the publication of this research was funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) (ROR identifier: 00x0ma614).

Data availability Electrical data for the experiments were used based on [43], available in https://www.kaggle.com/datasets/jonathandumas/liege-microgrid-open-data.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This paper contains no cases of studies with human participants performed by any of the authors.

Informed consent This study does not contain any studies with human participants or animals performed by any of the authors.

References

1. Caduff M, Huijbregts MA, Althaus H-J, Koehler A, Hellweg S (2012) Wind power electricity: the bigger the turbine, the greener the electricity? Environ Sci Technol 46(9):4725–4733
2. UNECE (2022) Life cycle assessment of electricity generation options.https://unece.org/sed/documents/2021/10/reports/life-cycle-assessment-electricity-generation-options. Available 16th March 2022
3. Koohi-Fayegh S, Rosen MA (2020) A review of energy storage types, applications and recent developments. J Energy Stor 27:101047
4. Kebede AA, Kalogiannis T, Van Mierlo J, Berecibar M (2022) A comprehensive review of stationary energy storage devices for large scale renewable energy sources grid integration. Renew Sustain Energy Rev 159:112213
5. Marcelino CG, Leite GMC, Delgado CADM, de Oliveira LB, Wanner EF, Jiménez-Fernández S, Salcedo-Sanz S (2021) An efficient multi-objective evolutionary approach for solving the operation of multi-reservoir system scheduling in hydro-power plants. Expert Syst Appl 185:115638. https://doi.org/10.1016/j.eswa.2021.115638
6. Marcelino CG, Leite GMC, Jiménez-Fernández S, Salcedo-Sanz S (2022) An improved C-DEEPSO algorithm for optimal active-reactive power dispatch in microgrids with electric vehicles. IEEE Access 10:94298–94311. https://doi.org/10.1109/ACCESS.2022.3203728
7. Talaat M, Elkholy M, Alblawi A, Said T (2023) Artificial intelligence applications for microgrids integration and management of hybrid renewable energy sources. Artif Intell Rev 56(9):10557–10611
8. Elkholy M, Metwally H, Farahat M, Nasser M, Senjyu T, Lotfy ME (2022) Dynamic centralized control and intelligent load management system of a remote residential building with V2H technology. J Energy Stor 52:104839
9. Abedi S, Yoon SW, Kwon S (2022) Battery energy storage control using a reinforcement learning approach with cyclic time-dependent Markov process. Int J Electr Power Energy Syst 134:107368
10. Thirunavukkarasu GS, Seyedmahmoudian M, Jamei E, Horan B, Mekhilef S, Stojcevski A (2022) Role of optimization techniques in microgrid energy management systems—a review. Energ Strat Rev 43:100899
11. Leite G, Marcelino C, Pedreira C, Jiménez-Fernández S, Salcedo-Sanz S (2023) Evaluating the risk of uncertainty in smart grids with electric vehicles using an evolutionary swarm-intelligent algorithm. J Clean Prod 401:136775. https://doi.org/10.1016/j.jclepro.2023.136775
12. Zhao F, Di S, Wang L (2022) A hyperheuristic with Q-learning for the multiobjective energy-efficient distributed blocking flow shop scheduling problem. IEEE Trans Cybern 53(5):3337–3350
13. Li K, Zhang T, Wang R (2020) Deep reinforcement learning for multiobjective optimization. IEEE Trans Cybern 51(6):3103–3114

