POLITECNICO DI TORINO
Repository ISTITUZIONALE

Optimal Cogeneration Scheduling: A Comparison of Genetic and POMDP-Based Deep Reinforcement Learning Approaches

Original
Optimal Cogeneration Scheduling: A Comparison of Genetic and POMDP-Based Deep Reinforcement Learning Approaches / Ghione, G., Randazzo, V., Pasero, E., Badami, M.. - In: IEEE ACCESS. - ISSN 2169-3536. - ELETTRONICO. - 13:(2025), pp. 128562-128581. [10.1109/access.2025.3590255]

Availability:
This version is available at: 11583/3002153 since: 2025-07-28T09:00:16Z

Publisher:
IEEE

Published
DOI:10.1109/access.2025.3590255

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright

(Article begins on next page)

18 November 2026

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 15 April 2025, accepted 28 June 2025, date of publication 17 July 2025, date of current version 25 July 2025.

Digital Object Identifier 10.1109/ACCESS.2025.3590255

RESEARCH ARTICLE

Optimal Cogeneration Scheduling: A Comparison of Genetic and POMDP-Based Deep Reinforcement Learning Approaches

GIORGIA GHIONE[1], (Graduate Student Member, IEEE), VINCENZO RANDAZZO[1], (Member, IEEE), EROS PASERO[1], (Senior Member, IEEE), AND MARCO BADAMI[2]

[1]Department of Electronics and Telecommunications, Politecnico di Torino, 10129 Turin, Italy
[2]Department of Energy, Politecnico di Torino, 10129 Turin, Italy

Corresponding authors: Giorgia Ghione (giorgia.ghione@polito.it) and Vincenzo Randazzo (vincenzo.randazzo@polito.it)

The work of Giorgia Ghione was supported in part by the Programma Nazionale Ripresa e Resilienza - Next Generation EU (PNRR-NGEU) from the Italian Ministry of University and Research (MUR) under Grant 38-033-32-DOT1332092-3353 (DM 352/2022) and in part by Trigenia S.r.l. The work of Vincenzo Randazzo was supported within the Programma Operativo Nazionale (PON) Ricerca e Innovazione of the Italian Ministry of University and Research (MUR) under the Contract 32-G-13427-2 (DM 1062/2021).

ABSTRACT Large processing facilities require multiple types of energy, such as electrical and thermal (hot water or steam). Cogeneration, or Combined Heat and Power (CHP), can provide significant economic and energy savings. However, scheduling its operation in real-time is challenging. This work compares deep reinforcement learning (DRL) and genetic algorithm (GA) approaches to control a real CHP in a processing facility. Traditionally, the CHP economic dispatch problem is modelled as a Markov Decision Process (MDP) with the assumption of complete observability. Due to the uncertainty of future electric and thermal demands, this assumption is unrealistic in real-world scenarios. Thus, this work proposes using a partially observable MDP (POMDP) for hourly CHP dispatch scheduling to address this partial observability. The selected DRL algorithms are Deep Q Network (DQN), Deep Deterministic Policy Gradient (DDPG), and Soft Actor-Critic (SAC), along with six GA variants. Performance was evaluated using multiple economic metrics, including Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA), an environmental analysis, and a sensitivity analysis under variable electric pricing. This work shows that POMDP effectively models the hourly dispatch scheduling problem of CHPs. The insights gained from this analysis offer multiple potential avenues for future research, including the development more advanced DRL algorithms for CHP economic dispatch and the evaluation of their resilience when inaccurate measurements and anomalous conditions occur.

INDEX TERMS CHP, cogeneration, deep reinforcement learning, EBITDA, economic dispatch, energy, genetic algorithm, partially observable Markov decision process.

I. INTRODUCTION

Large processing facilities typically need several types of energy, such as electrical, mechanical, or thermal energy (in the form of hot water or steam). Frequently, these energy outputs originate from diverse energy resources like steam or gas turbine generators, internal combustion engines (ICEs) and boilers. Cogeneration, also known as Combined Heat and Power (CHP), is widely utilized in industrial and residential settings [2]. It involves the simultaneous generation of thermal and electrical/mechanical energy from one primary energy source [3]. The national power grid serves as an additional electricity source for industries in case their plants do not generate sufficient energy. Conversely, it functions as a recipient for surplus electricity produced, when needed. The range of interdependent energy sources, coupled with

The associate editor coordinating the review of this manuscript and approving it for publication was Wencong Su.

© 2025 The Authors. This work is licensed under a Creative Commons Attribution 4.0 License.
For more information, see https://creativecommons.org/licenses/by/4.0/

the evolving technical and economic landscape, makes the development of effective strategies for reducing energy costs complex [4], given that they represent a primary component of the overall cost of owning and operating a processing facility. Cogeneration has the potential to provide significant savings, both in terms of economics and energy [5], and it can improve the flexibility of the overall power system [6]. The feasibility, profitability, and sustainability of CHP systems in manufacturing facilities have been widely studied [7], [8], [9], [10]. Nevertheless, the scheduling of their operation is still a challenging task nowadays [11], [12]: in particular, dynamic uncertainties in the energy demands, caused by human activities and weather conditions, make the optimization of CHP dispatch scheduling complex.

Many studies exist on the optimization of the dispatch strategy of CHP systems, seeking to minimize total production costs, maximize operating income or minimize carbon emissions while ensuring all constraints are met: Reinforcement Learning (RL) has emerged as a technology that is significantly impacting this field. Deep Reinforcement Learning (DRL) is a subcategory of machine learning which combines RL and deep learning. In these studies, the control problem is modelled as a Markov Decision Process (MDP), assuming the complete observability [13] of the system: however, this assumption rarely holds in real-world environments, which are subject to uncertainties and errors especially regarding energy demands. In the broader context of microgrids, multiple studies handled these challenges by describing the energy management optimization problem as a Partially Observable Markov Decision Process (POMDP), enhancing the robustness and adaptability of decision-making strategies. For example, [14] presented a DRL-based finite-horizon POMDP approach for energy dispatch in IoT-enabled smart isolated microgrids; [15] proposed a bi-level cooperative RL-based framework to address adaptive decision-making under incomplete information in distribution systems with multiple microgrids; in [16] a multiagent Bayesian DRL approach was introduced for microgrid energy management under communication failures; in [17] the residential energy trading and demand-side management problem was modelled as a POMDP and a DRL approach was adopted; [18] proposed a distributed energy management approach for multimicrogrids, formulating the problem as a decentralized POMDP; [19] adopted a POMDP approach to model customer response uncertainties for the optimal dispatch of residential distributed energy resources. However, the POMDP framework has not yet been explored in the field of optimal CHP dispatch scheduling to the best of authors' knowledge.

The contributions of this work are the following.

- This work proposes the novel application of the POMDP framework to the hourly CHP control problem based on the energy demands of a processing facility, and multiple DRL and genetic algorithm (GA) approaches are compared.
- The case study of this work involves the cogeneration unit of a factory situated in Italy: the real energy demands of the factory of the year 2021 were collected for this work.
- The DRL models compared in this work are Deep Q Network (DQN) [20], Deep Deterministic Policy Gradient (DDPG) [21] and Soft Actor-Critic (SAC) [22] and the influence of the size of the POMDP time horizon was assessed for each of them. Six different variants of the GA were tested. Multiple deterministic strategies were taken as a benchmark to evaluate the results.
- The performance of the algorithms was evaluated via multiple economic metrics, including the Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA). Also, an environmental analysis and a sensitivity analysis under variable electric energy pricing conditions were performed.

The remainder of this paper is organized as follows. In Section II a comprehensive review of the literature regarding the optimization of CHP dispatch strategies is presented. In Section III the main characteristics of the case study, the objective function and the constraints of the CHP scheduling problem are described, followed by an explanation of the GA and DRL methods utilized and the POMDP approach adopted. Section IV includes a description of the dataset which was employed for training and testing purposes, the baselines for evaluation, and, finally, a discussion of the results of each experiment. Lastly, the conclusion on this work is drawn and future works are explained in Section V.

II. LITERATURE REVIEW

Currently, various methods are employed for the control of energy systems, typically characterized by non-linearity, multi-modal objective functions, and a mix of discrete and continuous variables. Two main categories can be found: model-based control strategies and data-driven models.

Model-based control techniques leverage a mathematical model of physical phenomena to develop an effective control procedure [23]. On the other hand, data-driven models rely on insights gained from data which are processed in either offline or online mode, rather than relying on information derived from a mathematical model [24]: some methods which fall into this category are Reinforcement Learning (RL) [25] and evolutionary algorithms [26].

Table 1 provides a comprehensive overview of the methodologies utilized for the optimization of Combined Heat and Power (CHP) dispatch strategies. The studies were classified based on multiple aspects: the technologies in the energy system, the optimization or control methods employed, the type of energy demand data used (real-world or from simulations), whether variable energy prices (e.g., electricity prices) were considered, whether comparative analyses were conducted against other methods, whether uncertainty handling or partial observability (PO) were

addressed in the study and, lastly, whether a Partially Observable Markov Decision Process (POMDP) was used in modelling.

The reviewed studies encompass a wide range of technologies including CHP systems, renewable sources, energy storage solutions, and heating technologies such as gas boilers and electric boilers. As regards the methodologies utilized, the studies employ various optimization techniques such as SAC, PID, DDPG, GA, among others, leveraging real-world or simulation-based data. Variable energy prices are considered in many studies, emphasizing the importance of economic feasibility. Additionally, comparative analyses against alternative methodologies emerge as a necessary step in the validation of proposed strategies. The literature shows that a promising research avenue is the development of methods addressing the inherent complexities and uncertainties in energy system operations: uncertainty handling techniques were incorporated in several studies. However, literature lacks in the application of the POMDP framework in RL solutions for CHP dispatch scheduling. Indeed, as detailed in Section III-D1, real case applications cannot be fully modelled using MDP; therefore, this paper tries to overcome the literature limitations by applying POMDP.

This work aims to contribute to bridging these gaps by comparing RL, evolutionary and deterministic approaches for the optimization of the dispatch scheduling of a CHP in a real-world scenario, assessing the performance of POMDP against the traditional MDP framework. Furthermore, a sensitivity analysis is performed considering real variable energy prices. The basic approach was presented in [1] by the authors. Here, the framework is refined and extended by means of additional theoretical analysis of the working principles of the Reinforcement learning and Genetic Algorithm. Also, the proposed approach has been compared to three traditional CHP scheduling benchmarks, such as electric load following, three versions of the Genetic algorithm, and two additional deep reinforcement learning algorithms, namely DDPG and SAC.

III. METHODOLOGY

This section describes the case study, as well as the CHP scheduling problem and the methods utilized.

A. DESCRIPTION OF THE CASE STUDY

A real thermal power plant (TPP) in the Lombardy region of Italy is the case study of this work. The plant consists of a cogenerator powered by an internal combustion engine (ICE), which provides heat and power to a factory that produces adhesive solutions.

Table 2 summarizes the main characteristics of the presented CHP and Fig. 1 shows a scheme of the real system.

The factory presents a demand for both electric power and heat in the form of steam, high-temperature hot water, and low-temperature hot water, which will be aggregated into a single thermal demand term. Fig. 2 represents the duration curve of the electric and thermal load during the year 2021.

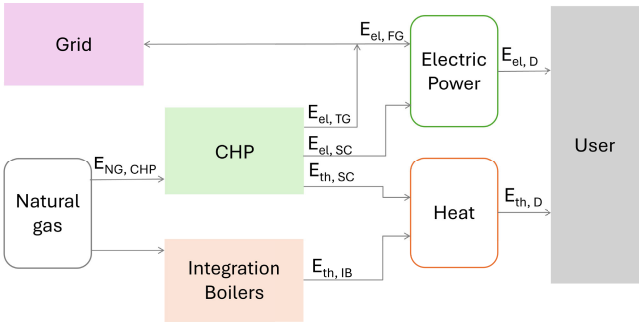


FIGURE 1. Scheme of the energy production problem.

The cogeneration plant is made of a single cogeneration unit, whereas the whole TPP is made of three boilers and a reciprocating engine. The combined production of heat and electricity is made using a reciprocating engine. The cogeneration module functions with a four-stroke Otto cycle, utilizes natural gas as fuel and is also connected in parallel to the power supply network of the national power grid, with a voltage amounting to 15 kV.

In the cogeneration unit, the heat recovery starts inside the engine block (including a lubricating oil circuit, an engine jacket water circuit, and an intercooler first stage circuit). Approximately 774 kW can be recovered from it, according to the design specifications. This power recovery, when supplied together with the 97 kW provided by the preheating coil, allows the generation of hot water at approximately 85 °C for the operational needs of the facility. The exhaust gases yielded by the engine block are conveyed to a shell-and-tube heat exchanger able to produce 396 kW (considering also the 41 kW provided by the economizer) as saturated steam at approximately 175 °C. This steam is completely self-utilized by the facility. Lastly, the heat of the second intercooler stage, which accounts for 118 kW, is recovered through an additional exchanger as low-temperature hot water at around 29.5 °C. This thermal output is self-utilized by the facility as well.

The cogenerator can only partially cover the electric power and heat demands of the factory (Fig. 2). The remaining heat demand is supplied by integration boilers with efficiency η_{IB} equal to 0.84, which is the experimental mean efficiency value of the currently installed boilers in the TPP. The remaining electric power demand is covered by purchasing power from the national power grid.

The heat and electricity produced by the cogenerator are almost completely self-utilized, thus the CHP plant can be considered a highly efficient one, by the European Parliament directive 2012/27/EU [47]. Given this consideration, it can be also assumed that the CHP plant has dispatching priority on the national power grid [47].

At present, plant technicians manually control the CHP based on their experience. This control is operated by hourly modifying the load of the CHP, which is denoted by the parameter α, called load coefficient. The parameter α is

TABLE 1. Literature review.

