Optimizing integrated lot sizing and production scheduling in flexible flow line systems with energy scheme: A two level approach based on reinforcement learning

Mohamed Habib Jabeur, Sonia Mahjoub, Cyril Toublanc, Veronique Cariou

Computers & Industrial Engineering 190 (2024) 110095

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Optimizing integrated lot sizing and production scheduling in flexible flow line systems with energy scheme: A two level approach based on reinforcement learning

Mohamed Habib Jabeur [a,*], Sonia Mahjoub [b], Cyril Toublanc [c], Veronique Cariou [a]

[a] Oniris, INRAE, STATSC, 44300 Nantes, France
[b] Oniris, Nantes université, LEMNA, CS 82225, 44322 Nantes, France
[c] Oniris, Nantes université, CNRS, GEPEA, UMR 6144, F-44000 Nantes, France

ARTICLE INFO

Keywords:
Lot sizing
Production scheduling
Flexible flow line
Renewable energy
Time-of-use prices
Multi-agent system

ABSTRACT

Many production environments are faced with the need to simultaneously determine the planning of lot sizing and the scheduling of production sequences while ensuring cost minimization. This issue becomes even more complex when integrating multiple energy sources with the goal of a low-carbon economy. To address this challenge, this paper proposes an integrated lot sizing and flexible flow line production scheduling model under a time-of-use pricing scheme. In addition, the model takes into account conventional grid power, on-site renewable energy sources, and an energy storage system. The associated objective function is solved adopting a two-level approach to optimize energy costs while maintaining production throughput and meeting customer demand. The implementation relies on reinforcement learning capabilities to tackle complexity issues. The proposed approach is evaluated on a benchmark case and its results are compared with those obtained with First-In-First-Out heuristic, genetic algorithm and CPLEX. These results highlight the promising aspect of the proposed approach in terms of performance.

1. Introduction

Within the scope of industry 4.0, improving industrial energy efficiency has become a key focus for many companies aiming to achieve their carbon-neutrality goals and sustain their competitive advantage. According to Eurostat (2019), the industrial sector accounted for 25 % of the total energy consumption in the European Union (EU). To reduce energy costs and decrease greenhouse gas (GHG) emissions, demand side management (DSM) and on-site renewable energy appear to be promising solutions. DSM is an energy policy implemented by utilities to modify users' energy consumption patterns, ultimately enhancing power grid efficiency (Duarte et al., 2020). DSM can be effectively implemented through demand response (DR) programs, which seek to alter energy demands in response to economic incentives (Wang et al., 2020). The DR program has emerged as an effective policy for industries to reduce their energy costs without compromising their production throughput. In practice, this program involves an increase in production rates during off-peak electricity demand hours and a decrease in production rates during peak grid demand hours (Basán et al., 2020). The DR can be considered as a production scheduling problem whose objective is to minimize both the energy consumed and the associated operational costs by prioritizing production during off-peak periods (Kelley, Baldick, & Baldea, 2019, 2020; Kelley, Pattison, Baldick, & Baldea, 2018). Consequently, a flexible and dynamic manufacturing process supported by an efficient energy system is of paramount interest to adapt to electricity market price signals. For this purpose, on-site renewable energy offers substantial flexibility and opportunity for manufacturers to reduce energy costs in the face of time-varying electricity prices. Furthermore, the integration of renewable energy sources (RES) into the manufacturing process is an effective sustainable strategy for reducing GHG emissions. The adoption of renewable energy technologies in the EU more than doubled between 2004 and 2017, and renewable energy production is expected to continue increasing in the coming years (Eurostat, 2018;2019). RES generation, such as wind and solar energy, exhibits high variability due to its dependence on weather conditions. The inherent uncertainties associated with these renewable

* Corresponding author.
E-mail addresses: mohamed-habib.jabeur@oniris-nantes.fr (M.H. Jabeur), sonia.mahjoub@oniris-nantes.fr (S. Mahjoub), cyril.toublanc@oniris-nantes.fr (C. Toublanc), veronique.cariou@oniris-nantes.fr (V. Cariou).

https://doi.org/10.1016/j.cie.2024.110095

Available online 28 March 2024

energy resources can lead to inaccuracies in potential scheduling solutions. To address the intermittent nature of renewable energy, an energy storage system (ESS) can store excess renewable energy and deploy it when needed.

In the light of the aforementioned work, this study proposes a dynamic sustainable production scheduling model that aims to optimize energy costs while maintaining production throughput. Specifically, the model considers an integrated lot sizing on a flexible flow line (FFL) scheduling problem with sequence-dependent setups under a time-of-use (TOU) pricing scheme. The proposed model considers a conventional grid, on-site photovoltaic solar panels and an ESS as energy sources. To the best of our knowledge, only a few research studies investigated the integration of lot sizing and sustainable production scheduling on a FFL with energy considerations. Given the relevance of such integration for many energy-intensive manufacturing industries (e.g. steel-making, chemical and food industries), a novel optimization framework is proposed to fill this gap, integrating energy management policies and renewable energy into the FFL environment. In addition, the proposed optimization model takes into account two interrelated decision levels, namely lot sizing and scheduling. A new resolution approach based on an interactive two-level algorithm is implemented to circumvent the complexity issues. At the lot sizing decision level, the algorithm aims at generating optimal lot sizes while trading off RES, ESS and inventory costs and economic benefits derived from energy saving. Regarding the scheduling decision level, an innovative and dynamic cooperative multi-agent approach is developed to address the complexity of machine allocation and product sequencing problems. This approach, based on Q-learning methodology, leads to consider that the Q-agents are trained in a dynamic FFL environment with the objective of meeting customer's demand while optimizing energy and operational costs.

The rest of the research is organized as follows. In Section 2, a concise overview of the literature is provided. Section 3 presents the problem formulation and its mathematical model. Section 4 outlines the proposed resolution approach based on reinforcement learning. In Section 5, results obtained by applying the proposed approach are detailed. In particular, performance aspects, including optimization outcomes and computational efficiency, are analyzed and discussed. Section 6 is dedicated to managerial insights. Finally, conclusions are presented in section 7.

2. Literature review

2.1. Integrated lot sizing and production scheduling (ILSPS) problem on flexible flow lines (FFL)

Operational production planning aims to determine the number of product lots to be produced (lot sizing) and the sequence of operations for their production (scheduling). These two critical decisions are generally considered independently in cases where there is no interdependence between machine setup and operation sequencing (Carvalho et al., 2022). However, when costs and machine setup time are influenced by the sequence of operations, the scheduling decision directly affects the lot sizing decision. Thus, integrating lot sizing and production scheduling decisions into a unified problem may generate better production planning. Within this scope, some studies investigated the integrated lot sizing and production scheduling (ILSPS) problem with different types of machine architectures including single machines, parallel machines, flow shops, and job shops. For example, Carvalho and Nascimento (2022) studied the ILSPS problem given non-identical parallel machines with non-triangular sequence-dependent setup costs and times. The authors proposed different matheuristics based on relax-and-fix, relax-and-optimize, path relinking and kernel search methods to efficiently solve the ILSPS problem on parallel machines. The computational results highlighted the better performance of the proposed algorithms compared to the CPLEX solver. Xiao et al. (2015) investigated a lot sizing and production scheduling problem on parallel machines with sequence-dependent setup times. To this end, the authors advocated a hybrid heuristic that combines simulated annealing with a Lagrangian-based algorithm. This hybrid approach has proven to outperform other heuristic algorithms. Mahdieh et al. (2011) also developed mathematical models for different integrated lot sizing and production scheduling on flexible flow lines (FFL). The experimental results highlighted the complexity of the problem, as optimality tests could only be performed on small instances. Babaei et al. (2014) examined the capabilities of a genetic algorithm applied on a multi-product integrated lot sizing and production scheduling problem on a flow line. The computational experiments demonstrated the effectiveness of this algorithm in problem solving. In the same vein, the present study addresses the ILSPS problem on FFL, which is commonly encountered in various production systems e.g. food processing and textile industries. Despite the importance of this approach in meeting the needs of the industries, the literature on ILSPS problem on FFL remains scarce even if it proposes various optimization methods, including exact methods, heuristics and *meta*-heuristics. Exact mathematical optimization methods aim to find an optimal solution in the entire solution space. However, due to the NP-hardness of those methods, they are unfortunately unable to solve larger problems within a reasonable period (Li, Pan and Liang, 2010, Lei et al., 2022). Conversely, heuristic and *meta*-heuristic methods have shorter computation times at the expense of less optimal scheduling results compared to exact methods. To make a balance between the quality result and computational time, several artificial intelligence approaches such as reinforcement learning (RL) (Van Moffaert and Nowe, 2014; Zou et al., 2021) have been investigated to solve complex combinatorial optimization problems (Mazyavkina et al., 2021). For instance, Lei et al. (2022) developed a deep reinforcement framework for solving a flexible job shop scheduling problem. The proposed framework involved two sub-policies designed either for job operation actions or for a machine action. The computational experiment results have proven that this deep reinforcement learning method outperforms heuristic and *meta*-heuristic methods in solution quality and running time, respectively. Wang et al. (2021) proposed a dynamic scheduling approach based on deep reinforcement learning to deal with uncertainties and dynamic events in a job shop scheduling environment. Reyna et al. (2019) explored a flow shop scheduling problem with an adapted reinforcement learning approach involving a heuristic to initiate the search space with a potential solution. Their approach provided high-quality solutions within short computational times. In another study, Lin et al. (2022) developed a Q-learning technique to guide the selection of a heuristic from a predesigned low-level heuristic for solving a semiconductor final testing scheduling problem. Yan et al. (2022) employed a combination of reinforcement learning and digital twin technology to solve a dynamic flexible job shop-scheduling problem with flexible maintenance policies. Numerical experiments validated the feasibility of the obtained scheduling results and highlighted the associated performance compared to three competitor algorithms. Finally, Cheng et al. (2022) proposed a Q-learning-based genetic algorithm to obtain near-optimal solutions for a complex production-scheduling problem.

As detailed above, several studies have already proven the relevance of RL techniques to solve production-scheduling problems such as flow shop and job shop scheduling problems. Notwithstanding, these techniques have not yet been considered to address the ILPSP in an FFL environment. In response to this research gap, this study proposes to implement a RL approach with an innovative architecture to tackle the ILSPS problem in an FFL environment. The efficiency of this approach is further demonstrated through comparisons with several conventional methods.

2.2. Production scheduling with energy consideration

To address the issue of energy costs, several studies were conducted on production scheduling with energy considerations to optimize energy

consumption in manufacturing systems. In this regard, Shrouf et al. (2014) developed a mathematical model to minimize energy consumption costs for a single machine production scheduling considering variable energy prices. Masmoudi et al. (2017) proposed two heuristics to solve a flow shop scheduling problem with energy consideration. Similarly, Rodoplu et al. (2020) developed a hybrid scheduling model for a single-item flow shop production scheduling, aiming to reduce energy consumption. Li et al. (2018) investigated a hybrid flow shop scheduling problem with setup energy consumptions using an energy-aware multi-objective optimization algorithm. Alternatively, making production scheduling sustainable can also be achieved using onsite renewable energy. Indeed, the use of renewable energy resources not only provides a solution to meet peak electricity demands but also helps to limit GHG emissions. The integration of renewable energy sources into production systems has been addressed several times in recent literature. Golari et al. (2017) formulated a lot sizing problem that integrates a renewable energy source as an optimization programming model. Wang et al. (2020) investigated a production scheduling problem that incorporates onsite renewable energy, the main grid, and an energy storage system (ESS). They elaborated a two-stage multi-objective stochastic model for flow shops with sequence-dependent setups to simultaneously minimize total weighted completion time and energy costs. Trevino-Martinez et al. (2022) introduced an optimization framework that incorporates renewable energy sources and an ESS into a job shop scheduling problem. Duarte et al. (2020) addressed a multi-process production planning problem that incorporates intermittent renewable energy sources and an ESS. They proposed a mathematical programming model and an algorithm to optimize production scheduling and energy management system decisions by considering grid power, onsite renewable energy sources, an ESS, and intermediate buffers. Their results show that the integration of these policies ensures a flexible manufacturing process. In this context, Peinado-Guerrero and Villalobos (2022) considered the use of an ESS and intermediate buffers in a manufacturing facility to reduce energy consumption during peak demand periods. Sun et al. (2014) developed a mathematical model that aims to maintain a good tradeoff between the economic costs of intermediate buffer systems and the potential cost savings generated from TOU pricing.