14. Zhang Z, Tang Q, Chica M, Li Z (2023) Reinforcement learning-based multiobjective evolutionary algorithm for mixed-model multimanned assembly line balancing under uncertain demand. IEEE Trans Cybern 54(5):2914–2927
15. Cheng G, Dong L, Yuan X, Sun C (2023) Reinforcement learning-based scheduling of multi-battery energy storage system. J Syst Eng Electron 34(1):117–128
16. Xu G, Shi J, Wu J, Lu C, Wu C, Wang D, Han Z (2024) An optimal solutions-guided deep reinforcement learning approach for online energy storage control. Appl Energy 361:122915
17. Panda DK, Turner O, Das S, Abusara M (2024) Prioritized experience replay based deep distributional reinforcement learning for battery operation in microgrids. J Clean Prod 434:139947
18. Kolodziejczyk W, Zoltowska I, Cichosz P (2021) Real-time energy purchase optimization for a storage-integrated photovoltaic system by deep reinforcement learning. Control Eng Pract 106:104598
19. Zhou K, Zhou K, Yang S (2022) Reinforcement learning-based scheduling strategy for energy storage in microgrid. J Energy Stor 51:104379
20. Zhu L, Wei Q, Guo P (2024) Synergetic learning neuro-control for unknown affine nonlinear systems with asymptotic stability guarantees. IEEE Trans Neural Netw Learn Syst 36:3479
21. Guo L, Zhao H (2023) Online adaptive optimal control algorithm based on synchronous integral reinforcement learning with explorations. Neurocomputing 520:250–261
22. Dong C, Chen L, Dai S-L (2023) Performance-guaranteed adaptive optimized control of intelligent surface vehicle using reinforcement learning. IEEE Trans Intell Veh 9(2):3581–3592
23. Schmidt-Hieber J (2020) Nonparametric regression using deep neural networks with ReLU activation function. Ann Stat 48(4):1875–1897
24. Lyu B, Wen S, Shi K, Huang T (2021) Multiobjective reinforcement learning-based neural architecture search for efficient portrait parsing. IEEE Trans Cybern 53(2):1158–1169
25. Yarotsky D (2017) Error bounds for approximations with deep ReLU networks. Neural Netw 94:103–114
26. Frankle J, Carbin M (2018) The lottery ticket hypothesis: finding sparse, trainable neural networks, arXiv preprint arXiv :1803.03635
27. Bartlett PL, Harvey N, Liaw C, Mehrabian A (2019) Nearly-tight VC-dimension and pseudodimension bounds for piecewise linear neural networks. J Mach Learn Res 20(1):2285–2301
28. Vapnik VN, Chervonenkis AY (2015) On the uniform convergence of relative frequencies of events to their probabilities, In: Measures of complexity: festschrift for alexey chervonenkis, Springer, pp. 11–30
29. Coello CAC, Lamont GB, Van Veldhuizen DA et al (2007) Evolutionary algorithms for solving multi-objective problems, vol 5. Springer, Berlin
30. Hayes CF, Rădulescu R, Bargiacchi E, Källström J, Macfarlane M, Reymond M, Verstraeten T, Zintgraf LM, Dazeley R, Heintz F, Howley E, Irissappane AA, Mannion P, Nowé A, Ramos G, Restelli M, Vamplew P, Roijers DM (2022) A practical guide to multi-objective reinforcement learning and planning. Auton Agent Multi-Agent Syst 36(1):1–59. https ://doi.org/10.1007/s10458-022-09552-y
31. Cao Y, Zhan H (2021) Efficient multi-objective reinforcement learning via multiple-gradient descent with iteratively discovered weight-vector sets. J Artif Intell Res 70:319–349. https://doi.org/10.1613/jair.1.12270
32. Nguyen TT, Nguyen ND, Vamplew P, Nahavandi S, Dazeley R, Lim CP (2020) A multi-objective deep reinforcement learning framework. Eng Appl Artif Intell 96:103915. https://doi.org/10.1016/j.engappai.2020.103915
33. Wei L, Chen Y, Chen M, Chen Y (2021) Deep reinforcement learning and parameter transfer based approach for the multi-objective agile earth observation satellite scheduling problem. Appl Soft Comput 110:107607. https://doi.org/10.1016/j.asoc.2021.107607
34. Wang H, Cheng J, Liu C, Zhang Y, Hu S, Chen L (2022) Multi-objective reinforcement learning framework for dynamic flexible job shop scheduling problem with uncertain events. Appl Soft Comput 131:109717. https://doi.org/10.1016/j.asoc.2022.109717
35. Li Y, Wang R, Yang Z (2021) Optimal scheduling of isolated microgrids using automated reinforcement learning-based multi-period forecasting. IEEE Trans Sustain Energy 13(1):159–169. https://doi.org/10.1109/TSTE.2021.3105529
36. Zaniolo M, Giuliani M, Castelletti A (2021) Neuro-evolutionary direct policy search for multiobjective optimal control. IEEE Trans Neural Netw Learn Syst 33(10):5926–5938
37. Wang Z, Yao S, Li G, Zhang Q (2023) Multiobjective combinatorial optimization using a single deep reinforcement learning model. IEEE Trans Cybern 54:1984
38. Leite G, Jiménez-Fernández S, Salcedo-Sanz S, Marcelino C, Pedreira C (2023) Solving an energy resource management problem with a novel multi-objective evolutionary reinforcement learning method. Knowl-Based Syst 280:111027
39. Marcelino CG, Leite GMC, Wanner EF, Jiménez-Fernández S, Salcedo-Sanz S (2023) Evaluating the use of a Net-Metering mechanism in microgrids to reduce power generation costs with a swarm-intelligent algorithm. Energy 266:126317. https://doi.org/10.1016/j.energy.2022.126317
40. Canadian Solar HiKu, https://www.csisolar.com/au/hiku/ Available in 11th july 2023 (2023)
41. Norvento (2023) https://www.norvento.com/productos/aerogeneradores-de-media-potencia/ Available in 11th July 2023