Article	Technology	Methods	Data	Variable prices	Comparative analysis	Uncertainty/PO	POMDP
Alabi et al., 2024 [27]	CHP, AC, HP, GB, EC, WT, PV, BES, TES	SAC	Real	Yes	-	Yes	-
Dong et al., 2021 [28]	CHP, EB, PV, WT, BES, TES, GS, PtG	A3C	Real	Yes	-	-	-
Gao & Lin, 2023 [29]	CCHP	DDQN	Real	Yes	-	Yes	-
Ginidi et al., 2021 [30]	CHP	HBOA	Simulation	-	Yes	-	n.a.
Hu et al., 2024 [31]	CHP, GB, EB, BES, TES	MINLP	Simulation	Yes	-	-	n.a.
Jia et al., 2024 [32]	CHP, BES, HP, TES, AC	TD3	Real	Yes	-	Yes	-
Kim et al., 2020 [33]	CHP	GA, FFNN	Real	Yes	-	-	n.a.
Liu et al., 2020 [34]	CHP, WT, CFTPU, EB, TES	MIQP	Simulation	-	-	-	n.a.
Lorestani & Ardehali, 2018 [35]	CHP, PVT, PV, WT, BES, TES, EH	PSO	Simulation	-	Yes	-	n.a.
Moretti et al., 2020 [36]	CHP, PV, WT, BES, GB, HP	RO	Real, Simulation	Yes	-	Yes	n.a.
Moustafa et al., 2024 [37]	CHP	MSA	Simulation	-	Yes	-	n.a.
Qiu et al., 2022 [38]	CHP, PV, WT, TES, HSS, HP, GB	DDPG	Real	Yes	Yes	Yes	-
Ruan et al., 2023 [39]	CCHP, PV, BES, TES	DDPG, TD3	Simulation	Yes	Yes	Yes	-
Sundaram, 2020 [40]	CHP	NSGA II-MOPSO	Simulation	-	Yes	-	n.a.
Tang, 2019 [41]	CHP, GT, GB, BES, TES, EH, PV, WT	MILP	Real	-	-	Yes	n.a.
Wu et al., 2024 [42]	CHP, HSS, HFC, GB, MR	Adaptive closed-loop control	Simulation	-	-	-	n.a.
Zhang et al., 2021 [43]	CHP, WT, HP, GT, BES	SAC	Simulation	-	Yes	Yes	-
Zhou et al., 2020 [44]	CHP, WT, TES	DPPO	Simulation	Yes	-	Yes	-
Zhou et al., 2022 [45]	CHP, PV, WT, BES, GB, HP	SAC	Real	Yes	Yes	Yes	-
Żymełka & Szega, 2021 [46]	CHP	GA	Real	Yes	-	-	n.a.
This work	*CHP, GB*	*SAC, DDPG, DQN, GA*	*Real*	*Yes*	*Yes*	*Yes*	*Yes*

Table 1 abbreviations: Absorption chiller: AC, Battery electric energy storage: BES, Coal-fired thermal power unit: CFTPU, Combined cooling, heating, and power: CCHP, Electric boiler: EB, Electric chiller: EC, Electric heater: EH, Feed-forward artificial neural network: FFNN, Genetic algorithm: GA, Gas boiler: GB, Gas turbine: GT, Gas storage: GS, Heap-based optimization algorithm: HBOA, Hydrogen fuel cell: HFC, Heat only: HO, Hydrogen storage system: HSS, Heat pump: HP, Manta Ray Foraging Optimization Algorithm: MRFOA, Mantis Search Algorithm: MSA, Methane reactor: MR, Multi Objective Particle Swarm Optimization: MOPSO, Nondominated Sorting Genetic Algorithm II: NSGA II, Particle swarm optimization: PSO, Power only: PO, Power to gas: PtG, Photovoltaic: PV, Photovoltaic-thermal panels: PVT, Reinforcement learning: RL, Soft actor-critic: SAC, Thermal energy storage: TES, Wind turbine: WT, Genetic algorithm: GA, Deep Q-network: DQN, Deep deterministic policy gradient: DDPG, Twin delayed deep deterministic policy gradient: TD3, Advantage actor-critic: A3C, Deep Q-network with experience replay: DDQN, Mixed-integer linear programming: MILP, Mixed-Integer Nonlinear Programming: MINLP, Mixed-integer quadratic programming: MIQP, Proximal policy optimization: PPO, Dual proximal policy optimization: DPPO, Model predictive control: MPC.

defined in Equation 1:

$$\alpha = \frac{P_{el,CHP}}{P_{el_{nom}}} \tag{1}$$

where $P_{el,CHP}$ is the electric power to be generated by the cogenerator and $P_{el_{nom}}$ is defined in Table 2. Thus, α represents the fraction of the rated electric power to be produced. At each time step t, the load coefficient can be increased or decreased by a value $\Delta\alpha_t$, as shown in Equation 2.

$$\alpha_t = \alpha_{t-1} + \Delta\alpha_t \tag{2}$$

The load coefficient can be set to 0 (the CHP is turned off) or to any number between 0.5 (half load) and 1 (full load): indeed, it is not possible to set α to values lower than

TABLE 2. Summary of the technical characteristics of the CHP.

Characteristic	Value
Rated full-load electric power output ($P_{el_{nom}}$) [kW]	1203
Rated full-load thermal power output ($P_{th_{nom}}$) [kW]	1385
Type of cycle	Four-stroke Otto cycle
Fuel type	Natural gas
Nominal electric efficiency [%]	40.2
Nominal maximum thermal efficiency [%]	47.8

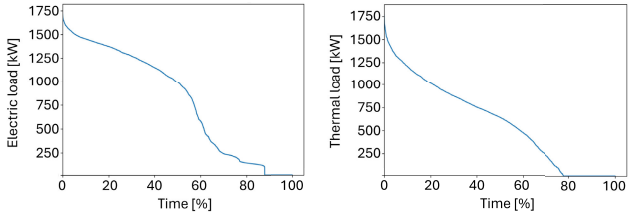


FIGURE 2. (a) Duration curve of the electric load during the year 2021. (b) Duration curve of the thermal load during the year 2021.

0.5 besides 0. This constraint is due to the minimum and maximum modulation thresholds defined for the CHP: the minimum modulation threshold is equal to half $P_{el_{nom}}$, and the maximum modulation threshold is equal to $P_{el_{nom}}$. This work focuses on the optimization of the scheduling of the load coefficient α via the hourly control of $\Delta\alpha_t$: the optimization concept presented in this work is graphically shown in Fig. 3, where the CHP block is as Fig. 1.

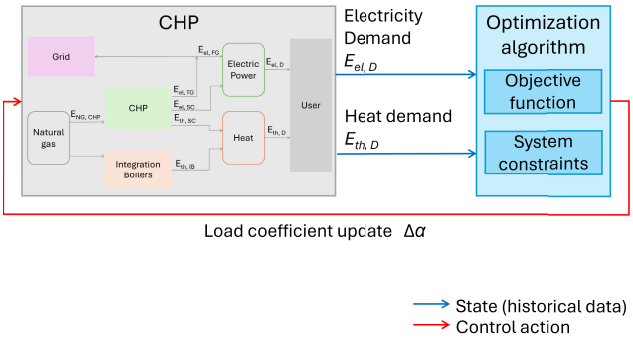


FIGURE 3. Scheme of the optimization concept. CHP block is as Fig. 1.

B. OBJECTIVE FUNCTION AND CONSTRAINTS

The Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA) metric [48] is the objective function to be maximized. The formulation of the hourly EBITDA is presented in Equation 3.

$$EBITDA = R + AC - C \tag{3}$$

All incentive criteria outlined in the Italian legislation pertaining to the primary energy-saving directive have been considered: the computation of Energy Efficiency Credits (EECs), which are documents provided by the Italian state to certify a specific reduction in energy consumption, was included. In particular, the EBITDA is calculated as the sum of revenues R (due to EECs and the revenue derived from selling excess electricity back to the grid) and the avoided costs AC from using the CHP (i.e. the savings from avoiding the cost of natural gas utilized for heat production and the cost of electricity self-consumed), from which the costs of the plant C were subtracted (considering the procurement of natural gas required to fuel the cogenerator, and operations and maintenance, i.e. O&M, costs). The heat self-consumption was factored into the EBITDA formulation in accordance with the European Parliament directive 2012/27/EU [47] regarding high-efficiency cogeneration, which pertains to the primary energy savings calculation. Since the main interest of this work is the economic return, the possible waste of thermal energy caused by the electric load following mode was not included as a cost in the EBITDA.

TABLE 3. Summary of economic data.

Notation	Description	Value
r_{EEC}	Price of EECs [€/EEC]	260[a]
r_{el}	Revenue per kilowatt-hour (kWh) from selling electric energy to the grid [€/kWh]	0.05[a]
c_{el}	Per-kWh cost for the purchase of electric energy from the grid [€/kWh]	0.15[a]
c_{NG}	Cost of natural gas [€/kWh]	0.041[a]
$c_{O\&M}$	Cost of operation and maintenance [€/h]	14[a]

[a] Mean value in Italy before the 2021 global energy crisis.

The detailed formulation of the EBITDA at each time t is the following:

$$\begin{aligned} EBITDA_t = {} & r_{EEC} \times EEC_t + r_{el} \times E_{el,TG,t} \\ & + c_{NG} \times \frac{E_{th,SC,t}}{\eta_{IB}} + c_{el} \times E_{el,SC,t} \\ & - c_{NG} \times E_{NG,CHP,t} - c_{O\&M} \times H_t \end{aligned} \tag{4}$$

where $E_{el,SC,t}$ is the self-consumed electric energy in the 1-hour interval Δt, equal to the electric energy demand; $E_{th,SC,t}$ is the self-consumed heat in the 1-hour interval Δt, equal to the heat demand; η_{IB} is the efficiency of integration boilers defined in Section III-A; the variable H_t represents the amount of operating hours of the cogenerator. The cost terms $c_{NG}, c_{el}, r_{el}, r_{EEC}$, and $c_{O\&M}$ are reported in Table 3: the mean values in Italy before the 2021 global energy crisis were utilized because this crisis introduced significant volatility and anomalies that could skew the results. The other variables are defined in Equations 5-7.

$$\begin{aligned} EEC_t = {} & k_{conv} \times k_{harmon} \\ & \times \left(\frac{E_{el,CHP,t}}{\eta_{el_{ref}}} + \frac{E_{th,SC,t}}{\eta_{th_{ref}}} - E_{NG,CHP,t} \right) \end{aligned} \tag{5}$$

$$E_{el,TG,t} = E_{el,CHP,t} - E_{el,SC,t} \tag{6}$$

$$E_{NG,CHP,t} = \frac{E_{el,CHP,t}}{\eta_{el,t}} \tag{7}$$

TABLE 4. Summary of parameters for EEC calculation.

Notation	Description	Value
k_{harmon}	Harmonization coefficient	1.4
k_{conv}	Conversion coefficient from MWh to tonnes of oil equivalent (toe)	0.086
$\eta_{el_{ref}}$	Conventional average efficiency of the Italian electric power generation fleet (between 50% and 100% of $P_{el_{nom}}$, defined in Table 2)	0.46
$\eta_{th_{ref}}$	Conventional average efficiency of the Italian thermal generation fleet	0.90

with

$$E_{el,CHP,t} = \alpha_t \times P_{el_{nom}} \times H_t \tag{8}$$

$$\eta_{el,t} = a + b \times P_{el,CHP,t} \tag{9}$$

Equation 5 provides the EEC calculation mandated by Italian legislation: here, EEC_t denotes the number of EECs assigned for the operation of the CHP; the parameters k_{conv}, k_{harmon}, $\eta_{el_{ref}}$, and $\eta_{th_{ref}}$ are defined in Table 4 ; $E_{el,CHP,t}$ is the total electric energy output of the cogenerator calculated in Equation 8, where α_t and $P_{el_{nom}}$ are defined in Equation 2 and Table 2, respectively. In Equation 6, $E_{el,TG,t}$ is the electric energy sold to the grid. In Equation 7, $E_{NG,CHP,t}$ is the primary energy input and the term $\eta_{el,t}$ is the electric efficiency. Since the electric power generation of the CHP can be regulated between 50% (α equal to 0.50) and 100% (α equal to 1) of the rated full-load electric power output, as described in Section III-A, the electric efficiency $\eta_{el,t}$ was simulated with a linear regression computed over real experimental data of the plant: the curve that was selected is reported in Equation 9 (where $a = 0.354$ and $b = 4.0e-5$ were empirically computed, and $P_{el,CHP}$ is the generated electric power). The maximum thermal efficiency of the CHP is computed based on the electric efficiency and a constant energy utilization factor equal to 0.88, as experimentally established by the CHP manufacturer.

The parameters of the EBITDA equation affected by the CHP scheduling via the load coefficient α_t and its change $\Delta\alpha$ are $E_{el,SC,t}$, $E_{el,TG,t}$, $E_{NG,CHP,t}$, $E_{th,SC,t}$, H_t, and EEC_t. Therefore, the optimization function is subject to constraints regarding the load coefficient α_t and its change $\Delta\alpha$, due to the technical specifications of the system explained in Section III-A. The following constraint on the load coefficient α_t must be satisfied:

$$\alpha_{min} < \alpha_t < \alpha_{max} \tag{10}$$

where α_{min} is the lower bound for the load coefficient, equal to 0.50, and α_{max} is the upper bound for the load coefficient, equal to 1. Additionally, α_t can be equal to zero to turn off the CHP.

An additional constraint must be satisfied, regarding the change in the load coefficient $\Delta\alpha_t$:

$$\Delta\alpha_{min} < \Delta\alpha_t < \Delta\alpha_{max} \tag{11}$$

where $\Delta\alpha_{min}$ is the lower bound for the change in α_t, equal to -0.50, and $\Delta\alpha_{max}$ is the upper bound for the change in α_t, equal to $+0.50$.

C. GENETIC ALGORITHM

Genetic algorithm (GAs) are metaheuristic techniques exploiting principles inspired by biological evolution processes to address optimization problems [49]. The points of the solution space are represented using a population of individuals, also called chromosomes, which evolve towards the optimum point. For this purpose, a fitness function is designed to attribute a score to individuals. The best ones are selected and recombined via mutation and crossover for the creation of offsprings.