In summary, this study focuses on addressing the ILSPS problem in an FFL (Flexible Flow Line) environment while considering energy efficiency. Compared to previous research, this study makes the following main contributions:

- A novel mixed-integer linear model is developed to accurately represent the ILSPS problem in the FFL environment under a Time-of-Use (TOU) pricing scheme. The mathematical model is specifically adapted to incorporate an onsite photovolatic solar power supply.
- To achieve a balance between the costs associated with buffers, Energy Storage Systems (ESS), Renewable Energy Sources (RES), and TOU pricing, a two-level method based on reinforcement learning is proposed.
- The reinforcement learning approach is developed to effectively handle the complexity of the model and provide good solutions.
- Numerical experiments are conducted to demonstrate the effectiveness of both the proposed model and the resolution approach.

3. Problem description and modeling

3.1. Problem description

This paper addresses the ILSPS problem within the FFL scope for a multi-products processing system powered by different energy sources: grid power, solar power generated by onsite PV panels, and an ESS (Fig. 1). To simplify the planning framework, the availability of solar power is considered here as a certain parameter over the planning horizon. This means that the planning process assumes constant and predictable solar energy generation over all micro-periods. The system comprises multiple consecutive processing stages, where each process stage has one or more non-identical parallel machines. The scheduling horizon is finite and divided into T macro-periods, which are subdivided into a finite number of micro-periods, denoted as F. To ensure that all product demands are met within the planning horizon, there is no micro-period delay between different processes. This means that a product produced at process p becomes immediately available for production at the next process. The manufacturing system also involves buffering routines, where products can be stored at process p and made available for the next process in the next micro-period. Buffer deployment is crucial for reducing product demands during peak energy periods, considering the application of TOU pricing schemes.

In this study, both sequence-dependent setup times and costs are considered. The cost and time necessary to set up for product j after product i on machine m are assumed to be different from those needed to set up for product j if it is produced before product i on the same machine. Furthermore, these setups are supposed to be cost and time

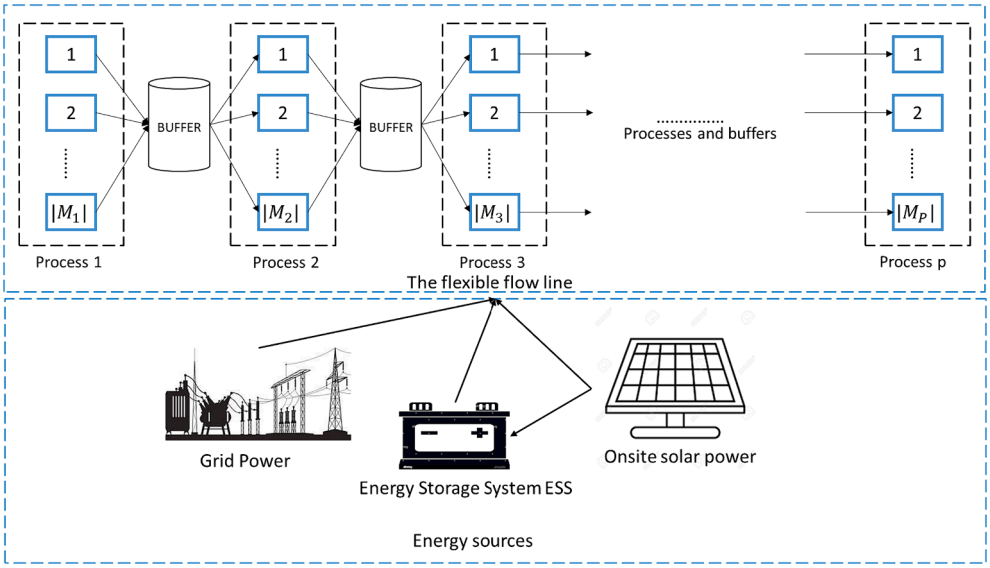

Fig. 1. The flexible flow line with energy sources.

dependent on the machine. It is supposed that: 1) for each beginning of the scheduling horizon, machines are set up for a specific product and this setup can be maintained if there is no other product to process; 2) each machine m can produce at most two different products in one micro-period; 3) each machine m is limited in the number of products it can produce within a single micro-period. The machine's capacity during a micro-period is proportional to its processing time and the duration of that micro-period.

3.2. Mathematical model

In this section, we present a Mixed Integer Linear Program (MILP) to address the ILSPS problem FFL, taking into account energy constraints. The primary objective is to minimize both production and energy costs. To achieve this goal, two crucial decisions are optimized simultaneously. Firstly, we determine the best production lot sizes required to meet the demand. Secondly, we identify the most efficient production sequence that minimizes setup costs, holding costs, and energy costs. Table 1 depicts the parameters including sets and indexes for the scheduling horizon, machines, products, and processes. Table 2 defines machine capacities, setups, costs related to production and energy consumption, and energy capacities. Finally, the decision variables, including production, buffers, setup, and energy variables are depicted in Table 3.

3.3. Objective function

The objective function is divided into two parts, the first one corresponding to the production costs and the second one to the energy costs. The production costs, denoted as C_{sched}(Eq. (1a)), consist of setup costs and holding costs. In C_{sched} , no associated cost with an empty setup $\left(y^{i,i}_{m,p,t_f}\right)$ is assumed. The energy costs, denoted as C_{energy} (Eq. (1b)), take into account the costs associated with energy consumption from various sources, including grid power under the TOU pricing scheme, onsite solar power, and ESS. The objective function defined in Eq. (1c) aims to minimize the total sum of production and energy costs of the manufacturing system during the scheduling horizon.

$$C_{sched} = \sum_{f\in F}\sum_{t\in T}\sum_{p\in P}\sum_{m\in M_p}\sum_{i\in N}\sum_{j\in N\setminus\{i\}}\left(C^{i,j}_{m,p}.y^{i,j}_{m,p,t_f}\right) + \sum_{t\in T}\sum_{p\in P}\sum_{i\in N}\left(C^{buffer}_{i,p,t}.Q^{i}_{p,t_f}\right) \tag{1a}$$

$$C_{energy} = \sum_{f\in F}\sum_{t\in T}\left(g_t.ec_{t_f} + r_{t_f}.er_{t_f} + C^{+}.r^{+}_{t_f} + C^{-}.r^{-}_{t_f}\right) \tag{1b}$$

$$C = \min C_{energy} + C_{sched} \tag{1c}$$

To achieve the above objective, the following two types of constraints need to be satisfied:

3.3.1. Lot sizing and production scheduling constraints

In this part of model, we consider constraints related to lot sizing and production scheduling features, such as demand meeting, process and buffer flow, cycle time, buffering capacities and finally setup constraints. An illustrative representation of the production flow over processes and micro-periods is depicted in Fig. 2. Specifically, the quantity

Table 1
Sets and indexes.

Symbol	Description
T,t	Set and index of macro-periods in production horizon.
F,f	Set and inex of micro-periods
P,p	Set and index of processes.
M_p,m	Set and index of machines in given process p
N, i, j	Set and indexes of products.

Table 2
Parameters.

Symbol	Description
$D_{i,t}$	Demand for product i at the end of macro-period $t(f = \|F\|)$.
$\tau^{i}_{m,p}$	Processing time of product i with machine m of process p.
$s^{i,j}_{m,p}$	Setup time from product i to product j with machine m of process p.
$C^{i,j}_{m,p}$	Setup cost from product i to product j with machine m of process p.
$C^{buffer}_{i,p,t}$	Unit cost per item buffered in process p at macro-period t.
d	Micro-period duration.
M	A big real number.
N_p	buffer capacity for process p.
s_{max}/s_{min}	Maximum / Minimum storage capacity for the ESS.
s_0	Initial energy level of the ESS.
R^{+}/R^{-}	Charging and discharging capacities of the ESS.
C^{+}/C^{-}	ESS charging and discharging cost.
$E^{on}_{i,m,p}$	Required power to produce one unit of product i in machine m of process p.
$E^{off}_{m,p}$	Consumed power in one time unit when machine m of process p is in setup mode.
G_{t_f}	Available renewable energy at micro-period f from macro-period t.
g_t	MWh Conventional energy price at macro-period t.
$\hat{g}_t$	Nominal value of conventional energy price at macro-period t.
r_{t_f}	MWh PV power cost at micro-period f from macro-period t.
η^{+}/η^{-}	Charging/discharging efficiency of the ESS.
α	The learning rate
γ	The discount factor

Table 3
Variables.

Symbol	Description
x^{i}_{m,p,t_f}	Quantity of product i produced by machine m of process p at micro-period f of macro-period t.
XT^{i}_{p,t_f}	Amount of product i produced in process p at micro-period f of macro-period t and transferred to next process.
XS^{i}_{p,t_f}	Amount of product i produced in process p at micro-period f of macro-period t and waiting for production at next process.
Q^{i}_{p,t_f}	Amount of product i buffered in process p in micro-period f.
QT^{i}_{p,t_f}	Amount of product i buffered in process p and progress to next process $p+1$ at micro-period $f+1$.
QS^{i}_{p,t_f}	Amount of product i buffered in process p and still buffered in process p at the next micro-period $f+1$.
$y^{i,j}_{m,p,t_f}$	1, if there is a setup from product i to product j in machine m of process p in the micro-period f of macro-period t, otherwise $= 0$.
w^{i}_{m,p,t_f}	1, if machine m of process p is setup for product i at the beginning of micro-period f, otherwise $= 0$.
ESS_{t_f}	ESS energy level at micro-period f of macro-period t.
$r^{+}_{t_f}/r^{-}_{t_f}$	Amount of power to charge /discharge the ESS at f of macro-period t.
er_{t_f}	Consumed energy from onsite solar panels at micro-period f of macro-period t.
ec_{t_f}	Grid conventional energy consumed at micro-period f of macro-period t.

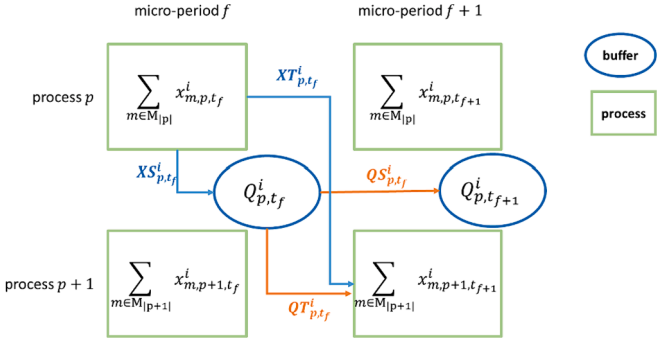

Fig. 2. Process and buffer flow.

of product i processed in process p during micro-period f $\left(\sum_{m\in M_p} x^i_{m,p,t_f}\right)$ can follow one of two paths; it can either be forwarded to process p+1 during micro-period f+1 $\left(XT^i_{p,t_f}\right)$, or it can be held in buffer of process p, awaiting further processing in subsequent micro-periods (XS^i_{p,t_f}). The buffer flow follows the same logic as the production flow using variables QT^i_{p,t_f} and QS^i_{p,t_f}. The production process described above is represented in constraints (2a)-(2n).