42. Liu F, Liu Q, Tao Q, Huang Y, Li D, Sidorov D (2023) Deep reinforcement learning based energy storage management strategy considering prediction intervals of wind power. Int J Electr Power Energy Syst 145:108608
43. Dumas J, Dakir S, Liu C, Cornélusse B (2021) Coordination of operational planning and real-time optimization in microgrids. Electr Power Syst Res 190:106634
44. Marcelino C, Baumann M, Carvalho L, Chibeles-Martins N, Weil M, Almeida P, Wanner E (2020) A combined optimisation and decision-making approach for battery-supported HMGS. J Oper Res Soc 71(5):762–774. https://doi.org/10.1080/01605682.2019.1582590
45. Fraunhofer I (2015) Energy charts, Fraunhofer Institute for Solar Energy Systems ISE. Available online: https://www.energy-charts.de. Accessed 13 Mar 2024
46. Stanley KO, Miikkulainen R (2002) Evolving neural networks through augmenting topologies. Evol Comput 10(2):99–127. https://doi.org/10.1162/106365602320169811
47. Deb K, Pratap A, Agarwal S, Meyarivan T (2002) A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Trans Evol Comput 6(2):182–197. https://doi.org/10.1109/4235.996017
48. Emmerich M, Beume N, Naujoks B (2005) An EMO algorithm using the hypervolume measure as selection criterion, In: International conference on evolutionary multi-criterion optimization, pp. 62–76. https://doi.org/10.1007/978-3-540-31880-4_5
49. Yue C, Suganthan PN, Liang J, Qu B, Yu K, Zhu Y, Yan L (2021) Differential evolution using improved crowding distance for multimodal multiobjective optimization. Swarm Evol Comput 62:100849. https://doi.org/10.1016/j.swevo.2021.100849
50. Arnold BC (2014) Pareto distribution. Statistics Reference Online, Wiley StatsRef, pp 1–10. https://doi.org/10.1002/9781118445112.stat01100.pub2
51. Kohl N, Miikkulainen R (2009) Evolving neural networks for strategic decision-making problems. Neural Netw 22(3):326–337
52. Kohl N, Miikkulainen R (2012) An integrated neuroevolutionary approach to reactive control and high-level strategy. IEEE Trans Evol Comput 16(4):472–488
53. Rubinstein RY (1997) Optimization of computer simulation models with rare events. Eur J Oper Res 99(1):89–112
54. Tang Q, Ma L, Zhao D, Lei J, Wang Y (2022) A multi-objective cross-entropy optimization algorithm and its application in high-speed train lateral control. Appl Soft Comput 115:108151
55. Marcelino CG, Pérez-Aracil J, Wanner EF, Jiménez-Fernández S, Leite GMC, Salcedo-Sanz S (2023) Cross-entropy boosted CRO-SL for optimal power flow in smart grids. Soft Comput 27(10):6549–6572
56. Stein G, Gonzalez AJ, Barham C (2015) Combining NEAT and PSO for learning tactical human behavior. Neural Comput Appl 26:747–764
57. Chen D, Wang Y, Gao W (2020) Combining a gradient-based method and an evolution strategy for multi-objective reinforcement learning. Appl Intell 50(10):3301–3317. https://doi.org/10.1007/s10489-020-01702-7
58. Goodfellow I, Bengio Y, Courville A (2016) Deep learning. MIT Press, Cambridge
59. Leite G, Jiménez-Fernández S, Salcedo-Sanz S, Marcelino C, Pedreira C (2023) Solving an energy resource management problem with a novel multi-objective evolutionary reinforcement learning method. Knowl Based Syst 280:111027. https://doi.org/10.1016/j.knosys.2023.111027
60. Vamplew P, Dazeley R, Berry A, Issabekov R, Dekker E (2011) Empirical evaluation methods for multiobjective reinforcement learning algorithms. Mach Learn 84(1):51–80. https://doi.org/10.1007/s10994-010-5232-5
61. Marcelino CG, Almeida PEM, Wanner EF, Baumann M, Weil M, Carvalho LM, Miranda V (2018) Solving security constrained optimal power flow problems: a hybrid evolutionary approach. Appl Intell 48:3672–3690. https://doi.org/10.1007/s10489-018-1167-5
62. Kruskal WH, Wallis WA (1952) Use of ranks in one-criterion variance analysis. J Am Stat Assoc 47(260):583–621
63. Conover WJ (1999) Practical nonparametric statistics, vol 350. Wiley, Hoboken
64. Holm S (1979) A simple sequentially rejective multiple test procedure. Scand J Stat 6:65–70 (http://www.jstor.org/stable/4615733)
65. Chakraborty S (2022) TOPSIS and modified TOPSIS: a comparative analysis. Decis Anal J 2:100021

Authors and Affiliations

Gabriel Matos Cardoso Leite[1,2] · Carolina Gil Marcelino[3] · Silvia Jiménez-Fernández[2] · Elizabeth Fialho Wanner[4] · Sancho Salcedo-Sanz[2] · Carlos Eduardo Pedreira[1]

✉ Carolina Gil Marcelino
carolina@ic.ufrj.br

Gabriel Matos Cardoso Leite
gmatos@cos.ufrj.br

[1] Systems and Computing Department, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

[2] Signal Processing and Communications Department, Universidad de Alcala (UAH), Madrid, Spain

[3] Institute of Computing, Federal University of Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil

[4] School of Engineering and Applied Science, Aston University, Birmingham, UK