In this work, multiple versions of the GA were developed. The first version, denoted as GA-LAST, is fitted on the previous 6-hour window of data and only the last element of the output individual is used as the $\Delta\alpha_t$, i.e. as the update of the load coefficient at the current time step t as defined in Equation 2. A short analysis of the GA used in this work is given in the following:

- An individual is an array of $dim_{individual} = 6$ values extracted from the set $\beta = \{0, 0.50, 0.55, 0.60, \ldots, 0.95, 1\}$, which represents an array of load coefficient updates $\Delta\alpha$ of the CHP: $[\Delta\alpha_1, \ldots \Delta\alpha_{dim_{individual}}]$.
- The population includes 100 individuals, which are randomly initialized at the beginning of the algorithm.
- The fitness function is the EBITDA value computed after averaging the 6-element long individual into a single value.
- Offsprings are obtained via the one-point crossover of the two candidates with the highest fitness and random mutations on these novel individuals. The mutation rate and one-point crossover rate are both set to 0.02.
- At each generation, the candidates with the two least fitting values are removed from the previous population and substituted with the newly generated chromosomes.
- The stopping criterion for the algorithm is twofold: the GA stops when a maximum number of generations set to 200 is reached, or the standard deviation of the maximum fitnesses in a time window comprising the latest 30 generations is lower than a threshold, set to 0.005.

All the mentioned hyperparameters were tuned via a manual search, evaluating the model on a subset of the training dataset, and the ones that yielded the best performance were selected. In particular, the values of the mutation rate and the one-point crossover rate which were tested were 0.02 and 0.05; the values of the population size were 50, 100, and 200; the values of the individual length were 2 and 6; the values of the maximum number of generations were 50, 100, 200, and 500.

The individuals generated by the GA may be invalid, i.e. they may violate the constraints of the system defined in Sec. III-B. Thus, each solution proposed by the GA is corrected

based on the constraints explained in Section III-B. However, since the minimization of such violations is desirable, two variants of the previously described GA algorithms were tested in order to increase the ability of the GA to satisfy the constraints:

1) GA with penalty (GA-LAST-penalty): a penalty term was subtracted from the EBITDA ($EBITDA_t$) in the fitness function, as shown in Equation 12.

$$fitness = \sum_{t=0}^{dim_{individual}} EBITDA_t - \mathbb{1}_{Violations_t} \times p \quad (12)$$

where p is the penalty amount and *Violations* is a sequence variable which stores the presence or absence of violations at each time t. The penalty p was subtracted from the EBITDA at each time t where a violation of the constraints occurred. Multiple penalties p were tested on a subset of the training dataset (€ 50, € 100, and € 150) and the value which yielded the best result is € 50.

2) GA with domain knowledge correction (GA-LAST-DK): in this case, the population was initialized randomly only with valid individuals, and the offsprings were corrected to make each individual valid before computing its fitness.

Fig. 4 details the flowchart of GA, with the two variants highlighted in red.

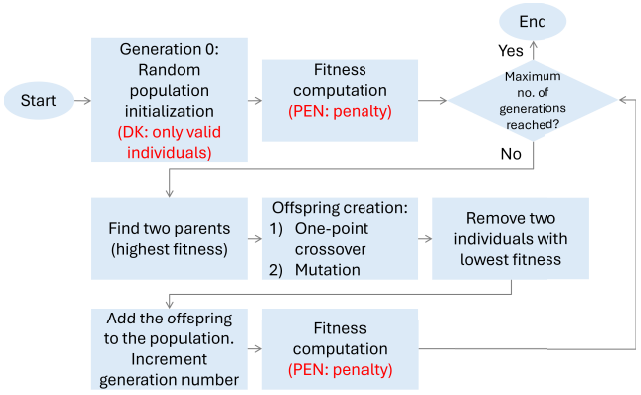


FIGURE 4. The GA flowchart.

Finally, another version of the GA was tested, called GA-AVG, which was fitted on the previous 6-hour window of data and the 6 elements of the output individual were averaged to compute the $\Delta\alpha_t$ for the current time step, instead of selecting only the last element. Also in this case, the GA-AVG-PEN and GA-AVG-DK variants were tested. The set of hyperparameters and the main implementation of the GA-AVG, GA-AVG-PEN, and GA-AVG-DK variants are the same as the GA-LAST, GA-LAST-PEN, and GA-LAST-DK ones.

D. DRL ALGORITHMS

The reinforcement learning (RL) paradigm is centred on discovering learning strategies that aim for the maximization of a numerical reward as an agent engages with and explores its environment [50]. The key components comprising a RL algorithm include: a policy, which dictates the computations determining the action taken by the agent in each time slot; a reward signal, representing the objective of the optimization problem; an optional environment model to facilitate predictions about future scenarios; and a value function, signifying the potential reward achievable from a particular state.

Within the realm of CHP dispatch scheduling, the mathematical model of the CHP system and the control system represent the environment and the agent, respectively. The agent actions denote changes in the load coefficient of the cogenerator. The reward at each step is determined by the EBITDA, depending on the actions taken by the agent within the environment.

1) PROBLEM MODELING: A NOVEL APPROACH FOR CHP DISPATCH SCHEDULING

Usually, RL problems are formulated as a Markov decision process (MDP). The MDP is a discrete-time stochastic control process which provides a mathematical framework to model decision-making problems. The MDP is defined by the tuple $(S, A, \mu_0, T, r, \gamma, H)$, where S is the state space, i.e. the set of environment states that can be observed by the agent; A is the action space, i.e. the set of actions that can be performed by the agent, $\mu_0 \in \Delta(S)$ is the initial state distribution; $T : S \times A \to \Delta(S)$ is the transition dynamics, where, for each state s and action a, $T(s, a)$ yields a probability distribution over states that the system may transition into when taking action a from state s; $r = S \times A \times S \to \mathbb{R}$ is the reward function; $\gamma \in [0, 1]$ is the discount factor, representing how much future rewards should be discounted when making decisions; H is the horizon, i.e. the maximum possible number of time steps in each episode.

The definition of MDP explicitly assumes that the Markov property in Equation 13 holds. The Markov property states that the distribution of the next state s_{t+1} of the environment depends only on the current state s_t and action a_t, rather than on the whole sequence of states and actions encountered up to time t, i.e.:

$$\forall t,\ p(s_t|s_1, a_1, \ldots, s_{t-1}, a_{t-1}) = p(s_t|s_{t-1}, a_{t-1}) \quad (13)$$

where s_t and a_t denote the state and action at time t, respectively.

However, this property rarely holds in real-world environments since the full state cannot be provided or includes uncertainty. Such problems are called partially observable problems [13] and they are commonly formulated as a partially observable Markov Decision Process (POMDP) [51]. A POMDP can be formally defined by a 6-tuple (S, A, T, r, Ω, O), where S, A, T, and r represent the

MDP states, actions, transition dynamics, and rewards, respectively. In a POMDP, the agent does not have access to the true system state: instead, it receives an observation $o \in \Omega$, where Ω is the observation space. This observation is derived from the actual system state based on the probability distribution $o \sim O(s)$.

Vanilla Deep Q-Learning lacks explicit methods to infer the true state of a POMDP and only works well if the observations closely match the underlying system states, i.e. if the Markov Property holds. Generally, estimating a Q-value from an observation can be highly inaccurate because $Q(o, a|\theta) \neq Q(s, a|\theta)$. Two primary methods to address partial observability can be found in the literature. The first involves simply incorporating historical information (defined as history) in the agent's observation to approximate the hidden state of the environment, thereby enabling more informed decision-making. For example, in [52] four consecutive Atari frames are stacked to mitigate partial observability in a DQN application. The second approach utilizes recurrent neural networks within the agent to process the history and detect relevant hidden state information. This second approach requires more computational resources and data to be trained due to its increased complexity. Reference [53] employ this method with DQN, demonstrating that the recurrent version can achieve comparable performance in Atari games even when provided with only a single frame as input. In this article, the first approach will be explored since it is less resource-intensive and, therefore, can be more easily adopted in a real-world context.

The RL task of the optimal CHP dispatch scheduling can be modelled as a POMDP, as explained in the following. The mathematical model of the CHP system and the control system represent the environment and the agent, respectively. The agent's action a at each time t is the change in the load coefficient of the CHP also denoted as $\Delta\alpha_t$:

$$a_t \in A \tag{14}$$
$$a_t = \Delta\alpha_t \tag{15}$$

where A is the action space, which is discrete for the DQN algorithm and continuous for the DDPG and SAC algorithms, as detailed in Sections III-D2-III-D4. The reward function r is the EBITDA.

$$r_t(s_t, a_t) = EBITDA_t \tag{16}$$

The state s of the environment at each time $t \forall t \in 1, 2, \ldots, T$ is defined as:

$$s_t \in S \tag{17}$$
$$s_t = (d_t, P_{el,D,t}, P_{th,D,t}, \alpha_{t-1}) \tag{18}$$

where S is the state space containing the variables that characterize the environment, namely the day of the week d_t, the electric demand $P_{el,D,t}$, the thermal demand $P_{th,D,t}$, and the previous load coefficient α_{t-1}. Due to the intrinsic uncertainty of future electric and thermal demands, the agent cannot observe $P_{el,D,t}$ and $P_{th,D,t}$ at the start of the time step t. Instead, it receives the observation o_t at the beginning of time step t, defined as follows:

$$o_t \in O \tag{19}$$
$$o_t = (d_t, P_{el,D,t-1}, P_{th,D,t-1}, \alpha_{t-1}) \tag{20}$$

where $P_{el,D,t-1}$ and $P_{th,D,t-1}$ represent the electric and thermal demands at time step $t-1$, respectively, which are readily available to the agent at the beginning of time step t. The terms d_t and α_{t-1} are the same as in s_t since the agent knows them without uncertainty. The state space S and the observation space O are defined by $S = O = [d^{min}, d^{max}] \times [P^{min}_{el,D}, P^{max}_{el,D}] \times [P^{min}_{th,D}, P^{max}_{th,D}] \times [\alpha^{min}, \alpha^{max}]$, where d^{min} and d^{max} are the first and last day of the week, $P^{min}_{el,D}$ and $P^{max}_{el,D}$ are the minimum and maximum electric demand values, $P^{min}_{th,D}$ and $P^{max}_{th,D}$ are the minimum and maximum thermal demand values, and α^{min} and α^{max} are the minimum and maximum load coefficients.

Since only the observation o_t is available instead of s_t, the agent receives the history h_t from the environment to select the action a_t. In a POMDP problem, history is typically defined as $h_t = (o_1, a_1, o_2, a_2, \ldots, o_{t-1}, a_{t-1}, o_t)$, i.e. it consists of the whole sequence of past observations. However, in order to reduce the complexity of the problem, in this work the history is defined as:

$$h_t = (o_{t-\tau+1}, \ldots, o_{t-1}, o_t) \tag{21}$$

where τ is the time horizon hyperparameter, i.e. the size of the past observation window (called history size).This definition includes only the observation history and not the action history since the latter does not contribute to the estimation of the state variables $P_{el,D,t}$ and $P_{th,D,t}$. Multiple time horizons τ were considered and their effects were analysed in this article. The first one is $\tau = 1$, which is comparable to the MDP approach proposed in most of the CHP dispatch scheduling literature. The second and third ones are $\tau = 2$, and $\tau = 6$, which provide two different trade-offs between computational complexity and degree of information provided by the history.

The action a at each time t is the change in the load coefficient of the CHP also denoted as $\Delta\alpha_t$:

$$a_t \in A \tag{22}$$
$$a_t = \Delta\alpha_t \tag{23}$$

where A is the action space, which is discrete for the DQN algorithm and continuous for the DDPG and SAC algorithms, as detailed in Sections III-D2-III-D4.

The reward function r is the EBITDA.

$$r_t(s_t, a_t) = EBITDA_t \tag{24}$$

To summarize, at each time t, the agent, which represents the control system, obtains the history h_t: based on h_t and the learned policy function π, it selects an action a_t. The environment then reacts to the action and outputs a new observation o_{t+1} and the reward value r_t. The agent receives such information and repeats the previous steps until the

end of the optimization episode. A schematic of the RL framework is shown in Fig. 5, where the Environment block is as Fig. 1.

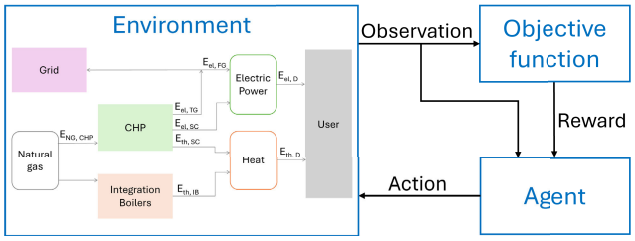


FIGURE 5. Scheme of the RL framework. Environment block is as Fig. 1.

2) DQN ALGORITHM

The initial DRL model chosen for this analysis is a DQN. DQN, originally presented in [20], is a DRL technique that combines Q-learning [54] with deep neural networks and experience replay for the maximization of the reward an agent obtained as it interacts with the environment. The off-policy, model-free approach called Q-learning integrated into DQN is suitable for RL problems that do not necessitate an explicit model of the environment [54]. The DQN model estimates the optimal Q value $Q^*(s, a) = \max \pi Q^\pi(s, a)$ by learning a parameterized value function $Q(s, a; \theta_t)$ via Q-learning. The Q-learning update of the θ parameters is expressed in [20] as:

$$\theta_{t+1} = \theta_t + \alpha \left(Y_t - Q(s, a; \theta_t)\right) \nabla_{\theta_t} Q(s, a; \theta_t) \tag{25}$$

where state s action a (together with an observed reward r) have been sampled from a replay memory storing past experiences to ensure off-policy learning of the greedy strategy $a'_t = max_A Q(s'_t, A; \theta_t)$ where s'_t has been observed by sampling an emulator of the system; α represents a step size; and Y_t is the target value, computed in [20] as:

$$Y_t \equiv r + \gamma \max_a Q(s, a; \theta_t^-) \tag{26}$$

where θ_t^- are the weights of the target network, and $\gamma \in [0, 1]$ is the discount factor, representing how much future rewards should be discounted when making decisions. The target network is initially a copy of the main network, and its weights θ_t^- are periodically updated by copying the weights of the main network.