$$\sum_{f\in F}\sum_{m\in M_{|P|}} x^i_{m,|P|,t_f} + Q^i_{|P|,t-1_{|F|}} - Q^i_{|P|,t|F|} \geq D_{i,t}, \forall i \in N, \forall t \in T \tag{2a}$$

$$\sum_{m\in M_p} x^i_{m,p,t_f} = XT^i_{p,t_f} + XS^i_{p,t_f}, \forall i \in N, \forall p \in P, \forall t \in T, \forall f \in F \tag{2b}$$

$$Q^i_{p,t_f} = QT^i_{p,t_f} + QS^i_{p,t_f}, \forall i \in N, \forall p \in P, \forall t \in T, \forall f \in F \tag{2c}$$

$$\sum_{m\in M_p} x^i_{m,p,t_f} = XT^i_{p-1,t_{f-1}} + QT^i_{p-1,t_{f-1}}, \forall i \in N, \forall p \in \{2,\cdots,|P|\}, \forall t \in T, \forall f \in F \tag{2d}$$

$$Q^i_{p,t_f} = QS^i_{p,t_{f-1}} + XS^i_{p,t_f}, \forall p \in P, \forall i \in N, \forall t \in T, \forall f \in F \tag{2e}$$

$$Q^i_{p,t_1} = Q^i_{p,t-1_{|F|}} + XS^i_{p,t_1}, \forall p \in P, \forall i \in N, \forall t \in \{2,\cdots,|T|\} \tag{2f}$$

$$\tau^i_{m,p}.x^i_{m,p,t_f} + \sum_{j\in N}\left(s^{i,j}_{m,p}.y^{i,j}_{m,p,t_f} + \tau^j_{m,p}.x^j_{m,p,t_f}\right) \leq d, \forall i \in N, \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \tag{2g}$$

$$\sum_{i\in N} Q^i_{p,t_f} \leq N_p, \forall p \in P, \forall t \in T, \forall f \in F \tag{2 h}$$

$$x^i_{m,p,t_f} \leq M.\left(w^i_{m,p,t_f} + \sum_{j\in N} y^{j,i}_{m,p,t_f}\right), \forall i \in N, \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F, \tag{2i}$$

$$\sum_{i\in N}\sum_{j\in N} y^{i,j}_{m,p,t_f} \leq 1, \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \tag{2j}$$

$$\sum_{i\in N} w^i_{m,p,t_f} = 1, \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \tag{2 k}$$

$$w^i_{m,p,t_f} + \sum_{j\in N} y^{i,j}_{m,p,t_f} + \sum_{j\in N} y^{j,i}_{m,p,t_{f+1}} \leq 1 + w^j_{m,p,t_{f+1}}, \forall i \in N, \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \tag{2 l}$$

$$x^i_{m,p,t_f}, XT^i_{p,t_f}, XS^i_{p,t_f}, Q^i_{p,t_f}, QT^i_{p,t_f}, QS^i_{p,t_f}, I^i_{t_f} \geq, 0, \forall i \in N, \forall p \in P, \forall m \in M_p, \forall t \in T, \forall f \in F \tag{2 m}$$

$$y^{i,j}_{m,p,t_f},\ w^i_{m,p,t_f} \in \{0,1\}, \forall i \in N, \forall p \in P, \forall m \in M_p, \forall t \in T, \forall f \in F \tag{2n}$$

In this model, constraint (2a) ensures that all final demands are met by the end of the scheduling horizon. Constraint (2b) models the production of product i on machine m in process p during micro-period f. Then, x^i_{m,p,t_f} is partitioned into segments: one is earmarked for storage, while the other will progress to next process in the upcoming micro-period. Constraint (2c) employs a similar rationale to (2b) and reflects the buffering level of the process p at micro-period f. Constraint (2d) maintains the production flow balance in process p at micro-period f, while constraints (2e) and (2f) ensure the flow balance of buffering in each process p. However, they apply over distinct timeframes. Constraint (2e) focuses on maintaining balance within the same macro-period, whereas (2f) ensures equilibrium across two consecutive macro-periods. Constraint (2g) models the production cycle time capacity for each machine and its setup times within a micro-period. In this sense, it states that a given machine m is limited to producing a maximum of two distinct items within a single micro-period f. Constraint (2h) defines the buffering capacity of each process p. Constraint (2i) ensures that the appropriate setup of a product i is performed before production, while constraint (2j) specifies that in a single micro-period i, only one changeover is allowed on a machine m. Constraint (2k) ensures that only one setup state can be defined at the beginning of each period. It also guarantees that each machine m has an initial setup configuration $\left(w^i_{m,p,t=1_{f=1}} = 1\right)$. Constraint (2l) relates the changeover and setup state variables. Thus, the setup state of the machine switches immediately from i to j when a changeover occurs from product i to product j. This transition happens in three possible cases: at the beginning of the micro-period, during the micro-period, or at the end of the micro-period. Fig. 3 provides an overview of how the changeover and the setup state variables interact and change in response to these three cases. Finally, non-negativity and binary requirements are stated in (2 m) and (2n). Overall, these constraints collectively aim to ensure efficient production planning and scheduling, while balancing the demand for final products with the capacity of the production system.

3.3.2. Energy constraints

The integration of renewable energy sources into an electrical distribution network may prove insufficient without the use of an ESS. Indeed, its purpose is to facilitate the integration of PV power into the manufacturing electrical network and to provide power during subsequent micro-periods. Charging and discharging transactions of the ESS, as well as the dynamic of the energy supply system are expressed in (3a)-(3 h).

$$S_{min} \leq ESS_{t_f} \leq S_{max}, \forall\ t \in T, \forall f \in F \tag{3a}$$

$$ESS_{t_f} = ESS_{t_{f-1}} + \eta^+.r^+_{t_f} - r^-_{t_f}\frac{1}{\eta^-}, \forall\ t \in T, \forall f \in F \tag{3b}$$

$$ESS_{t_1} = ESS_{t-1_{|F|}} + \eta^+.r^+_{t_f} - r^-_{t_f}\frac{1}{\eta^-}, \forall\ t \in \{2,..,|T|\} \tag{3c}$$

$$r^+_{t_f} \leq R^+, \forall\ t \in T, \forall f \in F \tag{3d}$$

$$r^-_{t_f} \leq R^-, \forall\ t \in T, \forall f \in F \tag{3e}$$

$$\sum_{i\in N}\sum_{p\in P}\sum_{m\in M_p}\left(E^{on}_{m,p}.x^i_{m,p,t_f}.\tau^i_{m,p} + \sum_{j=1}^{N} y^{i,j}_{m,p,t_f}.s^{i,j}_{m,p}.E^{off}_{m,p}\right) = ec_{t_f} + er_{t_f} + r^-_{t_f}, \forall t \in T, \forall f \in F \tag{3f}$$

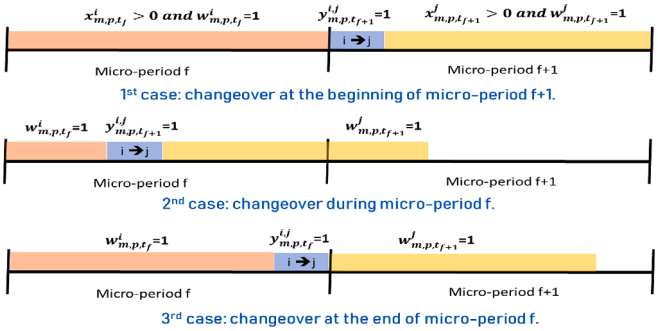

Fig. 3. Relations between binary variables according to changeover.

$$er_{t_f} + r^{+}_{t_f} \leq G_{t_f}, \forall\ \mathrm{t} \in \mathrm{T}, \forall f \in F \quad (3\ \mathrm{g})$$

$$ESS_{t_f},\ r^{+}_{t_f}, r^{-}_{t_f},\ ec_{l,t_f}, er_{t_f} \geq 0,\ \forall tT, \forall f \in F \quad (3\ \mathrm{h})$$

Constraint (3a) ensures that the level of charge in the ESS remains within specified lower and upper bounds. Constraints (3b) and (3c) update the power level of the ESS in each micro-period. Constraints (3d) and (3e) express the charge and discharge capacities of the ESS, respectively. Constraint (3f) models the energy requirements of the manufacturing system and specifies the sources of energy, including PV solar power, grid energy under TOU policies, and ESS energy. Constraint (3g) shows how the generated on-site renewable energy is distributed between the manufacturing process and the ESS. Finally, a non-negativity requirement is indicated in constraint (3h).

4. Solution approach: A two-level method

The formulated mathematical model aims to optimize the integrated lot sizing and production scheduling, taking into account several features, including energy conditions, TOU prices, production and inventory capacities, satisfying final demand, and changeover metrics on production machines. These various constraints impose limits on feasible solutions, which can significantly increase the computational time required to reach the optimum solution. In addition, the optimal solution must balance energy and production strategies at each macro-period. Furthermore, the integration of buffers, RES, and ESS into production has a significant impact on overall production cost, which, in turn, affects lot sizes and production schedules. To meet these challenges and achieve a good solution within a reasonable timeframe, we propose a novel two-level framework, as summarized in Fig. 4.

The first level optimization (FLO) is designed to handle the integrated lot sizing and production scheduling. It focuses on optimizing the allocation of production quantities and schedules while taking into account the constraints mentioned above. The objective is to determine a good production plan that optimizes costs and meets demands. However, the challenge in the FLO resides in determining the trade-off between TOU energy prices, RES, ESS, and buffer use, when solving the proposed model. All these factors have complex interactions, and striking the appropriate compromise may be challenging. To simplify the problem, we assume that there is no buffer ($N_p = 0$) and no TOU policy ($g_t = \widehat{g}_t$). Using this type of approximation may make it easier to solve the MP1, within the FLO framework. After obtaining the initial production schedule, we introduce the second level optimization (SLO) which is devoted to optimizing the balance between peak electricity prices, buffers, ESS, and RES. It looks for the economic strategy to utilize energy resources, taking advantage of peak electricity prices and ensuring efficient management of buffers, ESS, and RES. Once, the best trade-off is determined, the FLO is then resolved to check the schedules obtained in the first level and adjust them as necessary. The proposed resolution framework combines these two levels of optimization to overcome problem complexity. The following subsections describe the development of this framework.

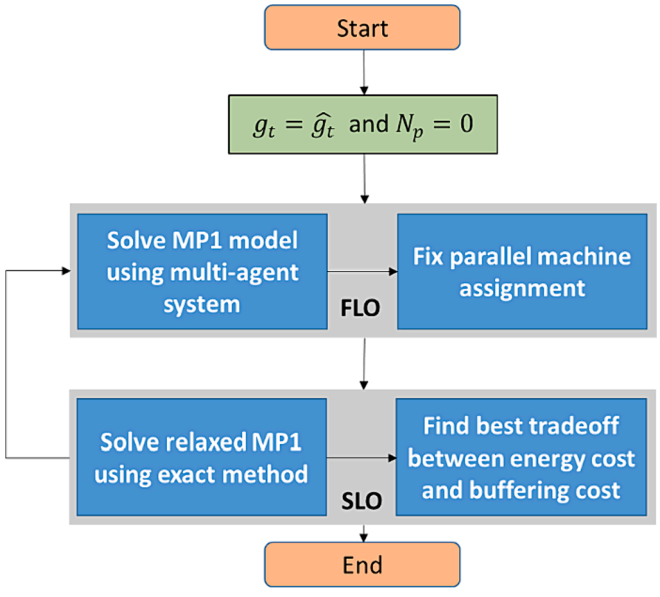

Fig. 4. Two-level approach framework.

4.1. First level optimization FLO: Production scheduling

The primary objective of the FLO is to determine the best sequence and assign the most suitable items to each machine, while considering stable lot sizes. Although the assumptions outlined in Section 4 were introduced to streamline the FLO resolution process, the optimization problem remains complex and is identified as NP-Hard (Alves et al., 2021). To cope with NP-hardness, a reinforcement learning (RL) approach based on a multi-agent system (MAS) is introduced to solve the FLO. In particular, the RL method aims to determine the best values for the binary variables w^{i}_{m,p,t_f} and $y^{i,j}_{m,p,t_f}$ within reasonable computational time.

4.1.1. Multi-agent system

The first-level model corresponds to a Multi-Agent System (MAS), where agents collaborate to achieve specific goals by interacting within a shared environment. In our case, the environment is related to the FFL system, which defines the production parameters. These include the number of agents, the connections between them, the number of products, and the characteristics of each machine, in particular its setup time and cost. By exchanging experiences and knowledge among agents, the MAS aims to optimize costs and improve the quality of solutions. Agents interact with the FFL environment through perception and action, learning how to map different situations to appropriate actions. Since the FFL problem consists of two sub-problems, namely the assignment problem and the sequencing problem, the learning process is divided into two stages. In the first stage, the agents learn how to select the most suitable machine for the processing of each product. In the second stage, the agents learn the best processing sequence for each product. With respect to these stages, the MAS consists of two types of agents: process agents and machine agents.

In situations where outcomes are influenced by both random elements and decisions made by individuals, the interaction process can be effectively modeled using a Markov Decision Process (MDP). Following this strategy, the MAS system is therefore modeled as a factored m-agent Decentralized Markov decision Process (DEC-MDP) (Fig. 5) consisting of a tuple of $(A_g, S, U, T, R, \delta, O)$. The set of agents, denoted asA_g, includes both process agents A_p and machine agents A_m. Each agent is associated with a specific resource within the FFL system. The state space , S, is factored into S_m and S_p, representing the states of the machine agents and process agents, respectively. The action space U is two-dimensional, with U_m representing the sub-action set of the machine agents and U_p representing the sub-action set of the process agents. Each agent has its own observation o $\in O$ based on the observation function $\delta(agent, u)$: $A_g \times U \rightarrow O$. The transition function $T(s, u, s')$ captures the probability of the system transitioning from state s to s' after executing a particular action u. It describes the dynamics of the system as it evolves over time. It is important to note that the reward function R assigns different rewards to each agent, addressing the issue of various agents having distinct objectives or goals.

4.1.2. First learning phase: assignment problem

During the learning phase, the focus is on solving the assignment

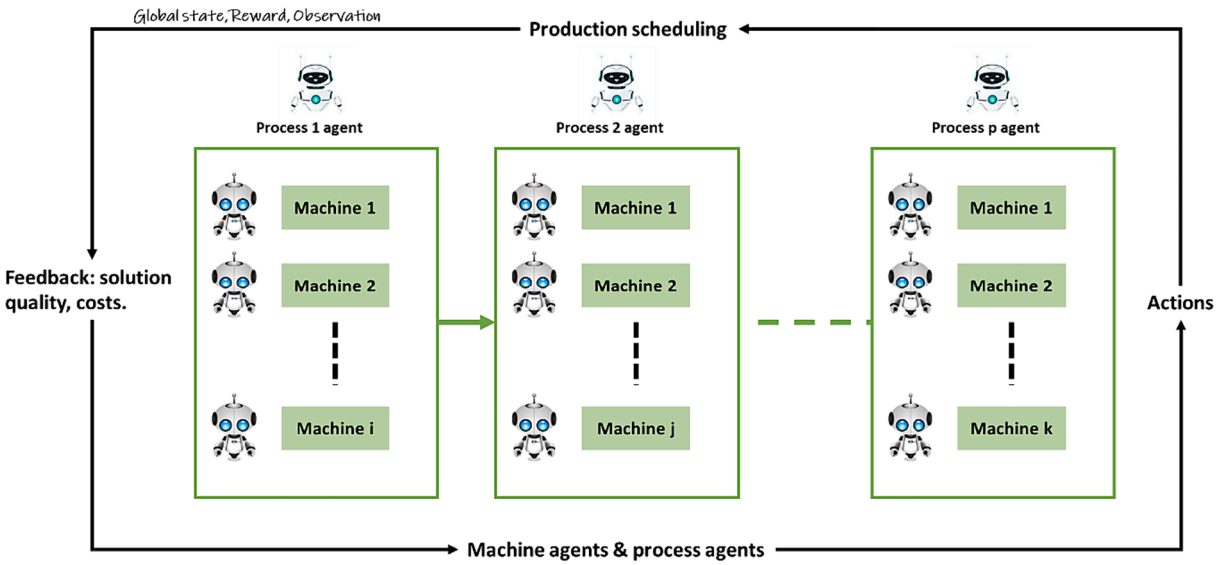

Fig. 5. Overview of the flexible flow line framework.

problem by considering various factors such as processing time, machine availability, and energy consumption. Each agent is trained to determine the most suitable processing machine for each product, as depicted in Fig. 6. The objective is to achieve a near-optimal allocation of products to parallel machines. To facilitate this optimization, each process p is associated with a process agent k. The process agent is responsible for boosting the reward by utilizing information about the energy consumption of the machines. The agent can ensure the appropriate allocation of products to machines, taking into account the overall energy efficiency of the system.

State representation: The state representation for each process is denoted by the local information about the current scheduling environment and the remaining products waiting to be processed. Each process agent has a local view of the available machines and the products waiting to be processed.

Action representation: The action space of a process agent in the FFL context is defined as the set of possible actions that the agent can take. This set typically includes selecting which machine to use from the available machines, considering their capacities. The primary objective of the process agent is to optimize the production process by achieving maximum efficiency while fulfilling quality standards and meeting customer demands.

Reward representation: the primary goal is to reduce the overall expenses associated with energy consumption and machine setups. Consequently, the reward system is designed to reflect this objective by assigning a lower value to actions that result in lower costs. More specifically, during this learning phase, the agent's task at each micro-period is to identify, for a product, the machine with the lowest corresponding energy consumption rate. Consequently, the process agent's reward consists in the power required to produce the given item on a machine. To ensure that demands are satisfied without any backorders at each macro-period, the reward is designed to penalize situations where the total processing time exceeds the duration of the macro-period ($|F|$). In such cases, a large reward value is assigned to discourage actions that lead to excessive processing time. The reward function is formulated as follows:

$$R_p(m,i) = \begin{cases} E^{on}_{i,m,p}.x^i_{m,p,t_f}, & f + \tau^i_{m,p}.x^i_{m,p,t_f} \leq |F| \\ big\ number\ M, & f + \tau^i_{m,p}.x^i_{m,p,t_f} > |F| \end{cases}$$

4.1.3. Second learning phase: sequencing problem

This learning phase addresses the sequencing problem, which aims to minimize the setup costs. Each machine within the process is associated with a corresponding machine agent. These machine agents are responsible for optimizing the sequencing of products to minimize costs (Fig. 7).

State and action representation: The state and action spaces for these machine agents are defined based on available products and the corresponding actions that can be taken in order to select from among the remaining products, respectively. In other words, the state space S_m is defined as the set of remaining products, while the action space U_m corresponds to the selection of a product.

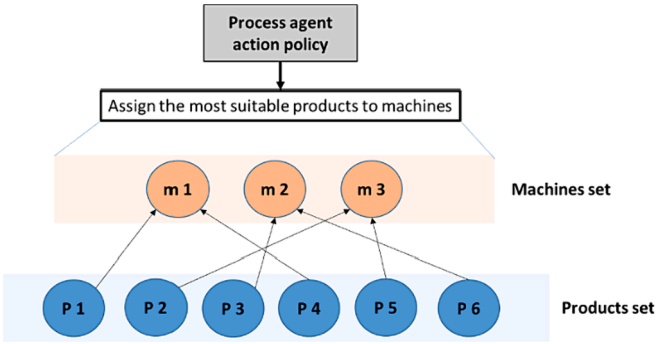

Fig. 6. Illustration of the action space of the process agent.

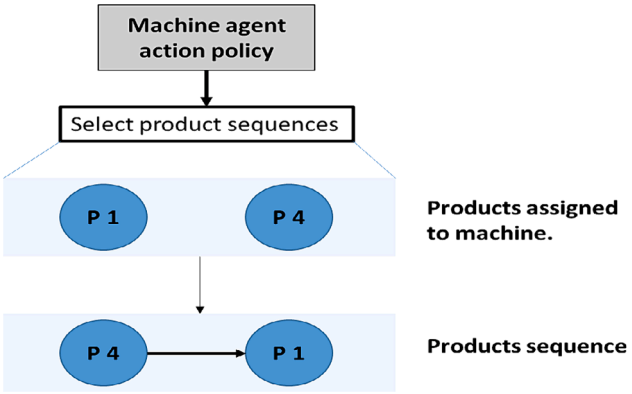

Fig. 7. Illustration of the action space of the machine agent.

Reward representation: The reward function for the machine agents is designed to minimize energy and setup costs, reflecting the objective of minimizing these costs during the sequencing process. The reward function can be expressed as follows:

$$R_m(i, remaining\ products) = \begin{cases} S^{j,i}_{m,p}.E^{off}_{m,p}.g_t + C^{j,i}_{m,p}, & f + \tau^i_{m,p}.quantity \le |F| \\ big\ number\ M, & f + \tau^i_{m,p}.quantity > |F| \end{cases}$$

4.1.4. Action selection strategy

In the reinforcement learning process, the selection of the action strategy plays a critical role. When the agent reaches a certain state S, it must make a decision between exploiting its previous experiences by selecting the best action based on the associated Q-value or exploring new paths by selecting a non-explored action. However, choosing the latter option prevents the agent from utilizing its prior knowledge. On the other hand, selecting the best action may lead to a known or sub-optimal path, hindering the agent's ability to minimize the cumulative reward and establish an efficient learning process. Therefore, it is necessary to strike a balance between exploration and exploitation to achieve the best results. A ε-greedy strategy is adopted here as the action selection method. Indeed, such a strategy has already been successfully applied in multi-agent scenarios (Gomes and Kowalczyk, 2009). The ε-greedy policy implies that the agent mainly exploits its existing knowledge by selecting the best action most of the time. However, occasionally, the agent selects a random action to explore new possibilities. The probability of selecting a random action is determined by the value of ε. In practice, to increase the learning rate and make use of prior knowledge effectively, less exploration is required for smaller scheduling problems. In contrast, larger problems necessitates a wider exploration process to discover better solutions and avoid being stuck in suboptimal paths.

4.1.5. MAS architecture

The MAS architecture presented in the context of the production scheduling problem in the FFL aims to minimize cumulative reward through agent-environment interactions. This architecture enables real-time scheduling and adaptive planning strategies based on incoming events, which can be classified into two types. The first type of events occurs at each micro-period f when a machine produces a certain quantity of a product and transfers it to the next process. Thereafter, process agents assess the product quantity and the available machines, and assign the products to a machine based on problem constraints. Products can then proceed for immediate production or wait temporarily, depending on the decision made by the machine agent. These events continuously update the system's information and status. The second type of event occurs when the production of a product is completed, resulting in a change in the agent and system status. This change affects the set of actions that the agent can choose from, requiring the agent to select another action from among the remaining actions. Overall, these outcomes lead to an exchange of information between agents, fostering interaction areas between them.

The proposed MAS architecture presented in Fig. 8 consists of two main parts: agent interaction and agent training. In the interaction part, agents operate with respect to the constraints of the framework and provide functionalities to establish the interaction network. This network enables communication and coordination between agents, facilitating their joint resolution of lot sizing and scheduling tasks. The training part of the architecture governs the dynamic behavior of the agents. Agents aim to minimize the cumulative reward by interacting with the environment, learning from their experiences and adjusting their behaviors over time to enhance their performance. Overall, this proposed MAS architecture shows promise for tackling complex manufacturing tasks that require coordination and cooperation among multiple agents.

4.1.6. Agent interactions

As seen previously, the proposed MAS architecture relies on, two types of agent interactions: process agents interacting with their associated machines, and interactions between consecutive processes. The first type of interaction occurs between process and the machines they are associated with. At each micro-period, process agents gather information from their associated machines to explore the environment. Then, this information is used as input to the policy model of the process agent to determine the action to take in order to minimize long-term cumulative rewards. Similarly, machine agents also communicate with process agents to gather information about the current state of the environment. The observations made by the agents are represented by

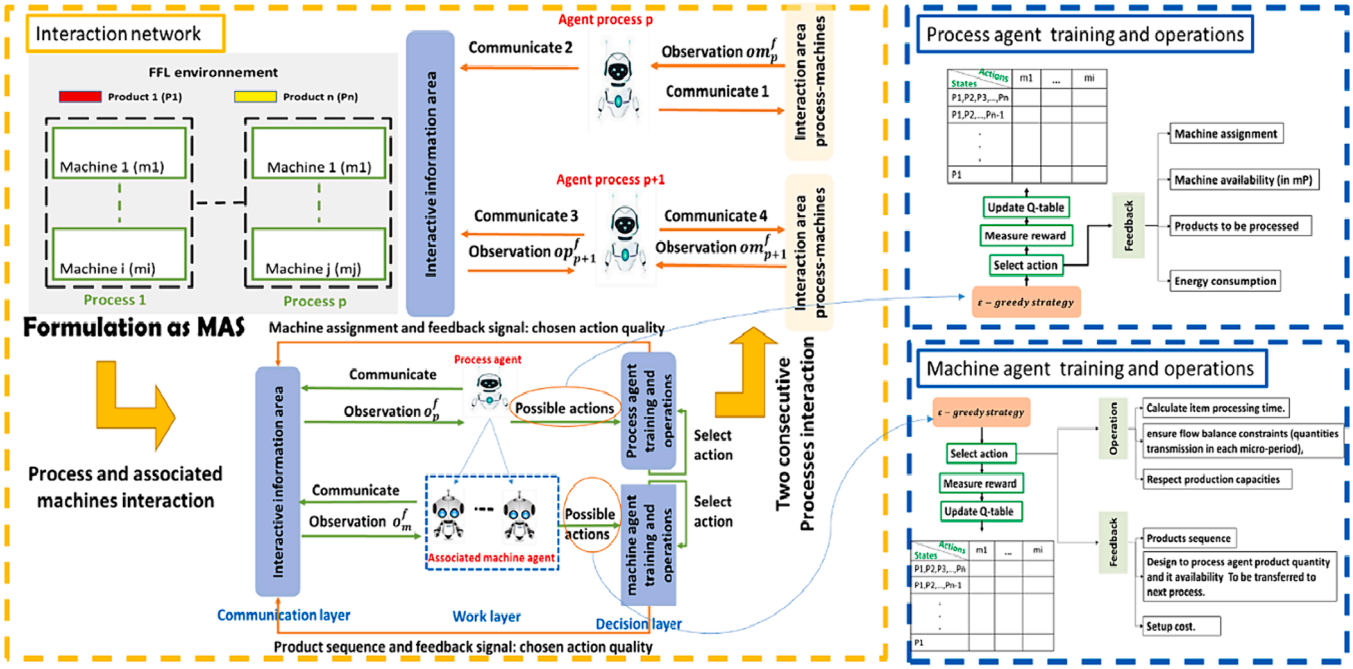

Fig. 8. MAS architecture.

o_m^f for machine agents and o_p^f for process agents. For machine agents, the observation includes information about the current product, the current micro-period f, and any affected products. For process agents, the observation includes information about available machines, the current micro-period f, and remaining products for production. The second type of interaction involves interactions between consecutive processes. At each micro-period, the process agent communicates with its associated machines (communication 1) to check if any quantities were produced during the previous micro-period and are ready to be transferred to the next process at the current micro-period (om_p^f). The process agent then passes this information to the next process. In turn, the agent linked to the latter communicates with its associated machines to make decisions. The observation for this type of interaction is represented by om_p^f, which includes information about the available products to be transferred to the next process, quantities, and the current micro-period.