In this work, the DQN algorithm is fed with input data representing the observation history of the system from the preceding τ hours, concatenated to form a single array of $\tau * d_o$ length, where d_o is the dimensionality of the observation o_t, i.e. 4. Leveraging this information, it can output the optimal load coefficient update for the forthcoming hour, aiming to maximize the reward function, i.e. the EBITDA. The three values of the history size τ (1, 2, and 6) were tested in three separate experiments, denoted as DQN-1, DQN-2, and DQN-6.

By utilizing the Q policy, the algorithm iteratively updates the weights of the neural network during training at each hour. This continual adjustment enables the refinement of the load coefficient scheduling to optimize performance.

In the initial history, the load coefficient is set as two adjacent values selected from the set {0, 0.50, 0.55, 0.60, ..., 0.95%, 1}, where a load coefficient set to 0 denotes the cogenerator engine being turned off). The action space comprises 21 possible changes to the load coefficient $\Delta\alpha$, ranging from -0.50 to $+0.50$ with a step equal to 0.05. These changes are constrained within the limits specified in Section III-B. The states, actions and rewards at each time t were re-scaled to the interval $[-1, 1]$, to improve the stability of gradients: a variant of the Min-Max scaling method was used, defined in Equation 27.

$$z_i = 2\left(\frac{x_i - x_{\min}}{x_{\max} - x_{\min}}\right) - 1 \tag{27}$$

where x_i is the original value, $x_{\min}$ is the minimum possible value, $x_{\max}$ is the maximum possible value, and z_i is the scaled value. The $x_{\min}$ and $x_{\max}$ values were set to -200 and 300, respectively, based on the reward formulation.

The objective is to maximize the EBITDA, which serves as the reward function. The neural network architecture utilized in this model is a Multi-Layer Perceptron (MLP) [55] with two hidden layers with 256 units each, Rectified Linear Unit (ReLU) activation functions [56], and Adam optimization with a learning rate set to 0.005. The agent employs an ϵ-greedy Q exploration policy to choose actions, coupled with the exponential decay strategy of the hyperparameter ϵ shown in Equation 28.

$$\epsilon_{\text{threshold}} = \epsilon_{\text{end}} + (\epsilon_{\text{start}} - \epsilon_{\text{end}}) \times e^{-\frac{\text{steps_done}}{\text{eps_decay}}} \tag{28}$$

The threshold for exploration $\epsilon_{\text{threshold}}$ is calculated by exponentially decreasing the value of epsilon from $\epsilon_{\text{start}} = 0.9$ to a $\epsilon_{\text{end}} = 0.05$ with eps_decay $= 1000$, and steps_done is the current number of steps. The memory capacity for experience replay was configured to be 10^6. The target Q network weights θ_t^- are updated with soft updates, based on Equation 29.

$$\theta^- \leftarrow \tau\theta + (1 - \tau)\theta^- \tag{29}$$

where τ is equal to 0.0005.

The main parameters of the implemented DQN method are summarized in Table 5. All the mentioned hyperparameters were tuned via a manual search, evaluating the model on a subset of the training dataset, and the ones that yielded the best performance were selected. In particular, the number of hidden units per layer was tuned by setting it to 200, 256 and 300; the batch size to 128, 256, and 512; the learning rate to 0.0005, 0.001, 0.005, 0.01; the replay size to 10^4 and 10^6; γ to 0.95 and 0.99.

Before performing an environment step and computing its reward, i.e. the EBITDA, the validity of the action selected by the model was checked: if one of the constraints defined in Section III-B was violated, the action was corrected to prevent the violation.

3) DDPG ALGORITHM

The DDPG algorithm, presented in [21], is a model-free off-policy DRL algorithm which can simultaneously learn a Q-function via the Bellman equation, and a policy thanks to the Q-function learned. It is an adaptation of the theory of Deterministic Policy Gradient (DPG) algorithms presented in [57] for RL problems with continuous action spaces. The DDPG makes use of the actor-critic architecture. Stochastic actor-critic models include an *actor* component which adjusts the weights θ of a stochastic policy denoted by $\pi\theta$, and a *critic* component which estimates the action-value function $Q(s, a; w) \approx Q^{\pi}(s, a)$ with parameters w, given state s, action a and the true action-value function $Q^{\pi}(s, a)$. On the other hand, the DDPG enables the use of continuous action spaces by learning a deterministic policy $\mu(s; \theta)$ instead, which maps a state s to a deterministic action a. Notably, the actor component in DDPG is a deep neural network denoted as $\mu(s; \theta^{\mu})$ with weights θ^{μ}, and a critic network denoted as $Q(s, a; \theta^{Q})$ with weights θ^{Q}. Taking inspiration from the DQN, the DDPG utilizes the concept of replay memory, as well as target networks Q' and μ', which are the target critic network and the target actor network with their respective weights $\theta^{Q'}$ and $\theta^{\mu'}$. The critic network is updated by minimizing the loss in Equation 30 by [21]

$$L = \frac{1}{N} \sum_{i} (y_i - Q(s_i, a_i; \theta^{Q}))^2 \tag{30}$$

where N is the mini-batch dimension; s_i and a_i are the state and action sampled from the replay buffer, with observed reward r_i and next state s_{i+1}, similarly to the DQN algorithm; y_i is the target Q value, computed in [21] as:

$$y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}; \theta^{\mu'}); \theta^{Q'}) \tag{31}$$

The actor network is updated via the sampled policy gradient as in [21]:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_{i} \nabla_a Q(s, a|\theta^{Q})\Big|_{s=s_i, a=\mu(s_i)} \nabla_{\theta^{\mu}} \mu(s|\theta^{\mu})\Big|_{s=s_i} \tag{32}$$

In [21], the target networks are updated via soft updates as follows:

$$\theta_{Q'} \leftarrow \tau\theta_{Q} + (1 - \tau)\theta_{Q'} \tag{33}$$

$$\theta_{\mu'} \leftarrow \tau\theta_{\mu} + (1 - \tau)\theta_{\mu'} \tag{34}$$

To improve exploration, a noise term $\mathcal{N}_t$ is added to the actions selected by the model at each time t, resulting in $a'_t = \mu(s'_t; \theta^{\mu}) + N_t$ [21], where $s't$ is the state sampled from the system emulator and $\mu(s't; \theta^{\mu})$ is the current policy.

In this work, the DDPG algorithm receives as input the observation history of the system from the preceding τ hours and provides the optimal adjustment of the load coefficient of the subsequent hour to maximize the EBITDA. The observation history is concatenated to form a single array of $\tau * do$ length, where do is the dimensionality of the observation ot, i.e. 4. The three values of the history size τ (1, 2, and 6) were tested in three separate experiments, denoted as DDPG-1, DDPG-2, and DDPG-6.

The OpenAI Spinning Up library [58] implementation of the DDPG was leveraged. In this implementation, the noise term is an uncorrelated mean-zero Gaussian noise scaled by a factor equal to 0.05. This implementation uses Polyak averaging [59] for the update of the target Q-networks parameters during training, with an interpolation factor that was set to 0.995 (the default value in the OpenAI Spinning Up library). To enhance exploration at the start of training, the agent samples actions from a uniform random distribution over viable actions for a fixed amount of steps equal to 10000 without network updates. After this period, the agent starts the normal exploration process and network updates, with the learning rate of both the agent and the critic set to 0.0005. The states and rewards at each time t were scaled to the interval $[-1, 1]$ via the scaling formula defined in Equation 27, and actions were clipped to the same interval via Tanh activation, to improve the stability of gradients.

Similarly to the presented DQN implementation, the validity of the action selected by the model was checked and a correction of the action was made in case of violations of constraints.

The main parameters of the implemented DDPG technique are summarized in Table 5. All the mentioned hyperparameters were tuned via a manual search, evaluating the model on a subset of the training dataset, and the ones that yielded the best performance were selected. In particular, the number of hidden units per layer was tuned by setting it to 200, 256 and 300; the batch size to 128, 256, and 512; the learning rate to 0.0005, 0.001, 0.005, and 0.01; the replay size to 10^4 and 10^6; γ to 0.95 and 0.99; the number of steps before starting network updates to 10^3 and 10^4.

4) SAC ALGORITHM

The SAC algorithm was introduced in [22] to overcome the limits of traditional Q-learning algorithms when they are applied to problems with large continuous domains: it is an off-policy actor-critic DRL algorithm which optimizes a stochastic policy, centered around the concepts of the maximum entropy RL framework. The three main components of the SAC algorithm are an actor-critic architecture with two separate networks for policy estimation and value function estimation, an off-policy formulation relying on a replay memory, and entropy maximization for exploration and stability. More specifically, maximum entropy RL consists of optimizing the policy to maximize the expected return and the expected entropy of the policy concurrently, based on the following update of the optimal policy π^* by [22]:

$$\pi^* = \arg\max_{\pi} \sum_{t} \mathbb{E}_{(s_t, a_t) \sim \rho_{\pi}} \left[r(s_t, a_t) + \alpha_{temp} \mathcal{H}(\pi(\cdot|s_t)) \right] \tag{35}$$

where $\mathcal{H}(\pi(\cdot|s_t))$ is the entropy of the policy at state s_t, and α_{temp} is the temperature parameter which controls the relative

importance of the entropy against the reward, thus affecting the exploration-exploitation trade-off.

The SAC algorithm considers a state value function $V_\psi(s_t)$ parameterized by the weights ψ of a neural network, and a Q-value function $Q_\theta(s_t, a_t)$ parameterized by the weights θ of another neural network. The soft value function network and the Q-value function network are trained to minimize the following squared residual errors, respectively, as presented in [22]:

$$J_V(\psi) = \mathbb{E}_{s_t\sim D}\left[\frac{1}{2}\left(V_\psi(s_t) - \mathbb{E}_{a_t\sim\pi_\varphi}\left(Q_\theta(s_t, a_t) - \log\pi_\varphi(a_t|s_t)\right)\right)^2\right] \tag{36}$$

$$J_Q(\theta) = \mathbb{E}_{(s_t,a_t)\sim D}\left[\frac{1}{2}\left(Q_\theta(s_t, a_t) - \hat{Q}(s_t, a_t)\right)^2\right] \tag{37}$$

with

$$\hat{Q}(s_t, a_t) = r(s_t, a_t) + \gamma\mathbb{E}_{s_{t+1}\sim p}\left[V_{\overline{\psi}}(s_{t+1})\right] \tag{38}$$

where D is the replay memory, $V_{\overline{\psi}}$ is the target value network, with $\overline{\psi}$ representing an exponentially moving average of the value network weights; γ is the discount factor and p is the unknown state transition probability.

The policy π is modelled as a Gaussian with mean and covariance determined by a network, so its parameters can be learned via the minimization of the expected Kullback-Leibler (KL) divergence as in [22]:

$$J_\pi(\varphi) = \mathbb{E}_{s_t\sim D}\left[\mathrm{D_{KL}}\left(\pi_\varphi(\cdot|s_t) \parallel \frac{\exp(Q_\theta(s_t,\cdot))}{Z_\theta(s_t)}\right)\right] \tag{39}$$

$$= \mathbb{E}_{s_t\sim D,\epsilon_t\sim\mathcal{N}}\left[\log\pi_\varphi\left(f_\varphi(\epsilon_t; s_t)|s_t\right) - Q_\theta\left(s_t, f_\varphi(\epsilon_t; s_t)\right)\right] \tag{40}$$

where the noise vector ϵ_t is sampled from a fixed distribution, $Z_\theta(s_t)$ is a partition function, and $f_\varphi(\epsilon_t; s_t)$ is a neural network transformation.

In this work, the SAC algorithm takes as input the observation history of the system from the preceding τ hours and generates the optimal $\Delta\alpha$ of the forthcoming hour to maximize the EBITDA. The observation history is concatenated to form a single array of $\tau * d_o$ length, where d_o is the dimensionality of the observation o_t, i.e. 4.The three values of the history size τ (1, 2, and 6) were tested in three separate experiments, denoted as SAC-1, SAC-2, and SAC-6.

The OpenAI Spinning Up library [58] implementation of the DDPG was employed. The Spinning Up implementation uses Polyak averaging [59] for the update of the target Q-networks parameters during training, with an interpolation factor that was set to 0.995. Also in this case, the initial 10000 steps were dedicated to the uniform random sampling of actions without network updates to improve exploration; states, actions and rewards were scaled in the interval $[-1, 1]$ via the scaling formula defined in Equation 27 to improve stability; and the validity of actions was checked and unfeasible actions corrected. Differently from the original SAC algorithm, in this work the $\alpha temp$ parameter was not set to a fixed value, which yielded poor performance: rather, it was automatically adjusted during training, as proposed in [60]. The objective for the computation of the gradients for alpha presented in [60] is:

$$J(\alpha) = \mathbb{E}_{a_t\sim\pi_t}\left[-\alpha\log\pi_t(a_t|s_t) - \alpha\overline{\mathcal{H}}\right] \tag{41}$$

where $\overline{\mathcal{H}}$ is the target entropy.

The main parameters of the implemented SAC method are summarized in Table 5. All the mentioned hyperparameters were tuned via a manual search, evaluating the model on a subset of the training dataset, and the ones that yielded the best performance were selected. In particular, the number of hidden units per layer was tuned by setting it to 200, 256 and 300; the batch size to 128, 256, and 512; the learning rate to 0.0005, 0.001, 0.005, and 0.01; the replay size to 10^4 and 10^6; γ to 0.95 and 0.99; the number of steps before starting network updates to 10^3 and 10^4.

TABLE 5. Summary of the hyperparameters of the DRL algorithms.