4.1.7. Agent training

The agents' policy model in the proposed MAS architecture is based on Q-learning, which is a popular reinforcement-learning algorithm. Q-learning learns by building an action-value function that estimates the expected cumulative reward for taking a particular action in a particular state. The Q-value of a state-action pair (s,u) is updated based on the reward received after selecting an action u in state s, using the update rule given by equation (4a).

$$Q(s,u) = (1-\alpha)Q(s,u) + \alpha[r + \gamma^* \min_{u'}(Q(s',u') - Q(s,u))] \quad (4a)$$

Here, α corresponds the learning rate that determines the weight given to the new information obtained by the agent, while γ is the discount factor that determines the importance given to future rewards. The term r represents the immediate reward received by the agent after taking the action u in state s. The term $min_{u'}$ (Q (s',u') – Q (s,u)) represents the estimate of the maximum future reward that can be obtained after transitioning to the next state s' and taking the optimal action u' based on the current Q-values. The objective in this study is to minimize production costs, and the Q-learning algorithm is designed to find the optimal policy that achieves this objective by minimizing the expected cumulative reward over time. The input of the models consists of agent observation and action. Agent interaction and all the proposed Q-learning algorithms are summarized in

Algorithm 1. Process agent QL

Initialize:
$\boldsymbol{Q}(s,u)$=*zeros_matrix*($\boldsymbol{products_{number}}$, $\boldsymbol{machines_{number}}$)
For each episode do:
Initialize:
$\boldsymbol{S}$ = {list of remaining products to be processed}
Possible_actions = {available machines list}
for Micro-period = $\boldsymbol{f}$, $\boldsymbol{s}$ = product:
Choose $\boldsymbol{u}$ from possible_actions using ***ε-greedy*** policy.
Take action $\boldsymbol{u}$, calculate $\boldsymbol{R_p}(u,s)$,$\boldsymbol{S'} = s\backslash\{s\}$Update Q(s,u)
S=S'
Return selected machine and process observations.
End for
End for

algorithms 1 and 2.

Algorithm 2. Machine agent QL

Initialize:
$Q(s,u) = zeros\_matrix(2^{products_{number}}, products_{number})$
For each episode do:
S = {list of remaining products to be processed}
Possible_actions = {products list}
While machine is available, micro-period = f and $s \in S$:
Choose u from possible_actions using ε *–greedy* policy.
Take action u, calculate $R_m(u,s)$,$S' = S\backslash\{s\}$Update Q(s,u)
$S = S'$ and Possible_actions = possible actions \ {u}

(*continued on next column*)

(*continued*)

Algorithm 2. Machine agent QL

Availability = not_available(u)
Return machine observation
End while
End for

4.2. Second level optimization SLO: Trade-off between buffer use, RES, ESS and TOU prices

In the context of production planning, the SLO focuses on managing the tradeoff between on-peak electricity demand at all macro-periods and the utilization of RES, ESS and buffers. The primary goal is to effectively manage energy consumption from different sources and inventory usage during the planning process. One feasible approach to achieve this goal is to produce a surplus of the required quantity during off-peak macro-periods and store it as reserve stock. This reserve stock can then be utilized to meet the demand during on-peak macro-periods, enabling machines to reduce their working time and energy consumption, resulting in significant energy savings. Since the on-peak demand is defined on a macro-period scale, the SLO problem aims to determine the quantities to be produced and the quantities to be stored at each macro-period. The decision-making process in this problem operates at the macro-period level, taking into account the overall demand and supply requirements. However, the decision-making process in this problem operates at the macro-period level, potentially overlooking the challenge of product sequencing. Nevertheless, the allocation of machines is crucial to ensure smooth operations and minimize energy costs. To address this challenge, a linear optimization model is formulated within the SLO. The primary idea is to separate binary variables from the continuous variables in the objective function and relax production sequence constraints. The binary variables related to machine allocation are assumed to be known. Since the SLO is a linear program, it can determine the best trade-off between energy consumption during peak macro-periods, RES, ESS, and stock use. The obtained SLO problem is presented below.

Relaxed MP1

$$min\left(\sum_{f\in F}\sum_{t\in T}\left(g_t.ec_{t_f} + r_{t_f}.er_{t_f} + C^+.r^+_{t_f} + C^-.r^-_{t_f} + \sum_{p\in P}\sum_{i\in N} C^{buffer}_{i,p,t} Q^i_{p,t_f}\right)\right) \quad (5a)$$

S.T.
(2a) - (2f) (5b)

$$\sum_{i\in N} \tau^i_{m,p}.x^i_{m,p,t_f} \le d - \max_{i,j}\left(s^{i,j}_{m,p}\right), \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \quad (5c)$$

(2 h) (5d)

(3a) – (3j) (5e)

The SLO does not take into account constraints (2i)-(2n) due to its focus on managing the trade-off between on-peak electricity demand macro-periods and RES, ESS, and buffer use. Although capacity constraint (2i) is strongly linked to both binary and continuous variables, a reformulation trick presented in (5c) can separate binary variables while still considering setup times. This involves ensuring that the total production time at each micro-period is less than or equal to the duration of the micro-period minus the maximum setup time on a machine. The transmission of products between processes (constraints (2a)-(2e)) is programmed with machine allocation, which is assumed to be known from the FLO.

4.3. Summary of resolution steps

To summarize, the proposed algorithm is a two-level optimization approach. At the first level, the integrated lot sizing and production scheduling problem is solved using MAS. The output of the FLO provides the best production schedules and machine affectation, which serves as input for the second level. The SLO problem is focused on managing the tradeoff between on-peak electricity demand macro-periods and RES, ESS, and buffer use. The objective is to effectively balance energy consumption and inventory usage during the planning process. To address this challenge, a deterministic polynomial planning problem is formulated. The obtained SLO problem is solved by a linear program, which seeks the best trade-off between energy consumption during peak macro-periods, RES, ESS, and stock use. Once the second level problem is solved, the best solutions for lot sizes and buffer inventory levels at the macro-period scale are used to update the initial FLO problem. The updated FLO problem is then solved to obtain new lot sizes and production schedules at the micro-period scale.

5. Computational results

The proposed model was implemented from scratch using Python 3.9 and executed on a personal computer equipped with a Core™ i7-11850H 2.5 GHz CPU and 32 GB of RAM. The computation times were measured in CPU seconds. For numerical studies, a benchmark experiment was conducted to assess the performance of the proposed framework. Subsequently, computational tests were performed on small, medium, and large problems to evaluate the performance of the RL algorithm.

5.1. Input data

For the benchmark experiment, the production parameters were adjusted based on Özdamar and Barbarasoglu's work (1999). They were randomly generated according to the following specifications: The scheduling horizon is a day-ahead horizon divided into $|T| = 4$ macro-periods, each further divided into $|F| = 6$ micro-periods. Each micro-period corresponds to an aggregate of one hour. The FFL production system comprises $|P| = 3$ sequential processes. The two first processes have $|M_1| = |M_2| = 1$ machine each, while the third has $|M_3| = 3$ non-identical parallel machines. The number of final products is set to $|N| = 3$ products. The final demand for products exhibits high variability and is generated from a uniform distribution $U(20, 70)$. The processing times $\tau^i_{m,p}$ (in minutes) are generated from $U(1, 4)$. Setup times, expressed in minutes, are proportional to the total processing time and are calculated using the formula: $s^{i,j}_{m,p} = \frac{S\sum_{j,t,m}\tau^j_{m,p}*D_{i,t}}{|T|*max_p(|M_P|)}$, where S is generated from $U(0.05, 0.01)$. The setup costs are proportional to the setup times, meaning that the setup costs will increase as the associated times increase. The inventory cost $C^{buffer}_{i,p,t}$ is constant across all processes and products and is set at 2 € per unit. The capacity of buffers N_p is set to 100 units. Tables 4, 5, and 6 depict the generated values for demands, processing times and setup parameters, respectively. Tables 7, 8 and 9 present machines energy consumption, PV energy availability and TOU prices, respectively.

The energy parameters are based on the Duarte et al. (2020) study. The availability of photovoltaic (PV) energy is obtained from a PV power plant located on the roof of the Polytech University of Nantes in France. Daily production of the PV panels corresponds to data collected in April 2019. The on-peak macro-period consists of macro-period $t = 3$. ESS capacity parameters are set as follows: $s_{min} = 0.4MWh$, $s_{max} = 1.5MWh$ and $s_0 = 0.6MWh$. The charging and discharging power limits are both set to $R^+ = R^- = 0.5\,MWh$. The ESS efficiency are set as follows: $\eta^+ = \eta^- = 1, c^+ = c^- = 25$ €. The cost of MWh Solar power is estimated from solar panels investment, panel life span and efficiency and it set as $r_{t_f} =$ 50€/MWh. $E^{off}_{i,m,p}$ is set at 0.005 $MWh/unit$. The learning parameters associated to agents are set as follows: α = 0.2, γ = 0.5 and ε = 0.2.

Table 4
Product demand.

time \ products	$i = 1$	$i = 2$	$i = 3$
$t = 1, f = 6$	40	41	33
$t = 2, f = 6$	36	41	52
$t = 3, f = 6$	53	42	54
$t = 4, f = 6$	29	65	30

5.2. Benchmark experiment

5.2.1. 1st step: Solve the first level optimization: FLO

The FLO step aims to identify the best sequence and machine allocation using a reinforcement learning (RL) approach. Fig. 9 and Fig. 10 illustrate the training process of the process agents and machine agents, respectively. It is apparent that the cumulative rewards of the two agents

Table 5
Processing times (min).

Machine \ products	$i = 1$	$i = 2$	$i = 3$
$p = 1, m = 1$	2	1.3	1.35
$p = 2, m = 1$	1.8	1.3	1.3
$p = 3, m = 1$	2.5	1.8	2
$p = 3, m = 2$	1.7	2	1.2
$p = 3, m = 3$	2.2	1.3	2.2

Table 6
Setup times and costs for two machines demand.

Machine		Setup time (min)				Cost (€)		
$p = 1, m = 1$		$i = 1$	$i = 2$	$i = 3$		$i = 1$	$i = 2$	$i = 3$
	$j = 1$	0	7	3	$j = 1$	0	80	44
	$j = 2$	8	0	6	$j = 2$	86	0	62
	$j = 3$	6	9	0	$j = 3$	60	83	0
Machine		Setup time (min)				Cost (€)		
$p = 2, m = 1$		$i = 1$	$i = 2$	$i = 3$		$i = 1$	$i = 2$	$i = 3$
	$j = 1$	0	9	5	$j = 1$	0	99	58
	$j = 2$	9	0	3	$j = 2$	93	0	32
	$j = 3$	9	11	0	$j = 3$	102	108	0

Table 7
Machines power consumption $E^{on}_{i,m,p}$ ($MW/unit$).

Machine \ products	$i = 1$	$i = 2$	$i = 3$
$p = 1, m = 1$	0.02	0.015	0.014
$p = 2, m = 1$	0.019	0.013	0.014
$p = 3, m = 1$	0.026	0.021	0.02
$p = 3, m = 2$	0.017	0.019	0.017
$p = 3, m = 3$	0.025	0.015	0.015

Table 8
PV power availability (MW).

f	1	2	3	4	5	6
G_{1_f}(MW)	0	0	0	0	0	0
G_{2_f}(MW)	0.1	0.4	0.56	0.634	0.82	0.73
G_{3_f}(MW)	0.8 6	0.71	0.62	0.78	0.43	0.3
G_{4_f}(MW)	0.28	0.1	0	0	0	0

Table 9
TOU prices (€/MWh).

t	1	2	3	4
g_t(€/MWh)	70	70	130	70

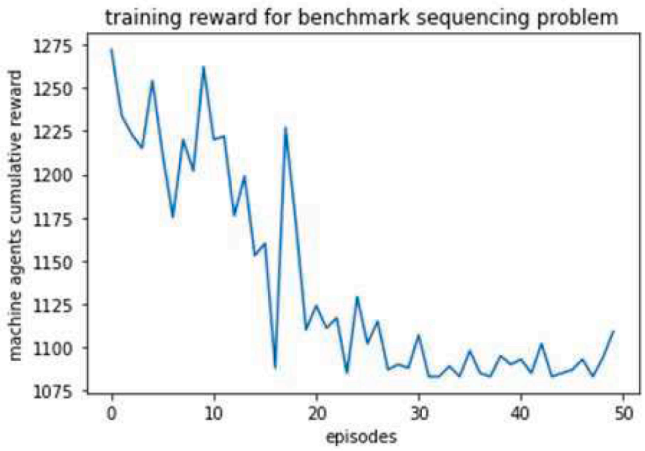

Fig. 9. Training reward for machine agents.

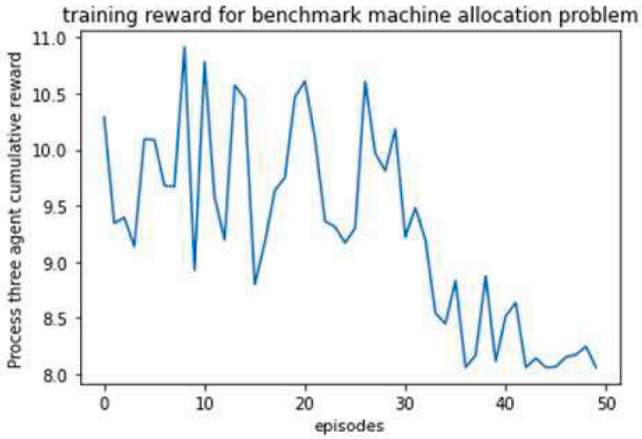

Fig. 10. Training reward for process agents.

display a downward trend, with occasional increases in value due to the exploitation-exploration strategy related to the QL algorithm. Fig. 11 presents agent's states and actions that depict the best sequence and machine allocation. It can be seen that only the first process agent is able to initiate an initial action at the first micro-period ($t = 1$ and $f = 1$) and all products are assigned to the machine, M1, of the first process. Subsequently, M1 chooses product $i = 2$ among the set of three products.

Upon completion of the processing of $i = 2$, the aforementioned sets are updated (Fig. 11. (c)). Next, other agents become available to act (i.e., transmit products) at the second micro-period ($t = 1$ and $f = 2$) (Fig. 11. (b)). Therefore, the best product sequence selected by the MAS of process 1 and 2 at the first macro-period is $i = 2/i = 3/i = 1$. This sequence ensures no delays and leads to a lowest setup cost = 256€. After that, the agent of the third process assigns one product to each machine with the goal of minimizing the setup cost. These machine changeovers have been maintained throughout the entire scheduling horizon to avoid further setup costs.

5.2.2. 2nd step: solve the second level optimization

The resolution of the relaxed MP1 aims to ensure the best trade-off between energy consumption during the on-peak macro-period, inventory, RES and ESS utilization. Fig. 12 illustrates the best production scheduling of the first product $i = 1$. It can be observed that the production of $i = 1$ at the first macro-period exceeds the demands in process 1 and 2. Consequently, the excess is stored in the intermediate inventory to be further used in the on-peak macro-period. The energy consumption planning derived from the obtained scheduling scheme is illustrated in Fig. 13. It is obvious that the integration of both onsite RES and an ESS, as well as inventory policies, reduces the grid energy consumption during on-peak macro-period ($t = 3$). As a result, the total inventory cost is set at 461 €, the grid energy cost amounts to 898 €, while the solar power and ESS cost is set at 365 €.

5.2.3. 3rd step: resolve the FLO

The final step involves solving the FLO problem once again in order

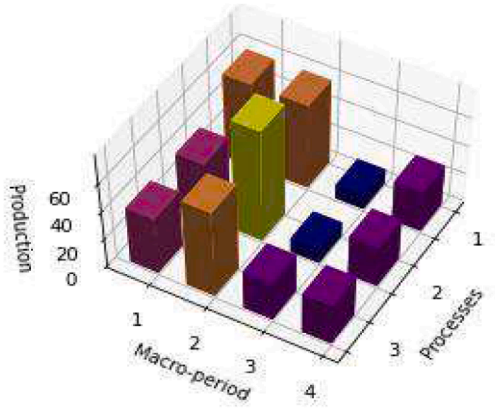

Fig. 12. Best production schedule for item 1.

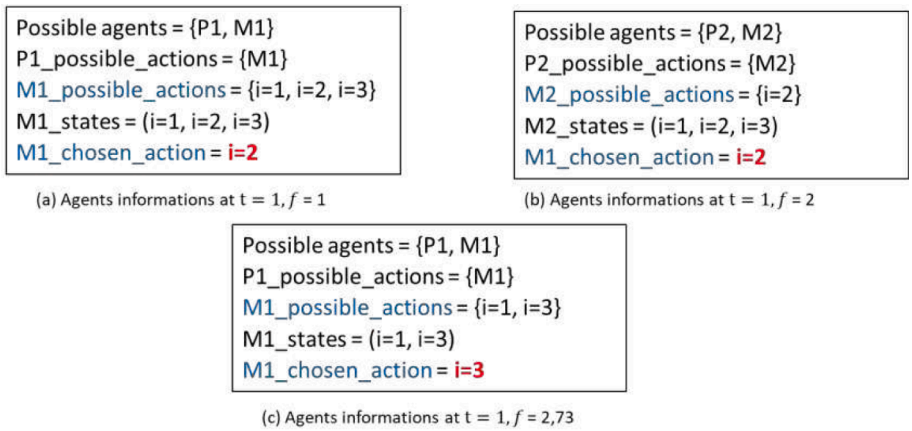

Fig. 11. System sets at different time steps.

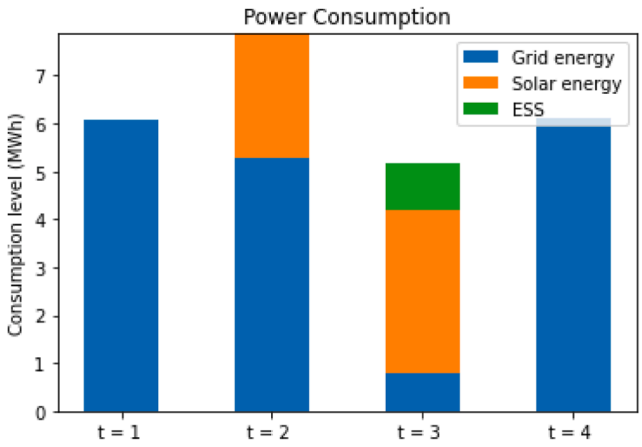

Fig. 13. Total energy consumption at each macro-period.

to explore scheduling and lot sizing at the micro-period scale. The Gantt chart of the best-obtained scheduling scheme is depicted in Fig. 14. The color-coded boxes provide a visual representation of the produced and buffered quantities across different products through the scheduling horizon. The green boxes correspond to product 1 ($i = 1$). The orange boxes show the quantities produced and buffered of product 2 ($i = 2$), while the blue boxes correspond to product 3 ($i = 3$). It can be seen that the schedule is reasonable in the sense that there is no overlap between different production steps of the same product, and the next production step is started only when the previous production step is completed. The best sequence that leads to a minimal setup cost of 1078 € is presented in Table 10. The proposed approach successfully solves the problem in three primary stages with a CPU time of 3.28 s, resulting in a total cost of 2 802 €.

5.2.4. Sensitivity analysis on TOU macro-periods

5.2.4.1. Modification of on-peak macro-periods. In this experiment, the on-peak-macro-period is switched from the third to the second macro-period ($t = 2$). The other parameters are maintained. Results are shown in Figs. 15-16. As can be seen in Fig. 15, following the same concept, an extra quantity of $i = 1$ is manufactured and stored in the first macro-period to be used in the on-peak macro-period. The energy management scheme shown in Fig. 16 is changed. It can be seen that the ESS is used in the second macro-period. The ESS is discharged in $t = 2$, $f = 1$ and charged in the third macro-period. The manufacturing system relies mainly on alternative sources of energy rather than on conventional energy. Regarding the scheduling problem, the machine assigning

Table 10
Best sequence.

Macro-period	Sequence	Cost
$t = 1$	$i = 2/i = 3/i = 1$	256 €
$t = 2$	$i = 1/i = 3/i = 2$	293 €
$t = 3$	$i = 2/i = 3/i = 1$	256 €
$t = 4$	$i = 1/i = 2/i = 3$	273 €

and production sequence are the same, with respect to the benchmark case. Finally, the total cost, is set to 2 919 € as follows; 1 538 € operational cost, 1 104 € grid energy cost and 276 € solar energy and ESS cost.

5.2.4.2. Modification of time granularity. In this case, a scheduling horizon of thirty-six hours ahead is simulated, where $|T| = 6$ macro-periods. Each macro-period consists of six micro-periods of one hour. Parameters that depend on time such as demands, solar energy availability, and TOU energy prices are adjusted considering the new planning horizon. The on-peak macro-periods are set to macro-periods $t \in \{3, 5, 6\}$ and the off-peak macro-periods are set to $t \in \{1,2,4\}$. In this case, the new time-dependent parameters are shown in Tables 11 and 12. All the other parameters used in the benchmark were kept constant.

Results are shown in Figs. 17-18. As presented in Fig. 17, conventional grid energy is less used during $t \in \{3, 5, 6\}$ compared to other macro-periods. In these on-peak macro-periods, manufacturing facility consumes energy from alternative sources like ESS and PV panels. It can be seen in Fig. 18 that the ESS is discharged in on-peak macro-periods. Finally, the total cost is set to 4482 € as follows; conventional energy cost = 1891 €, solar energy = 389 €, ESS cost = 97 € and operational cost = 2105 €.

5.2.5. Cost comparison under different scenarios

The proposed model has been tested under different scenarios to demonstrate the economic performance of our research. We used the proposed approach for the resolution process. In the baseline model, we supposed that there is no RES, no ESS and no inventories ($G_{t_f} = 0MW, R^+ = 0MWh$ and $N_p = 0$). The second scenario carried out the proposed model without considering the RES and the ESS. In the third scenario, the model was tested with the assumption that there were no inventories ($N_p = 0$). As shown in Table 13, it is evident that our model is the most cost-effective, with a total cost equal to 2802 €. Additionally, it can be observed that the conventional electricity cost obtained by our model is approximately 59 % less than the case without RES and ESS. Hence, these results highlight the fact that the integration of RES and ESS can significantly reduce electricity consumption costs without compromising production throughput. Results obtained from the same scenario without an inventory policy emphasize the importance of buffers in reducing energy consumption during on-peak macro-

		Macro-period 1 (t=1)						Macro-period 2 (t=2)						Macro-period 3 (t=3)						Macro-period 4 (t=4)					
		f=1	f=2	f=3	f=4	f=5	f=6	f=1	f=2	f=3	f=4	f=5	f=6	f=1	f=2	f=3	f=4	f=5	f=6	f=1	f=2	f=3	f=4	f=5	f=6
Production	p=1,m=1	41	36	33	26		0	30	23	52	41	0	0	42	47	25	0	0	0	20	65	30	0	0	0
	p=2,m=1	0	41	36	33	26	0	0	30	23	52	41	0	0	42	47	16	0	0	0	29	65	30	0	0
	p=3,m=1	0	0	0	33	0	0	0	0	0	7	45	0	0	0	0	30	19	0	0	0	0	0	30	0
	p=3,m=1	0	0	0	3	30	7	0	0	30	31	0	0	0	0	0	0	27	0	0	0	29	0	0	0
	p=3,m=1	0	0	41	0	0	0	0	0	0	0	0	41	0	0	42	0	0	0	0	0	0	65		0
Inventories	p=1,m=1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	9	9	9	9	0	0	0	0	0
	p=2,m=1	0	0	0	0	3	19 3	19 3	19 3	19 3	11 3	11	11	11	11	11	11	0	0	0	0	0	0	0	0
	p=3,m=1	0	0	0	0	33	0	0	0	0	0	7	0	0	0	0	0	30	0	0	0	0	0	0	0
	p=3,m=1	0	0	0	0	0	0	0	0	0	25	25	25	25	25	25	25	53	0	0	0	0	29	29	0
	p=3,m=1	0	0	0	41	41	0	0	0	0	0	0	0	0	0	0	42	42	0	0	0	0	0	45	0

Fig. 14. Best production scheme.