Hyperparameter	DQN	DDPG	SAC
Batch size	256	256	256
γ	0.95	0.95	0.95
No. layers	2	2	2
No. hidden units per layer	256	256	256
Learning rate	0.005	0.0005	0.005
Replay size	10^6	10^6	10^6
Update every (No. steps)	1	3	3
Update after (No. steps)	1	10^4	10^4

IV. RESULTS AND DISCUSSION

In this section, a description of the datasets for training and testing the algorithms is provided. Then, the baselines for performance evaluation are explained. Lastly, the results of the application of each algorithm on the test dataset are reported and discussed.

A. DATASET

The dataset of energy demands of the Italian plant is made of 8760 data points collected hourly each day in 2021. Each point is composed of the following three features:

- day of the week;
- electrical power demand;
- thermal power demand.

The last item was computed as the sum of the heat demand in the form of steam, high-temperature hot water, and low-temperature hot water. The dataset was split into a training set and a test set: the training set was created by extracting one week of data every two months, for a total of seven weeks (1176 data points); the remaining data formed the test set, for a total of 45 weeks and 3 days (7584 data points).

B. BASELINES

Multiple deterministic baselines were selected based on the experience of domain experts, to assess the performance of the models.

1) **Scheduling based on the day of the week (DoW)**: the load coefficient of the CHP was increased from 0 up to 0.85 with a 0.50 step on Monday at 00:00, and it was kept fixed until Friday at 23:00 when it was decreased to 0 with a 0.50 step. The process was repeated every week.
2) **Scheduling with a fixed load coefficient (FIX)**: the load coefficient was set to 0.85 during the whole test period.
3) **Scheduling with Electric Load Following (ELF)**: the load coefficient was adjusted hour-by-hour to follow the electric power load trend. If the absolute difference between the electric power load and the electric power output of the CHP at the previous hour was larger than a tolerance value, set to 0.5, then the load coefficient was increased or decreased by 0.1 while respecting the technical constraints of the system explained in Section III-B. When the electric power load was below the minimum modulation threshold, the CHP was turned off.

The scheduling with a fixed load coefficient can be considered as a minimum boundary for the EBITDA. On the other hand, the scheduling with electric load following is taken as a benchmark since it is recognized as an optimal solution by domain experts.

C. RESULTS

This section presents the training and test process utilized for each algorithm, followed by the presentation and discussion of the results.

1) TRAINING AND TEST CONFIGURATIONS

The baseline strategies, i.e. DoW, FIX and the ELF, were applied directly to the test dataset, assuming that the CHP is turned off at the beginning of the simulation. The DRL models (DQN, DDPG, and SAC) were trained using the training dataset as input data, then they were tested on the test dataset. Finally, the GAs were directly fitted and tested on the test dataset, as explained in Section III-C.

2) YEARLY TEST RESULTS

Table 6 presents a summary of the revenues and costs obtained at test time. The revenues reported are the *EBITDA*, the revenues in terms of EECs (R_{EEC}), the revenues from selling the generated electricity exceeding the demand to the grid (R_{TG}), and the avoided costs of the electricity (AC_{el}) and natural gas (AC_{th}), due to self-consumption of power. These variables were computed as follows:

$$R_{EEC} = \sum_t r_{EEC} \times EEC_t \tag{42}$$

$$R_{TG} = \sum_t r_{el} \times E_{el,TG,t} \tag{43}$$

$$AC_{el} = \sum_t c_{el} \times E_{el,SC,t} \tag{44}$$

$$AC_{th} = \sum_t c_{NG} \times \frac{E_{th,SC,t}}{\eta_{IB}} \tag{45}$$

$$C = \sum_t c_{NG} \times E_{NG,CHP,t} + c_{O\&M} \times H_t \tag{46}$$

The costs C include the O&M costs and the cost of the natural gas which was purchased. Finally, the benefit-cost ratio (*BCR*) is reported in the last column. The *BCR* definition is shown in Equation 47.

$$BCR = \frac{R_{EEC} + R_{TG} + AC_{el} + AC_{th}}{AC_{th}} \tag{47}$$

As far as the baselines are concerned, the EBITDA obtained via the FIX strategy is the lowest, followed by the DoW strategy. The baseline algorithm which performed best is ELF, with an EBITDA equal to € 461354: this is an expected result since ELF is considered an optimal solution by domain experts. This result is taken as a benchmark for the comparison of the results of the GA and DRL algorithms. Notably, the ELF strategy is the one with the lowest O&M and natural gas costs C (€ 614554) and the highest *BCR* (1.75).

The DQN algorithms exhibit diverse performance based on the history size. DQN-1, achieves the highest EBITDA (€ 462084), benefiting from substantial avoided costs of electricity (€ 810503) and natural gas (€ 206009). Increasing the history size does not lead to better performance in this case: DQN-6 leads to a 3% lower EBITDA.

DDPG-2 achieves the highest EBITDA overall (€ 466082), which is 55% higher than the FIX baseline one, but only 1% higher than the ELF one. This variant shows high AC_{el} (€ 811386) and moderate R_{TG} (€ 23291). Notably, in DDPG-6 the EBITDA is less than 1% lower than the DDPG-2 one, and has the same *BCR*. The results suggest that a history size equal to 2 provides sufficient context without the drawbacks of excessive complexity.

SAC algorithms have consistent performance across different history sizes. SAC-1 achieves the highest EBITDA (€ 465621), with the highest R_{EEC} (€ 101301). SAC-6 and SAC-2 follow closely, indicating that these models effectively utilize the history information to provide an efficient operational strategy. Indeed, they exhibit the highest *BCR* after ELF, which is less than 3% higher.

Regarding the GAs, GA-AVG achieved the highest R_{TG} and avoided costs, but caused the highest C overall (€ 1026017), resulting in a lower *EBITDA* (€ 315006). This pattern characterizes all the GAs: they all have a *BCR* similar to the FIX one, suggesting that their strategies may be similar. This hypothesis will be confirmed in Section IV-C4.

Overall, the history size parameter τ, i.e. the time horizon in the POMDP, influences the performance of DRL models. For DQN, a history size equal to 2 provides the best balance between performance and computational complexity. SAC and DDPG models, however, demonstrate robustness across different τ: they can effectively integrate historical information without substantial performance degradation. These findings underscore the importance of carefully

TABLE 6. Revenues and costs obtained by applying each algorithm on the test dataset. Highest values are highlighted in bold.

Category	Algorithm	$EBITDA$ [€]	R_{EEC} [€]	R_{TG} [€]	AC_{el} [€]	AC_{th} [€]	C [€]	BCR
Baselines	ELF	461354	94007	5910	783447	192542	614554	**1.75**
	DoW	390329	92314	67945	772994	195131	738055	1.53
	FIX	300935	85062	127334	781187	215031	907679	1.33
DQN	DQN-1	462084	101234	29779	810503	206009	685441	1.67
	DQN-2	449361	97790	21993	770504	199762	640688	1.70
	DQN-6	447025	99788	39238	806503	203783	702289	1.64
DDPG	DDPG-1	447167	100126	31589	789118	206469	680137	1.66
	DDPG-2	**466082**	99898	23291	811386	204089	672583	1.69
	DDPG-6	464833	99511	23452	808922	202712	669765	1.69
SAC	SAC-1	465621	**101301**	26179	801798	205597	669255	1.70
	SAC-2	464174	96091	12825	788445	194505	627692	1.74
	SAC-6	464598	98057	19826	799333	198440	651058	1.71
GA	GA-LAST	315039	88868	161740	872692	**217455**	**1025717**	1.31
	GA-LAST-PEN	282962	82585	145877	783761	205739	935000	1.30
	GA-LAST-DK	373721	92891	98084	831854	209521	858631	1.44
	GA-AVG	315006	88872	**161842**	**872843**	**217465**	**1026017**	1.31
	GA-AVG-PEN	281656	81936	142965	775453	204083	922782	1.31
	GA-AVG-DK	310591	86464	130661	801673	212069	920278	1.34

TABLE 7. Algorithm execution time.

Algorithm	Time (ms)
SAC-1	2.11 ± 0.23
GA-LAST-DK	2097.92 ± 1312.98
DQN-1	1.80 ± 0.21
DDPG-2	1.91 ± 0.82

selecting τ to enhance CHP dispatch scheduling performance while managing computational complexity.

Finally, the best-performing algorithm in terms of EBITDA for each category were selected and their per-time step computation time was computed in the test phase. This research utilized an Intel Xeon CPU @ 2.30GHz with 8 threads, an NVIDIA Tesla T4 GPU with 15.36 GB memory, and CUDA version 12.2. Table 7 summarizes these results: notably, the three DRL algorithms have similar computation times, while the GA time is around three orders of magnitude larger.

3) QUARTERLY ANALYSIS

Fig. 6(a)-(d) illustrate the returns and costs across four quarters for DQN-1, DDPG-2, SAC-1, GA-LAST-DK, and ELF, which are the best-performing algorithms in terms of EBITDA for each category. Notably, R_{EEC} remains relatively low and stable across all algorithms in each quarter, whereas R_{TG}, AC_{el}, and AC_{th} exhibit greater variation, emphasizing the differences in how each algorithm manages excess generation and self-consumption. The trends in $EBITDA$, AC_{el}, and AC_{th} for DQN-1, DDPG-2, SAC-1, and ELF suggest that these models maintain efficiency and cost-effectiveness. Conversely, GA-LAST-DK's consistently high C highlights the lack of trade-off between profit and operational expenses. Such a pattern is evident when the daily cost-benefit ratio of each algorithm shown in Fig. 7 is examined: GA-LAST-DK exhibits the highest variability and the broadest range.

4) WEEKLY ANALYSIS

Fig. 8, 9 and 10 show the total electric power outputs and the self-consumed thermal outputs of the best-performing model for each category (DQN-1, DDPG-2, SAC-1, GA-LAST-DK, and ELF), compared to the energy demands during a winter example week (18 January 2021 - 24 January 2021), a summer example week (12 July 2021 - 18 July 2021) and a mid-season example week (25 October 2021 - 31 October 2021), respectively. It is possible to notice that the hourly electric and thermal demands in winter and mid-season are similar, while the thermal demand lowers in mid-season, on average. During weekdays, most of the strategies correctly set the CHP load coefficient to 100% to fulfil the high energy demand. The only exception is the GA-LAST-DK strategy, which causes over-generation, especially during lower demand periods. The ramp-up and ramp-down periods of the electric and thermal demand are the most critical, i.e. the beginning of the week and the beginning of the weekend. During these periods, DDPG-2 and SAC-1 strategies are highly responsive but show instability, with CHP power outputs strongly oscillating. This behavior can cause mechanical stress and potential wear on the CHP components; thus, it should be avoided. Multiple methods may be utilized for the real-world application of these strategies to ensure the safety constraints of the system are guaranteed: the reward function may be tuned to incorporate a penalty in case of oscillating patterns in the scheduling strategy, to encourage the agent to adopt the desired behaviour. In addition, the control pipeline may be extended by introducing an algorithm that modulates the actions of the agent, forcing them to safe ranges. Conversely, the DQN-1 scheduling and ELF scheduling present a more stable pattern. However, ELF scheduling is sub-optimal while the demands are ramping up or down, due to its intrinsic delayed response.

The example week in the summer is a particular week in which the electric power demand does not follow the common weekly dynamics detected in the rest of the dataset. This anomalous condition further highlights the unstable scheduling of the load coefficient from the DDPG-2 and SAC-1 algorithms. The DQN-1 scheduling is more stable, but an over-consumption episode occurs in the first 24 hours.

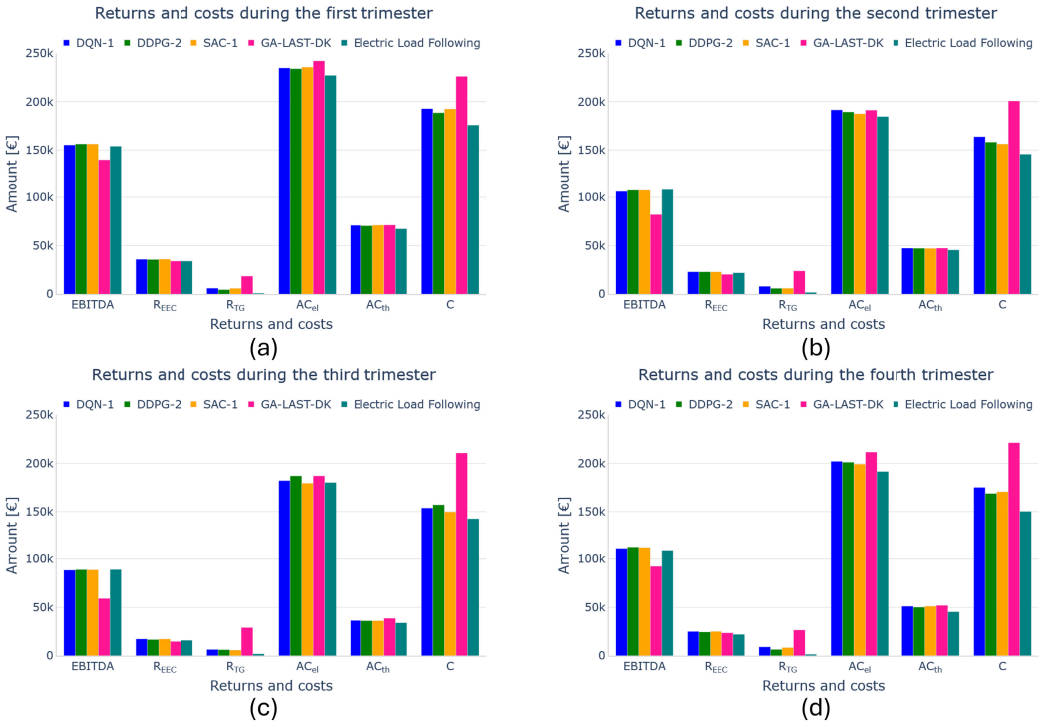


FIGURE 6. Comparison of rewards and costs on the first (a), second (b), third (c), and fourth (d) quarter of the year 2021.