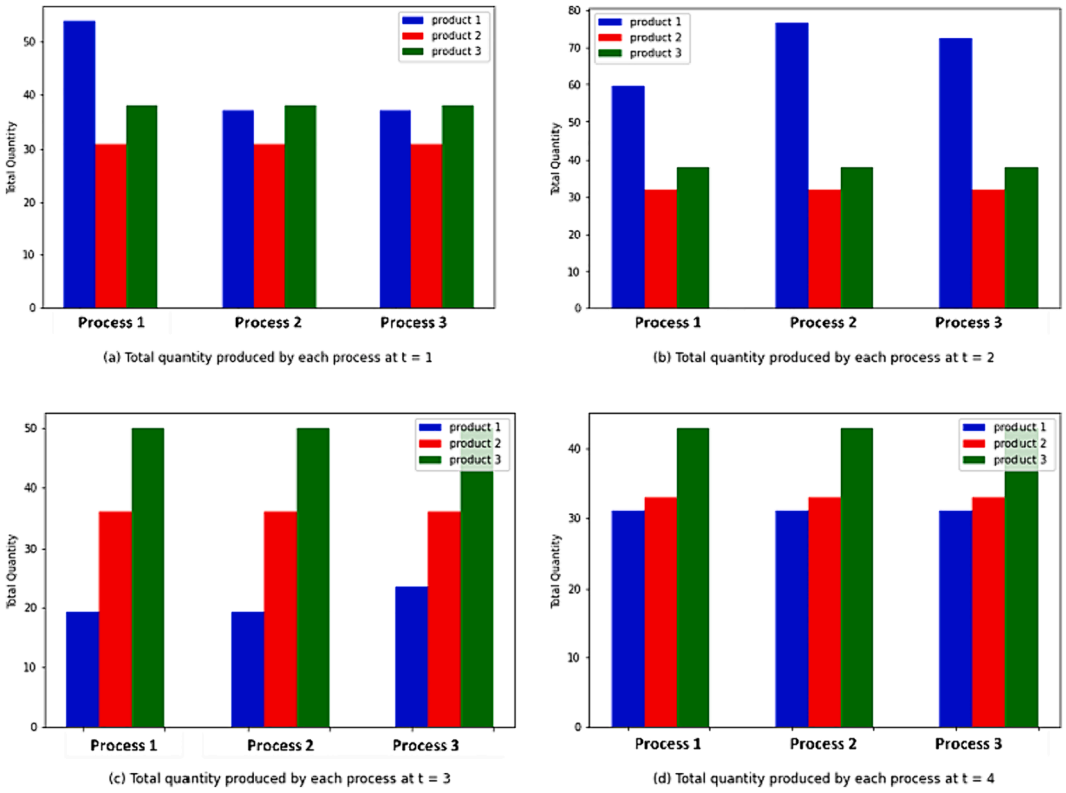

Fig. 15. Best production schedule for modified peak macro-period.

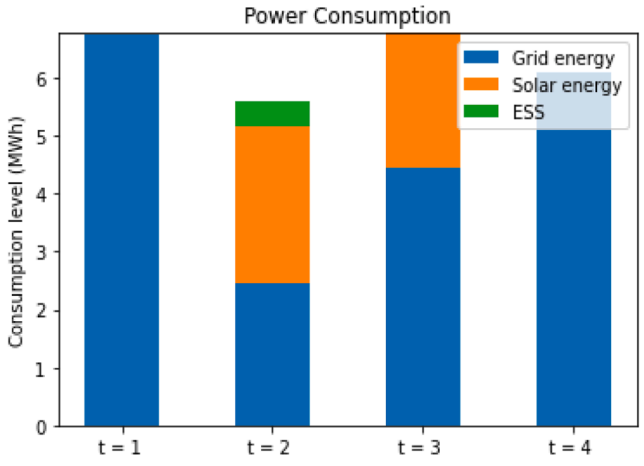

Fig. 16. Energy consumption for modified peak macro-period.

Table 11
Demands for the new scheduling horizon.

time \ products	$i = 1$	$i = 2$	$i = 3$
$t = 1, f = 6$	40	41	33
$t = 2, f = 6$	36	41	52
$t = 3, f = 6$	53	42	54
$t = 4, f = 6$	29	65	30
$t = 5, f = 6$	45	31	27
$t = 6, f = 6$	25	50	31

periods. However, it should be noted that inventories alone do not necessarily lead to a reduction in energy consumption during on-peak macro-periods. Interestingly, their integration with other components such as RES and ESS leads to more optimal power consumption and cost savings.

Table 12
Energy availability for the new scheduling horizon.

f	1	2	3	4	5	6
G_{1_f}(MW)	0	0	0	0	0	0
G_{2_f}(MW)	0	0	0	0.2	0.37	0.67
G_{3_f}(MW)	0.76	0.61	0.52	0.68	0.33	0.2
G_{4_f}(MW)	0.18	0.1	0	0	0	0
G_5(MW)	0	0.2	0.4	0.434	0.72	0.43
G_{6_f}(MW)	0.84	0.45	0.77	0.51	0.27	0.19

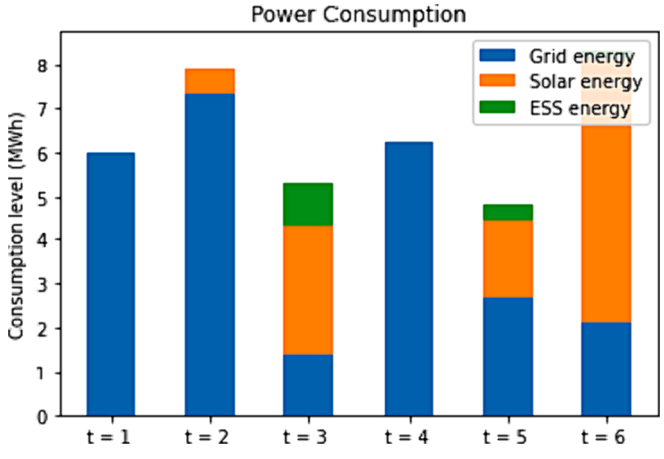

Fig. 17. Energy consumption for the new scheduling horizon.

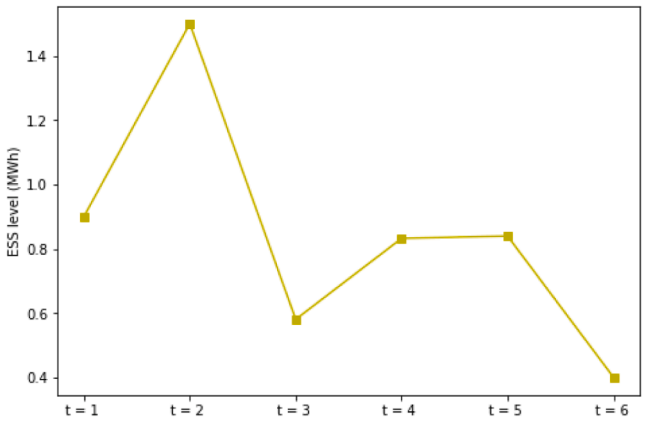

Fig. 18. Energy consumption for the new scheduling horizon.

Table 13
Comparison of costs among different scenarios.

Model	Conventional electricity charge (€)	Inventory charge (€)	RES & ESS charge(€)	Total cost (€)
1-Baseline	2198	0	0	3276
2-No RES & no ESS	1649	461	0	3188
3-No inventories	1580	0	365	3023
4-Proposed model	898	461	365	2802

5.3. Performance evaluation of the reinforcement learning approach

Based on Özdamar and Barbarasoglu (1999), we conducted computational experiments on three groups of instances. To this end, small, medium, and large-scale ILSPS instances were generated randomly to validate the effectiveness of our RL algorithm. The small-scale instances (S1 to S5) comprise four products, two processes, and two parallel machines per process. The medium instances (M1 to M5) consist of eight products, three processes, and three parallel machines within each process, while the large-scale instances (L1 to L5) consist of twelve products, four processes and four parallel machines per process. It is important to note that in this analysis, all instances were assumed to have a uniform planning horizon of $|T| = 1$ and $|F| = 6$. For each group of instances, demands and machine processing times exhibit variability from one instance to another. These values were randomly generated according to the methodology outlined before.

5.3.1. Training process analysis

The training process and cumulative rewards for process agents and machine agents are shown in Figs. 19, 20, 21 and 22, for one medium instance (MI) and one large instance (LI) respectively. These figures shed light on the evolution and performance of the MAS architecture during the training process. In the assignment problem, the cumulative reward curves start to converge around 100 episodes for the medium instance and around 400 episodes for the large instance. This convergence state indicates that the process agents have learned an optimal policy and consistently achieved high rewards. It also indicates that the process agents have reached a stable level of performance. In addition, the downward trend in energy consumption values in Figs. 19 and 21 demonstrates that process agents are able to reduce energy consumption throughout the training process. Nevertheless, some fluctuations in the energy consumption curve can be observed after the convergence of the model. Such a phenomenon may arise due to the exploitation-exploration trade-off of the process agent. However, the maximum energy levels reported in Figs. 19 and 21 are 67.08 MWh for the medium instance and 96.23 MWh for the large instance. These values represent the worst-case scenarios encountered by process agents, yet they still demonstrate improved energy consumption compared to initial or random policies. Additionally, it is worth mentioning that the best energy consumption achieved for the medium instance reaches 59.81

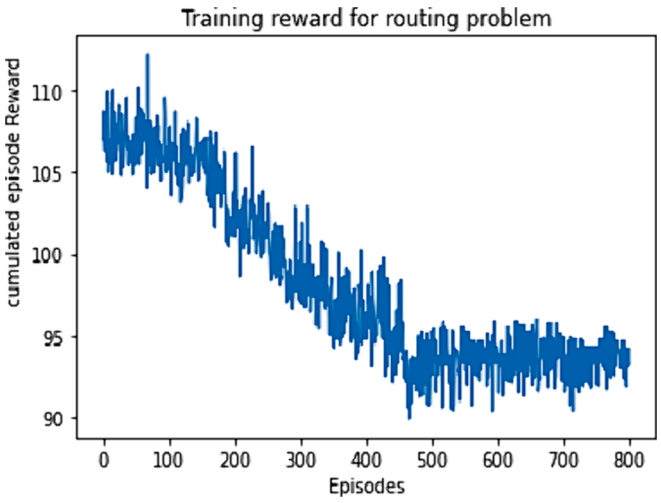

Fig. 19. MI Training reward for process agents.

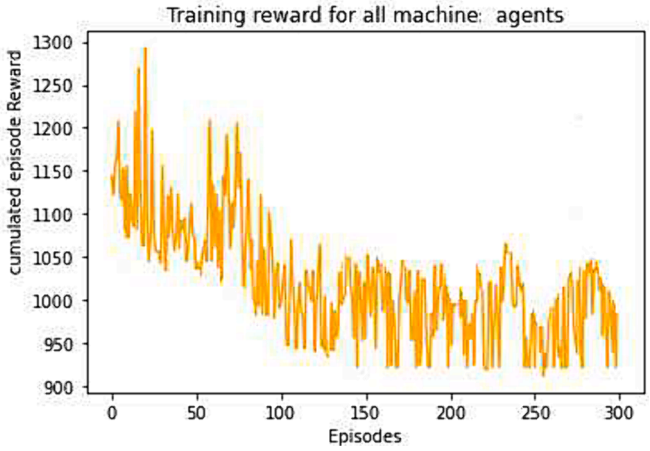

Fig. 20. MI Training reward for machine agents.