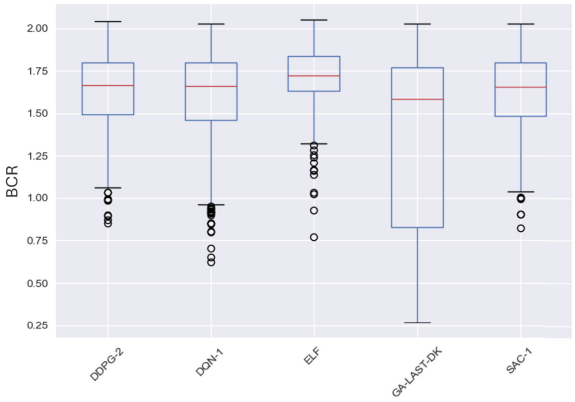


FIGURE 7. Box plots of the daily BCR of the DDPG-2, DQN-1, ELF, GA-LAST-DK, and SAC-1 strategies.

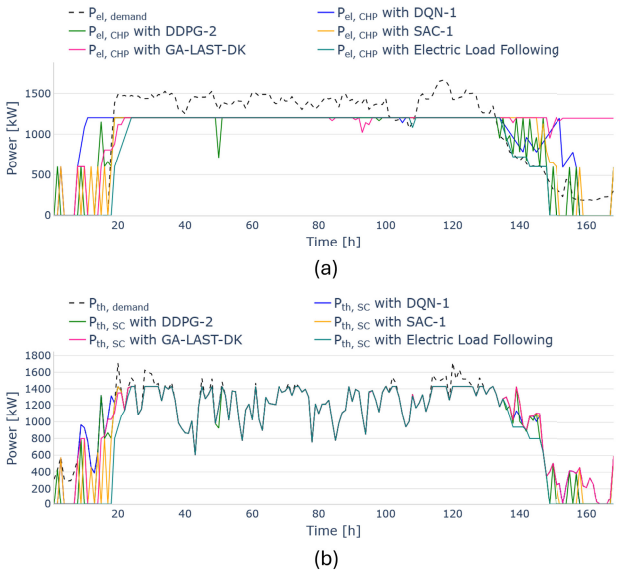


FIGURE 8. Total electric power output (a) and self-consumed thermal power output (b) generated via each operation strategy, compared with the electric and thermal demands, respectively, during an example week in winter (18 January 2021 - 24 January 2021).

GA-LAST-DK tends towards over-generation, whereas the ELF strategy maintains a stable profile by keeping the engine turned off in the first 36 hours of the week. These observations are supported by the fact that the overall costs C derived by the application of ELF during the third quarter (see Fig. 6(c)) is lower than the GA-LAST-DK costs by 32%. The over-generation observed in the GA-LAST-DK strategies confirms the hypothesis that the GAs tend to learn patterns similar to those produced by the FIX strategy.

5) SENSITIVITY ANALYSIS: VARIABLE PRICE OF ELECTRIC ENERGY

A sensitivity analysis was conducted to evaluate the economic performance of the models under a variable electric energy price scenario. This analysis is crucial to understand the robustness and adaptability of each method in response to fluctuating economic conditions, which are inherent to the CHP economic dispatch problem. In this analysis the real hourly price of electric energy c_{el_t} was utilized, instead of the fixed one defined in Table 3. The prices from 2023 rather than 2021 were used because the latter were highly unstable

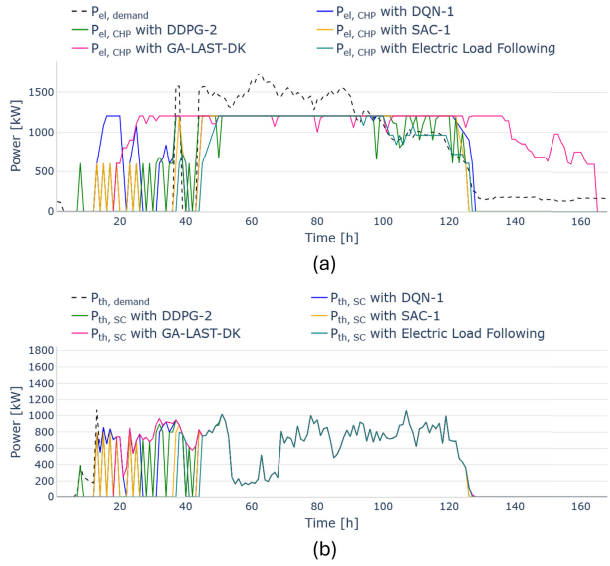


FIGURE 9. Total electric power output (a) and self-consumed thermal power output (b) generated via each operation strategy, compared with the electric and thermal demands, respectively, during an example week in summer (12 July 2021 - 18 July 2021).

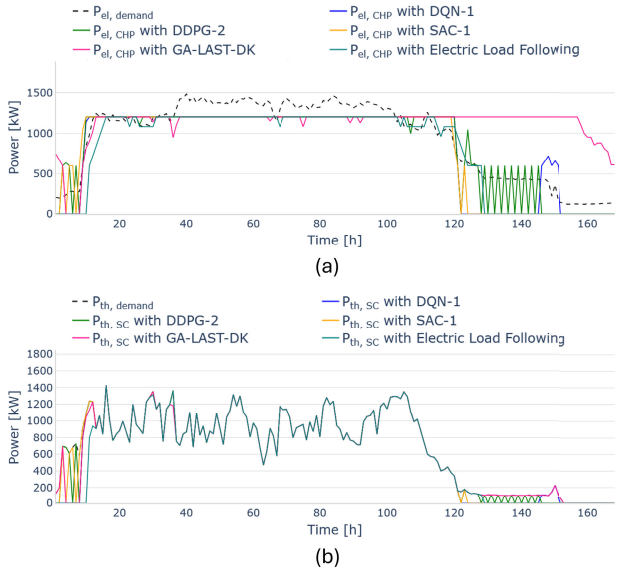


FIGURE 10. Total electric power output (a) and self-consumed thermal power output (b) generated via each operation strategy, compared with the electric and thermal demands, respectively, during an example week in mid-season (25 October 2021 - 31 October 2021).

due to the 2021 global energy crisis. In contrast, the prices in 2023 provide a more reliable basis to evaluate the economic performance of the models under typical market conditions.

To perform this analysis, each previously trained DRL model was tested on the test dataset employing the real c_{el_t} in the EBITDA computation defined in Equation 4 at each time step t. The GA and baseline scheduling strategies found in the previous experiments were applied in the same way. The results are shown in Table 8.

When subjected to variable electric energy prices, all algorithms show an improvement in *EBITDA* and *BCR*. For instance, DQN-6, which had an *EBITDA* of €447025 and a *BCR* of 1.64 under fixed prices, improved to €510113 and 1.73, respectively, under variable prices. Similarly, SAC-2 increased its *EBITDA* from €464174 to € 489451, and its *BCR* is the highest. The GA algorithms, particularly GA-LAST and GA-AVG, also showed significant increases in *EBITDA* and revenues from selling generated electricity (R_{TG}) under variable prices, due to their over-generation patterns which prevent the purchase of electric energy from the grid: indeed, their overall costs C increased as well. Overall, the variable pricing scenario demonstrates the potential for the algorithms to enhance profitability even when hourly price variations occur.

6) ENVIRONMENTAL ANALYSIS

From a decarbonization perspective, the environmental impact of CHP is a key factor in determining its effectiveness in contributing to a future low-carbon electricity mix. Therefore, an environmental analysis of each model was performed, considering two metrics. The first one is the total CO_2 emissions from the fuel combustion in the CHP (C_{tot}), equal to:

$$C_{tot} = V_{NG} \times LHV_{NG} \times c_{NG} \tag{48}$$

where V_{NG} is the volume of natural gas used, LHV_{NG} is the lower heating value of natural gas, assumed to be 9.766 kWh/Sm^3 [61], and cNG is the CO2 intensity value of the natural gas provided to the CHP plant from the national gas grid, assumed equal to 201.6 g$CO2$/kWh [61]. The second metric considered is the CO2 intensity of the electricity generated by the CHP (cel) [62], defined as:

$$c_{el} = \frac{C_{el}}{E_{el}} \tag{49}$$

with

$$C_{el} = C_{tot} \frac{E_{el}}{E_{el} + E_{th}} \tag{50}$$

where C_{el} is the amount of CO_2 emissions allocated to the electricity output, E_{el} and E_{th} are the electricity and heat production, respectively. Figure 11 shows the total CO_2 emissions of each approach. The results highlight significant differences, with GA-LAST-DK exhibiting the highest emissions, indicating poor environmental efficiency. In contrast, ELF achieves the lowest emissions, making it the most sustainable approach. SAC-1, DDPG-2, and DQN-1 show similar performance. Figures 12 and 13 illustrate weekly and monthly average variations in CO_2 intensity, respectively. The weekly CO_2 intensity remains stable mid-week, but shows differences in the first 12 hours of the week and at the weekend: during these times, GA-LAST-DK consistently produces the highest emissions and ELF the lowest, with an overall weekly average equal to 196 g_{CO2}/kWh. The DRL algorithms have a similar pattern, with SAC-1

having the lowest weekly average value (203 g_{CO2}/kWh). The monthly CO_2 intensity trends reveal seasonal fluctuations, with GA-LAST-DK consistently underperforming. The other models display closer trends, though ELF shows a notable decrease in CO_2 intensity in December and SAC-1 in August, suggesting good adaptability to seasonal conditions.

Overall, ELF consistently outperforms the other models in total CO_2 emissions and CO_2 intensity, making it the most sustainable choice for CHP operations and SAC-1 as the best reinforcement learning-based alternative.

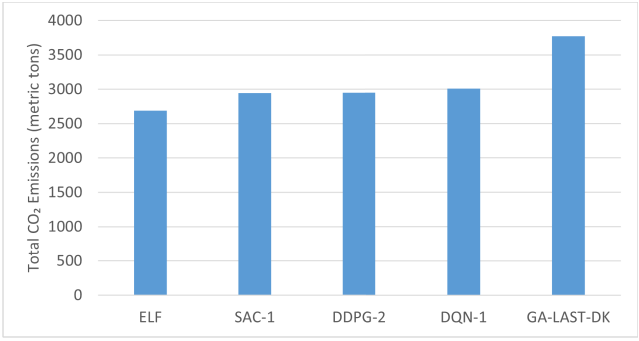


FIGURE 11. Total CO_2 emissions.

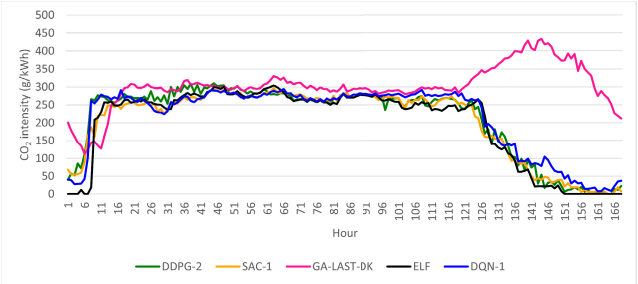


FIGURE 12. Weekly average CO_2 intensity.

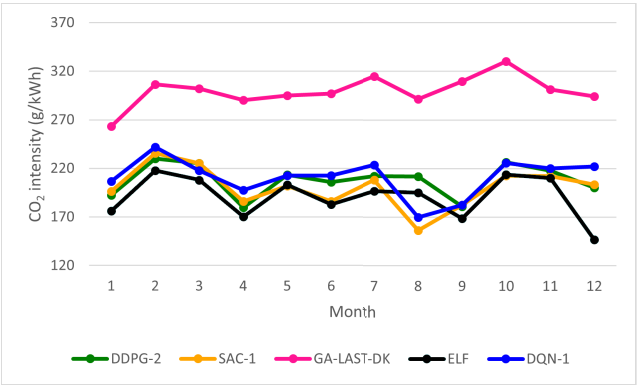


FIGURE 13. Monthly average CO_2 intensity.

D. COMPARISON OF THE ADVANTAGES AND DISADVANTAGES OF ELF, GA, AND DRL

All the strategies are promising for optimal CHP dispatch scheduling. However, notable differences exist between ELF, GA, and DRL approaches when they are implemented in a real-world application. This concluding section highlights these key differences and their implications for the stakeholders in processing facilities.

1) MAIN DISADVANTAGES OF ELF, GA, AND DRL

The main disadvantage of the ELF strategy is its limited flexibility and responsiveness to dynamic changes in the system, as it primarily focuses on matching the load demand without considering overall efficiency and cost savings. GAs suffer from computational inefficiency, particularly in large search spaces, as they require numerous iterations to converge on an optimal solution, which can be resource-intensive. DRL strategies have the drawback of requiring large amounts of training data and computational resources to develop effective policies. During the development of this research, these challenges were encountered in the tuning of DRL and GA model hyperparameters: the trade-off between model performance and convergence time had to be taken into account.

Additionally, DRL models can be sensitive to the quality of the training data and may not generalize well to scenarios significantly different from those encountered during training [63]. Specific failure cases include situations with highly variable or unpredictable energy demands, significant operational disturbances, or integration of energy components such as battery storage systems or intermittent renewable sources. However, framing the control problem as a POMDP has the potential to provide the necessary flexibility and robustness to overcome these challenges. Although demonstrated on a CHP-based system, this approach can adapt to other configurations by adjusting state-action representations and constraints and retraining with suitable operational data.

2) COMPLEXITY OF THE MATHEMATICAL FORMULATION

The mathematical formulation of ELF is relatively simple, involving straightforward rules to match power generation with load demand: therefore, ELF is easy to implement for plant managers, its ability to adapt to highly variable scenarios is limited. In contrast, GA involves more complex mathematical operations, including selection, crossover, and mutation processes, which require careful parameter tuning and can be challenging to implement effectively [64]. DRL presents the highest complexity in mathematical formulation, involving the design of reward functions, observation and action spaces, and the use of advanced optimization techniques. This complexity demands a deep understanding of machine learning principles and significant computational resources. The adoption of a POMDP approach further increases the complexity of the problem, but it can enhance the ability of the model to handle the uncertainties of a partially observable system. Finally, both DRL and GA do not inherently guarantee that hard constraints will be met [64], [65]: they rely on the proper definition of the reward and fitness function, respectively, which may require time-consuming tuning [66]. Nevertheless, a safety pipeline may be put in place to prevent the agent from executing actions that can cause mechanical stress and potential wear on the CHP components.

TABLE 8. Revenues and costs obtained by applying each algorithm in the variable electric energy price scenario. Highest values are highlighted in bold.

Category	Algorithm	*EBITDA* [€]	R_{EEC} [€]	R_{TG} [€]	AC_{el} [€]	AC_{th} [€]	*C* [€]	*BCR*
Baselines	ELF	471909	94007	13943	785970	192542	614554	1.77
	FIX	487377	85062	312229	782734	215031	907679	1.54
	DoW	474588	92314	153986	771211	195131	738055	1.64
DQN	DQN-1	504569	99394	91112	808956	205361	700255	1.72
	DQN-2	487161	97790	54837	775460	199762	640688	1.76
	DQN-6	510113	99788	98508	810321	203783	702289	1.73
DDPG	DDPG-1	498107	100126	77796	793852	206469	680137	1.73
	DDPG-2	502403	99898	56323	814674	204089	672583	1.75
	DDPG-6	501428	99511	56210	812759	202712	669765	1.75
SAC	SAC-1	510162	**101301**	65615	806902	205597	669255	1.76
	SAC-2	489451	96091	32632	793914	194505	627692	**1.78**
	SAC-6	498340	98057	49221	803680	198440	651058	1.77
GA	GA-AVG	**553008**	88873	**397073**	**875615**	**217465**	**1026018**	1.54
	GA-AVG-DK	501947	86465	320331	803360	212070	920279	1.55
	GA-AVG-PEN	493113	81936	352173	777703	204084	922783	1.53
	GA-LAST	**552957**	88869	**396926**	**875425**	**217455**	**1025718**	1.54
	GA-LAST-PEN	548928	88628	391965	869207	217322	1018193	1.54
	GA-LAST-DK	522127	92891	242695	835650	209522	858631	1.61

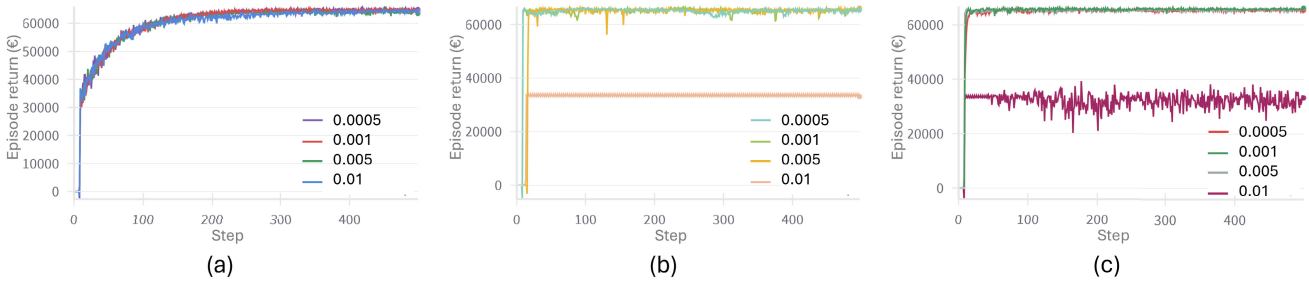


FIGURE 14. Episode return with different learning rates: (a) DQN, (b) DDPG, (c) SAC.

3) FLEXIBILITY TO ADAPT TO SYSTEM MODIFICATIONS

When it comes to adapting to changes in the mode of operation or the configurations of the system, DRL strategies require a fine tuning of their model: the consequent long maintenance times may imply the interruption of the optimal control of the CHP. However, the DRL implementation in an online setting [67] (i.e. with the continuous update of the model based on the latest data available) would be a solution to this limitation: it would provide uninterrupted optimal control of the CHP. Some challenges still exist in the online DRL setting: the development and testing of agents able to not forget useful past information and to flexibly but robustly respond to sudden changes is still tricky [68]. GA explores a wide range of potential solutions and evolves over time; however, the various solutions are pruned (i.e. forgotten) during the fitting, which implies re-tuning and re-running the algorithm to adapt it to significant changes. ELF, on the other hand, is the least adaptable as it follows predefined rules and lacks a learning mechanism.

V. CONCLUSION AND FUTURE WORK

The hourly dispatch scheduling problem of CHPs is traditionally treated as an MDP in the RL literature, assuming the validity of the Markov Property and, thus, the complete observability of the problem. However, due to the intrinsic uncertainty of future electric and thermal demands, such an assumption does not hold in real-world environments. Therefore, this work proposes the novel adoption of a POMDP approach to address the partial observability of the hourly CHP dispatch scheduling problem.

The case study of this work involved the cogeneration unit of a factory situated in Italy: the real energy demands of the factory of the year 2021 were collected for this work. Multiple GA, DRL and deterministic optimization approaches were compared.

The approach which yielded the highest yearly EBITDA is DDPG-2, with a 1% improvement compared to the ELF baseline. SAC-1 followed closely, delivering stable quarterly profits and indicating that even a single-step observation window is sufficient for robust learning.

A sensitivity analysis was performed to test the models under variable electric energy pricing conditions, which validated the robustness and adaptability of the models to fluctuating economic conditions.

Finally, an environmental analysis highlighted the role of optimized scheduling in reducing CO_2 emissions of a CHP. The ELF approach outperformed the other approaches in both environmental metrics, followed by SAC-1. Interestingly, the algorithm producing the best environmental outcomes

differed from the one achieving the highest economic performance.

Overall, this work shows that POMDP is a valid approach to model the hourly dispatch scheduling problem of CHPs. A DRL-based strategy such as SAC-1 is most beneficial in highly variable scenarios: it emerged as the best trade-off between cost and emissions while remaining robust to price fluctuations. When a lower computational complexity is required, the simpler ELF approach is a valid alternative.

However, the proposed method suffers from data scarcity, since only the data of a single year were available; thus, it may not capture long-term variability. Future works will involve the collection of a larger dataset, to be utilized for an extensive analysis of the reliability and robustness of the models. This will enable the development of more advanced DRL algorithms which leverage recurrent neural networks, to better capture the hidden dependencies in the history of the POMDP. Future work will analyze the resilience and flexibility of the strategies when inaccurate measurements and anomalous conditions occur. In this sense, the performance of DRL algorithms will be assessed in energy systems of increased complexity. Because DRL is expressly designed for high-dimensional state spaces, it is expected to surpass traditional benchmarks under these conditions as well. Finally, the associated environmental impact will be quantified and embedded into the objective function, and alternative approaches will be tested, including Model Predictive Control and other DRL exploration strategies, such as the introduction of action noise and additive bonus terms.

APPENDIX
HYPERPARAMETER TUNING

Choosing appropriate hyperparameters is essential for optimal algorithm performance. In RL, hyperparameter tuning is computationally expensive and time consuming, as the agent must repeatedly interact with the environment and update its policy over numerous timesteps [69]. Although automated hyperparameter tuning might be beneficial, performance improvements can be minor, with the risk of overfitting and high computational cost. Thus, a manual hyperparameter tuning was performed in this work and an analysis of the tuning of the key hyperparameters is reported below.

One of the most important hyperparameters in RL algorithms is the learning rate. The learning rate was tuned independently for each RL model while keeping other hyperparameters from Table 5 fixed. For the DQN algorithm (Fig. 14(a)), no significant differences emerged across tested learning rates; thus, the learning rate was set to 0.005. For DDPG (Fig. 14(b)), the rate of 0.005 led to performance degradation after 15 episodes, while other rates showed no significant differences, resulting in a selected learning rate of 0.0005. Lastly, for SAC (Fig. 14(c)), the learning rate of 0.01 yielded lower final returns, with no significant differences among other learning rates; hence, the learning rate was set to 0.005.

Regarding the GA, one of the most impactful hyperparameters is the population size. Three population sizes were tested on the training dataset using the simple GA-LAST model: 50, 100 and 200. The population size equal to 50 yielded a €96725 return, the one equal to 100 yielded a €96358 return, and the one equal to 200 yielded a €96765 return. Thus, the population size was set to 200.

ACKNOWLEDGMENT

A special thanks to Porro Gioele and Pellerey Valeria of Trigenia S.r.l. for their support of this work. An earlier version of this paper was presented in part at the 8th International Conference on Smart and Sustainable Technologies (SpliTech) [DOI: 10.23919/SpliTech58164.2023.10193518].

REFERENCES

[1] G. Ghione, V. Randazzo, A. Recchia, E. Pasero, and M. Badami, "Comparison of genetic and reinforcement learning algorithms for energy cogeneration optimization," in *Proc. 8th Int. Conf. Smart Sustain. Technol. (SpliTech)*, Jun. 2023, pp. 1–7.

[2] M. Badami, G. Chicco, A. Portoraro, and M. Romaniello, "Micro-multigeneration prospects for residential applications in Italy," *Energy Convers. Manage.*, vol. 166, pp. 23–36, Jun. 2018.

[3] S. D. Hu, *Cogeneration*. Old Tappan, NJ, USA: Prentice-Hall, 1985.

[4] T. Schütz, M. H. Schraven, S. Remy, J. Granacher, D. Kemetmüller, M. Fuchs, and D. Müller, "Optimal design of energy conversion units for residential buildings considering German market conditions," *Energy*, vol. 139, pp. 895–915, Nov. 2017.

[5] J. H. Horlock, *Combined Heat and Power*. Elmsford, NY, USA: Pergamon Books Inc., 1987.

[6] M. B. Hadi, M. Moeini-Aghtaie, M. Khoshjahan, and P. Dehghanian, "A comprehensive review on power system flexibility: Concept, services, and products," *IEEE Access*, vol. 10, pp. 99257–99267, 2022.

[7] P. Ghadimi, S. Kara, and B. Kornfeld, "The optimal selection of on-site CHP systems through integrated sizing and operational strategy," *Appl. Energy*, vol. 126, pp. 38–46, Aug. 2014.

[8] H. Hosseinian and H. Damghani, "The economic practicality of exploitation CHP(combined heat and power) to scale back prices in instance home appliance manufacturing company," in *Proc. Smart Grid Conf. (SGC)*, Nov. 2018, pp. 1–5.

[9] C. A. Wheeley, P. J. Mago, and R. Luck, "A comparative study of the economic feasibility of employing CHP systems in different industrial manufacturing applications," *Energy Power Eng.*, vol. 3, no. 5, pp. 630–640, 2011.

[10] P. Ghadimi, S. Kara, and B. Kornfeld, "Advanced on-site energy generation towards sustainable manufacturing," in *Re-Engineering Manuf. for Sustainability*, A. Y. C. Nee, B. Song, and S.-K. Ong, Eds., Singapore: Springer, 2013, pp. 153–158.

[11] M. A. Bagherian, K. Mehranzamir, A. B. Pour, S. Rezania, E. Taghavi, H. N. Afrouzi, M. Dalvi-Esfahani, and S. M. Alizadeh, "Classification and analysis of optimization techniques for integrated energy systems utilizing renewable energy sources: A review for CHP and CCHP systems," *Processes*, vol. 9, no. 2, p. 339, Feb. 2021.

[12] E. Abdollahi, H. Wang, and R. Lahdelma, "An optimization method for multi-area combined heat and power production with power transmission network," *Appl. Energy*, vol. 168, pp. 248–256, Apr. 2016.

[13] K. J. Åström, "Optimal control of Markov processes with incomplete state information," *J. Math. Anal. Appl.*, vol. 10, no. 1, pp. 174–205, Feb. 1965.

[14] L. Lei, Y. Tan, G. Dahlenburg, W. Xiang, and K. Zheng, "Dynamic energy dispatch based on deep reinforcement learning in IoT-driven smart isolated microgrids," *IEEE Internet Things J.*, vol. 8, no. 10, pp. 7938–7953, May 2021.

[15] Q. Zhang, K. Dehghanpour, Z. Wang, and Q. Huang, "A learning-based power management method for networked microgrids under incomplete information," *IEEE Trans. Smart Grid*, vol. 11, no. 2, pp. 1193–1204, Mar. 2020.

[16] H. Zhou, A. Aral, I. Brandic, and M. Erol-Kantarci, "Multiagent Bayesian deep reinforcement learning for microgrid energy management under communication failures," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 11685–11698, Jul. 2022.

[17] J. Wang, L. Li, and J. Zhang, "Deep reinforcement learning for energy trading and load scheduling in residential peer-to-peer energy trading market," *Int. J. Electr. Power Energy Syst.*, vol. 147, May 2023, Art. no. 108885.

[18] Y. Zhang, J. Hu, G. Min, and X. Chen, "Scalable and privacy-preserving distributed energy management for multimicrogrid," *IEEE Trans. Ind. Informat.*, vol. 21, no. 2, pp. 1439–1448, Feb. 2025.

[19] X. Chen, Q. Hu, Y. Zhang, M. Song, Z. Wu, and J. Xiong, "POMDP-based dispatch scheme for residential distributed energy resources under customer fatigue consideration," *IEEE Trans. Smart Grid*, vol. 15, no. 6, pp. 5665–5677, Nov. 2024.

[20] V. Mnih, K. Kavukcuoglu, D. Silver, A. Graves, I. Antonoglou, D. Wierstra, and M. Riedmiller, "Playing Atari with deep reinforcement learning," 2013, *arXiv:1312.5602*.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," 2015, *arXiv:1509.02971*.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, "Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor," in *Proc. 35th Int. Conf. Mach. Learn.*, Jan. 2018, Art. no. 18611870.

[23] J. DeCarolis, H. Daly, P. Dodds, I. Keppo, F. Li, W. McDowall, S. Pye, N. Strachan, E. Trutnevyte, W. Usher, M. Winning, S. Yeh, and M. Zeyringer, "Formalizing best practice for energy system optimization modelling," *Appl. Energy*, vol. 194, pp. 184–198, May 2017.

[24] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surveys*, vol. 51, no. 5, pp. 1–42, Aug. 2018.

[25] L. P. Kaelbling, M. L. Littman, and A. W. Moore, "Reinforcement learning: A survey," *J. Artif. Intell. Res.*, vol. 4, pp. 237–285, 1996.

[26] T. Bartz-Beielstein, M. Preuß, and H.-P. Schwefel, "Model optimization with evolutionary algorithms," in *ROOSEN, P. (Hrsg.): Emergence, Analysis, and Evolution of Structures—Concepts and Strategies Across Disciplines*. Berlin, Germany: Springer, 2010, pp. 47–62.

[27] T. M. Alabi, L. Lu, and Z. Yang, "Real-time automatic control of multi-energy system for smart district community: A coupling ensemble prediction model and safe deep reinforcement learning," *Energy*, vol. 304, Jun. 2024, Art. no. 132209.

[28] J. Dong, H. Wang, J. Yang, X. Lu, L. Gao, and X. Zhou, "Optimal scheduling framework of electricity-gas-heat integrated energy system based on asynchronous advantage actor-critic algorithm," *IEEE Access*, vol. 9, pp. 139685–139696, 2021.

[29] W. Gao and Y. Lin, "Energy dispatch for CCHP system in summer based on deep reinforcement learning," *Entropy*, vol. 25, no. 3, p. 544, Mar. 2023.

[30] A. R. Ginidi, A. M. Elsayed, A. M. Shaheen, E. E. Elattar, and R. A. El-Sehiemy, "A novel heap-based optimizer for scheduling of large-scale combined heat and power economic dispatch," *IEEE Access*, vol. 9, pp. 83695–83708, 2021.

[31] J. Hu, Y. Zou, and N. Soltanov, "A multilevel optimization approach for daily scheduling of combined heat and power units with integrated electrical and thermal storage," *Expert Syst. Appl.*, vol. 250, Sep. 2024, Art. no. 123729.

[32] B. Jia, F. Li, and B. Sun, "Knowledge-network-embedded deep reinforcement learning: An innovative way to high-efficiently develop an energy management strategy for the integrated energy system with renewable energy sources and multiple energy storage systems," *Energy*, vol. 301, Aug. 2024, Art. no. 131604.

[33] M. J. Kim, T. S. Kim, R. J. Flores, and J. Brouwer, "Neural-network-based optimization for economic dispatch of combined heat and power systems," *Appl. Energy*, vol. 265, May 2020, Art. no. 114785.

[34] B. Liu, J. Li, S. Zhang, M. Gao, H. Ma, G. Li, and C. Gu, "Economic dispatch of combined heat and power energy systems using electric boiler to accommodate wind power," *IEEE Access*, vol. 8, pp. 41288–41297, 2020.

[35] A. Lorestani and M. M. Ardehali, "Optimization of autonomous combined heat and power system including PVT, WT, storages, and electric heat utilizing novel evolutionary particle swarm optimization algorithm," *Renew. Energy*, vol. 119, pp. 490–503, Apr. 2018.

[36] L. Moretti, E. Martelli, and G. Manzolini, "An efficient robust optimization model for the unit commitment and dispatch of multi-energy systems and microgrids," *Appl. Energy*, vol. 261, Mar. 2020, Art. no. 113859.

[37] G. Moustafa, H. Alnami, S. H. Hakmi, A. M. Shaheen, A. R. Ginidi, M. A. Elshahed, and H. S. E. Mansour, "A novel mantis search algorithm for economic dispatch in combined heat and power systems," *IEEE Access*, vol. 12, pp. 2674–2689, 2024.

[38] D. Qiu, Z. Dong, X. Zhang, Y. Wang, and G. Strbac, "Safe reinforcement learning for real-time automatic control in a smart energy-hub," *Appl. Energy*, vol. 309, Mar. 2022, Art. no. 118403.

[39] Y. Ruan, Z. Liang, F. Qian, H. Meng, and Y. Gao, "Operation strategy optimization of combined cooling, heating, and power systems with energy storage and renewable energy based on deep reinforcement learning," *J. Building Eng.*, vol. 65, Apr. 2023, Art. no. 105682.

[40] A. Sundaram, "Combined heat and power economic emission dispatch using hybrid NSGA II-MOPSO algorithm incorporating an effective constraint handling mechanism," *IEEE Access*, vol. 8, pp. 13748–13768, 2020.

[41] J. Tang, M. Ding, S. Lu, S. Li, J. Huang, and W. Gu, "Operational flexibility constrained intraday rolling dispatch strategy for CHP microgrid," *IEEE Access*, vol. 7, pp. 96639–96649, 2019.

[42] H. Wu, Z. Liu, Y. He, M. Ding, B. Xu, and M. Zhang, "Two-layer optimal scheduling method for regional integrated energy system considering flexibility characteristics of CHP system," *Energy*, vol. 308, Nov. 2024, Art. no. 132970.

[43] B. Zhang, W. Hu, D. Cao, T. Li, Z. Zhang, Z. Chen, and F. Blaabjerg, "Soft actor-critic –based multi-objective optimized energy conversion and management strategy for integrated energy systems with renewable energy," *Energy Convers. Manage.*, vol. 243, Sep. 2021, Art. no. 114381.

[44] S. Zhou, Z. Hu, W. Gu, M. Jiang, M. Chen, Q. Hong, and C. Booth, "Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach," *Int. J. Electr. Power Energy Syst.*, vol. 120, Mar. 2020, Art. no. 106016.

[45] Y. Zhou, Z. Ma, J. Zhang, and S. Zou, "Data-driven stochastic energy management of multi energy system using deep reinforcement learning," *Energy*, vol. 261, Dec. 2022, Art. no. 125187.

[46] P. Zymelka and M. Szega, "Short-term scheduling of gas-fired CHP plant with thermal storage using optimization algorithm and forecasting models," *Energy Convers. Manage.*, vol. 231, Mar. 2021, Art. no. 113860.

[47] E. Parliament and C. of 25 October 2012 on Energy Efficiency, "Directive 2012/27/EU of the European parliament and of the council of 25 October 2012 on energy efficiency, amending directives 2009/125/EC and 2010/30/EU and repealing directives 2004/8/EC and 2006/32/EC text with EEA relevance," *Official J., L*, vol. 315, pp. 1–56, Nov. 2012.

[48] J. Bouwens, T. de Kok, and A. Verriest, "The prevalence and validity of EBITDA as a performance measure," *Comptabilité Contrôle Audit*, vol. Tome 25, no. 1, pp. 55–105, Apr. 2019.

[49] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021.

[50] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*. Cambridge, MA, USA: MIT Press, 2018.

[51] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra, "Planning and acting in partially observable stochastic domains," *Artif. Intell.*, vol. 101, nos. 1–2, pp. 99–134, May 1998.

[52] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, and G. Ostrovski, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[53] M. Hausknecht and P. Stone, "Deep recurrent Q-learning for partially observable MDPs," 2015, *arXiv:1507.06527*.

[54] C. J. Watkins and P. Dayan, "Q-learning," *Mach. Learn.*, vol. 8, nos. 3–4, pp. 279–292, May 1992.

[55] S. Haykin, *Neural Networks and Learning Machines* (Neural Networks and Learning Machines), vol. 10. Upper Saddle River, NJ, USA: Prentice-Hall, 2009.

[56] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proc. 27th Int. Conf. Mach. Learn. (ICML-10)*, Jun. 2010, pp. 807–814.

[57] D. Silver, G. Lever, N. Heess, T. Degris, D. Wierstra, and M. Riedmiller, "Deterministic policy gradient algorithms," in *Proc. 31st Int. Conf. Mach. Learn.*, Jun. 2014, pp. 387–395.

[58] OpenAI. *Spinning Up in Deep RL, Version: 2.3.0*. Accessed: 2024. [Online]. Available: https://spinningup.openai.com/en/latest/

[59] B. T. Polyak and A. B. Juditsky, "Acceleration of stochastic approximation by averaging," *SIAM J. Control Optim.*, vol. 30, no. 4, pp. 838–855, Jul. 1992.
[60] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft actor-critic algorithms and applications," 2018, *arXiv:1812.05905*.
[61] Ministry of the Environment and Energy Security. *EU ETS-Italia*. Accessed: 2023. [Online]. Available: https://www.ets.minambiente.it/
[62] G. Fambri, M. Noussan, M. Badami, and D. Chiaramonti, "Hourly carbon intensity of natural gas combined cycles compared to the current and future electricity mixes in Italy," *J. Phys., Conf. Ser.*, vol. 2893, no. 1, Nov. 2024, Art. no. 012034.
[63] A. Aubret, L. Matignon, and S. Hassas, "An information-theoretic perspective on intrinsic motivation in reinforcement learning: A survey," *Entropy*, vol. 25, no. 2, p. 327, Feb. 2023.
[64] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm—A literature review," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*, Feb. 2019, pp. 380–384.
[65] Y. Wang, S. S. Zhan, R. Jiao, Z. Wang, W. Jin, Z. Yang, Z. Wang, C. Huang, and Q. Zhu, "Enforcing hard constraints with soft barriers: Safe reinforcement learning in unknown stochastic environments," in *Proc. 40th Int. Conf. Mach. Learn.*, Jan. 2022, pp. 36593–36604.
[66] Y. Wei, M. Tian, X. Huang, and Z. Ding, "Incorporating constraints in reinforcement learning assisted energy system decision making: A selected review," in *Proc. IEEE/IAS Ind. Commercial Power Syst. Asia (I&CPS Asia)*, Jul. 2022, pp. 671–675.
[67] J. Schrittwieser, T. Hubert, A. Mandhane, M. Barekatain, I. Antonoglou, and D. Silver, "Online and offline reinforcement learning by planning with a learned model," in *Proc. Adv. Neural Inf. Process. Syst.*, Jan. 2021, pp. 27580–27591.
[68] Y. Li, "Reinforcement learning in practice: Opportunities and challenges," 2022, *arXiv:2202.11296*.
[69] R. R. Afshar, Y. Zhang, J. Vanschoren, and U. Kaymak, "Automated reinforcement learning: An overview," 2022, *arXiv:2201.05000*.



GIORGIA GHIONE (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in computer engineering from Politecnico di Torino, Turin, Italy, in 2018 and 2022, respectively, where she is currently pursuing the Ph.D. degree in electrical, electronics, and communications engineering with the Neuronica Laboratory Research Group.

Since 2022, she has been actively engaged in research on applying deep learning methods to time series forecasting, focusing specifically on energy demand forecasting and renewable energy generation. Her research interest includes deep reinforcement learning for energy management in microgrids.



VINCENZO RANDAZZO (Member, IEEE) received the master's degree (Hons. and cum laude) in computer engineering from the University of Palermo, in 2014, and the Ph.D. degree (cum laude) in electrical, electronics, and communications engineering from Politecnico di Torino, in 2022, with a final thesis on "Novel Neural Approaches to Data Topology Analysis and Telemedicine."

Currently, he is an Assistant Professor with the Department of Electronics and Telecommunications, Politecnico di Torino, working on the development of new methodologies and devices that exploit the Internet of Things (IoT) paradigm and artificial intelligence algorithms to monitor industrial processes and related consumption both in terms of raw materials and energy.

Dr. Randazzo is a member of the Italian Society of Neuronic Networks (SIREN) and the IEEE CIS-GAC AI in Healthcare Task Force and IEEE CIS GAC. He is the contact person for the CIS (Italy Section) Chapter of the Affinity Group Young Professional for Italy Section. He was the IEEE Young Professionals Vice-Chair for Italy Section and the Former Treasurer of Politecnico di Torino IEEE Student Branch. He is the YP Representative for IEEE Italy Section.



EROS PASERO (Senior Member, IEEE) has been a Professor of electronics with Politecnico di Torino, Turin, Italy, since 1991, after a four year appointment as a Professor of electronics engineering with the University of Rome. He is the author of more than 150 international articles. In 1990, he established the Neuronica Laboratory, where hardware and software neurons and synapses are studied for practical applications; innovative wired and wireless sensors are also developed for biomedical, environmental, and automotive applications. Data coming from sensors are post processed by means of artificial neural networks.

He served as the President of SIREN, the Italian Society for Neural Networks, from 2012 to 2022; he was the v. General Chair of IJCNN2000, Como, the General Chair of SIRWEC2006, Turin, and the General Chair of WIRN2015, WIRN2016, WIRN2017, WIRN 2018, WIRN 2019, WIRN 2020, and WIRN 2022, Vietri. He was a IEEE Keynote Speaker at the 2014 Symposium series on Computational Intelligence, Orlando, FL, USA; and a Distinguished Lecturer of the 2016 IEEE Medical Information Summer School and a Distinguished Lecturer of the 2017 IEEE School "Smarter Engineering for Industry 4.0" Keynote Speaker at I2MTC2023, Kuala Lumpur. He is the IEEE I&M Society Distinguished Lecturer, from 2021 to 2026.



MARCO BADAMI was born in Turin, Italy. He received the master's degree (Hons.) in mechanical engineering from Politecnico di Torino, and the Master of Science degree in management from the London School of Economics, London, U.K.

He began his career in academia as a Researcher in fluid machines with the Department of Energy, Politecnico di Torino, in 1990. In 1999, he was a Visiting Lecturer with Imperial College London for one year, and was appointed as an Associate Professor, in 2000. Currently, he is a Full Professor of energy and environmental systems with the Department of Energy, Politecnico di Torino. Over his career, he has led numerous research and consultancy projects for both public and private institutions. He has published extensively and led significant projects as a Principal Investigator, including the European H2020 PLANET Project and the EU HEGEL Project of the 6th Framework Program. He also directed consultancy teams for defining regional energy balances and conducting feasibility studies on biomass district heating systems. He has collaborated with prominent industrial organizations, such as the FIAT Research Center, Ferrari, Ferrero, MV Agusta, and Stogit-Snam. His primary research interests include complex energy systems, optimization and control of cogeneration plants, and the rational use of energy.