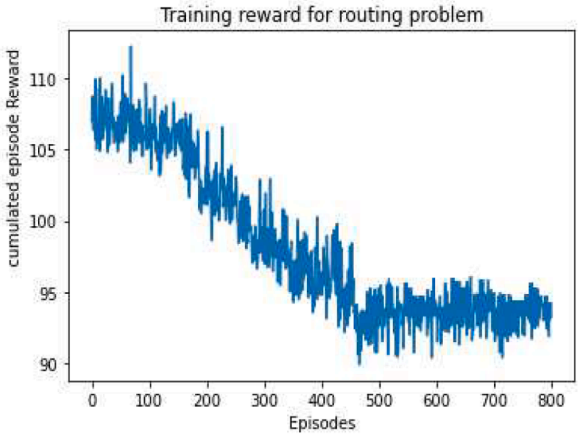

Fig. 21. LI Training reward for process agents.

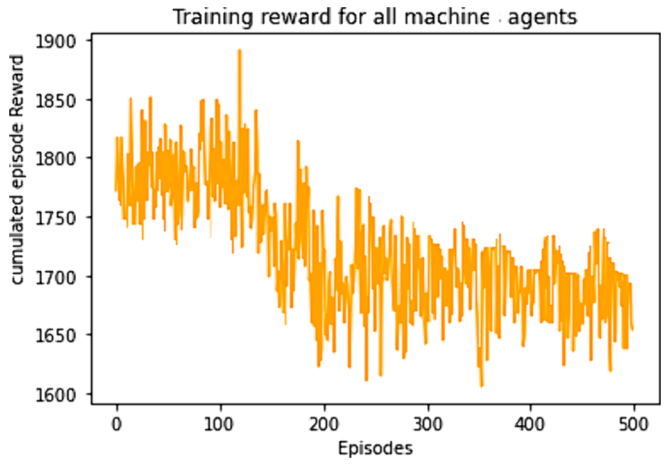

Fig. 22. LI Training reward for machine agents.

MWh after 300 episodes.

Regarding sequencing problems, each machine agent aims to determine the best product sequencing to minimize setup costs. The cumulative reward of the machine agents represents the setup costs. In Figs. 20 and 22, we can observe a downward trend in the setup costs, indicating that the machine agents have been successfully reducing these costs over time. The cumulative reward curves start to converge around 100 episodes for the MI and around 200 episodes for the LI. This trend reflects that agents have learned to improve their product sequencing strategy, leading to better setup processes and ultimately lower costs. The worst case scenarios found by machine agents within one episode are set at 1298 € and for the MI and 1896 € for the LI. On the other hand, the best scenarios are set at 922 € for the MI and 1607 € for the LI.

To conclude, MAS agents have been trained effectively, which exhibit a downward trend in energy consumption and setup cost as well as convergence of the cumulative reward curve. The agents have learned an optimal policy, resulting in stable performance and consistently improved scheduling decisions.

5.3.2. *Computational results of generated instances*

Further experiments were conducted using the instances presented in section 5.3. In this subsection, the rewards derived from the solutions obtained by our approach were compared with those generated by CPLEX for both assignment and sequencing problems. In the next simulation, CPLEX v.12.10 was used limited in 1000 s.

Table 14 presents the outcomes of these instances for the assignment problem. Notably, the cumulative reward achieved in the assignment problem matches the best energy consumption for all small instances, resulting in a relative error (RE) of 0 % for each instance. It is worth mentioning that smaller instances have fewer variables, constraints and possible actions, which makes it easier for the process agent to quickly explore different actions and learn the optimal policy. For medium instances, the RE varies from 0.61 % to 1.56 %, with an average RE of 1.01 %. These results indicate that the achieved energy consumption is relatively close to the best energy consumption obtained by CPLEX. However, for large instances, the RE varies from 1.5 % to 4.6 % with an average RE of 2.8 %. This implies a slightly higher deviation from energy consumption obtained by the automatic solver compared to smaller instances. The exponential expansion of the search space for potential solutions in larger instances makes it more challenging to explore and to determine good solutions.

Results, associated with the sequencing problem, are shown in Table 15. They largely follow the observations made for the assignment problem. The cumulative rewards obtained by the machine agents match the same solution obtained by CPLEX for all small instances,

Table 14
Process agent results on different instances.

Instances Problem Size	Instance reference	The best cumulative reward of the assignment problem given by the RL approach	The cumulative reward of the assignment problem calculated by CPLEX	cost RE
	S1	29	29	0.0 %
Small	S2	21.233	21.233	0.0 %
(4,2,2)	S3	22.043	22.043	0.0 %
	S4	21.047	21.047	0.0 %
	S5	20.205	20.205	0.0 %
Average		22.705	22.705	0.0 %
	M1	57.836	57.296	0.94 %
	M2	56.699	55.823	1.56 %
Medium	M3	58.494	57.869	1.08 %
(8,3,3)	M4	61.014	60.64	0.61 %
	M5	59.81	59.28	0.89 %
Average		58.7706	58.1	1.01 %
	L1	91.15	87.14	4.6 %
Large	L2	88.631	85.24	3.97 %
(12,4,4)	L3	93.31	91.6	1.86 %
	L4	94.49	92.36	2.3 %
	L5	90.41	89.05	1.5 %
Average		91.59	89.07	2.8 %

Table 15
Machine agent results on different instances.

Instances Problem Size	Instance reference	The best cumulative reward of the sequencing problem given by the RL approach	The cumulative reward of the assignment problem calculated by CPLEX	cost RE
	S1	329	329	0.0 %
Small	S2	347	347	0.0 %
(4,2,2)	S3	321	321	0.0 %
	S4	272	272	0.0 %
	S5	325	325	0.0 %
Average		318.8	318.8	0.0 %
	M1	1173	1173	0.0 %
Medium	M2	922	920	0.2 %
(9,3,3)	M3	1133	1125	0.7 %
	M4	1198	1185	1.1 %
	M5	1029	1024	0.4 %
Average		1091	1085.4	0.48 %
	L1	1802	1781	1.2 %
Large	L2	1607	1581	1.6 %
(12,4,4)	L3	1701	1666	2.1 %
	L4	1818	1777	2.3 %
	L5	1792	1753	2.2 %
Average		1744	1711.6	1.88 %

resulting in a relative error of 0 % for each instance. For medium instances, the relative errors vary from 0.2 % to 1.1 %, with an average equal to 0.48 %. This indicates that the achieved sequencing problem rewards are relatively close to the best solution, with only a small deviation. However, for large instances, the relative errors increase, varying from 1.2 % to 2.3 %, with an average value equal to 1.88 %. These larger instances exhibit a slightly higher deviation from the CPLEX solution compared to the smaller ones. Overall, the analysis of both the performance of the machine agents and the process agents indicates that the proposed method performs well, and the achieved rewards values being relatively close to the best solutions. The results align with the observations made previously regarding the impact of problem size, complexity, and search space.

5.3.3. *Comparison of results with FIFO and genetic algorithm (GA) approaches*

In this subsection, we kept the instances previously used (S1 to S5, M1 to M5 and L1 to L5) to conduct a comparative analysis of the RL approach against the heuristic rule FIFO and genetic algorithm (GA)

approaches. This evaluation encompasses considerations of costs, computation time (CPU) and performance measured by the obtained relative error. The heuristic FIFO, a widely used simple rule, prioritizes products based on their arrival order. In the first micro-period of the scheduling horizon, FIFO assigns products to the primary process machines. This assignment is predicated on the machines' minimal energy consumption and their current availability. Subsequently, for each designed machine, FIFO assigns the first product in line. This precise selection process ensures that the earliest arriving products are processed first. As the workflow progresses to the subsequent process, FIFO allocates the foremost available product to a designated machine. This process goes on for the remaining micro-periods and processes, ensuring that products are processed in the order they arrive and machine are allocated on the basis of their energy consumption. Applying FIFO in our context leads to feasible solutions with partially optimized energy consumption and non-optimized product sequencing. The GA is a popular metaheuristic technique known for its exploration capabilities. With regard to GA application, the method proposed by Valledor et al. (2018) was adapted to our setting. In GA, two populations are defined; the first one is responsible for machine assignment and the second one is responsible for sequencing problem. GA starts with two initial populations, performing selection, crossover and mutation between individuals and returns a feasible solution. The parameters of the GA are set as follows: iteration number = 200, population size = 30, mutation probability = 0.2 and crossover probability = 0.8.

Table 16 presents a comparison of costs obtained by the three methods for the same problem instances. CPLEX_1 cost in columns represents the cost obtained by CPLEX starting from scratch. CPLEX_2 reports the cost obtained by CPLEX starting from the initial solution generated by our proposed reinforcement learning technique. The objective of this strategy is to ascertain whether optimality can be achieved within a time constraint. If the execution time is below 1000 s, the corresponding instance is solved to optimality.

The results demonstrate that the RL approach consistently outperforms FIFO in terms of cost minimization across all problem sizes. In comparison to GA, the RL approach generally achieves better overall results, with smaller total costs on most instances. Only on two instances, the GA outperforms the RL approach in terms of total cost. Nevertheless, the RL approach achieves results close to the CPLEX_1 solutions, with negligible relative errors for small and medium instances. Even for large instances, the RL approach maintains its competitiveness, although with a slightly increased relative error. One-way analysis of variance (ANOVA) was conducted on each type of instances (Small, Medium, Large) using R software, version 4.3.1 (R Core Team, 2021). The cost mean plots together with LSD (Least Significant Difference) 95 % confidence intervals associated with the various approaches are reported in Fig. 23 according to the type of instances. As shown in the figure, the RL approach shows significant improvements over heuristic rule FIFO and performs competitively against GA. The RL approach consistently achieves lower costs and produces results close to solutions obtained by CPLEX_1, indicating its effectiveness in solving ILSPS problems. The learning nature of agents in the RL approach contributes to this performance.

Regarding computational time evaluation criteria, the average execution times corresponding to the different methods applied to the varying size problems are shown in Table 17. For small instances, both the CPLEX scenarios reach the optimum in all instances. For instances M3 and M5 from medium instances, using the initial solution of the proposed RL approach, CPLEX_2 can find a better solution than CPLEX. However, these solutions are not optimal since their execution times exceed 1000 s as shown in Table 17. For large instances, due to the increasing complexity of these instances, CPLEX_2 also fails to prove optimality within 1000 s.

Regarding other approaches, for all problem instances, the FIFO method takes at most 0.23 s to find a local solution. While FIFO may provide faster execution times, it is not able to lead to the best-quality solutions. On the other hand, GA involves a more sophisticated algorithm that aims to explore the search space and find better solutions. This exploration process typically requires more computational effort. The resulting execution time appears to be very long, to respond to the changes in the FFL environment. However, the RL approach did not exceed 8 s to make a decision while the automatic solver reached a low bound solution within 1000 s. The RL approach has the potential to explore the search space more comprehensively and find solutions that are closer to the CPLEX solver. It is able to find higher-quality solutions than the GA in less time.

Finally, it is important to note that the RL approach can adapt immediately to the environment changes and consequently deal with the uncertainty in the ILSPS problem. It offers a combination of efficiency, solution quality, and adaptability, making it a promising approach for optimizing ILSPS processes.

6. Managerial insights

This paper proposes a novel approach for production scheduling that has practical implications and offers valuable insights for different stakeholders. First, the proposed approach can help operations managers select the most energy-efficient production scheduling strategy without compromising production throughput. This dynamic sustainable production-scheduling model that takes energy costs into account

Table 16
Experimental results.

Instances Problem Size	Instance reference	RL approach cost (€)	GA cost (€)	FIFO cost (€)	CPLEX_1 cost (€)	CPLEX_2 cost (€)	MAS RE compared with CPLEX_1
	S1	2939	2947	3667	2939	2939	0.0 %
Small	S2	2258	2316	2914	2258	2258	0.0 %
(4,2,2)	S3	2305	2342	3298	2305	2305	0.0 %
	S4	2166	2201	2921	2166	2166	0.0 %
	S5	2143	2159	3153	2143	2143	0.0 %
Average		2362	2393	3191	2362	2362	0.0 %
	M1	6378	6351	7117	6330	6330	0.7 %
Medium	M2	6025	6113	7415	5944	5944	1.3 %
(8,3,3)	M3	6397	6429	7381	6333	6297	1.01 %
	M4	6689	6740	7479	6642	6642	0.7 %
	M5	6412	6571	6987	6359	6329	0.83 %
Average		6380	6402	72,767	6322	6308	0.9 %
	L1	10,005	10,846	11,763	9624	6924	3.9 %
	L2	9678	9874	11,475	9338	9338	3.6 %
Large (12,4,4)	L3	10,004	9931	11,468	9825	9825	1.8 %
	L4	10,322	10,428	12,389	10,089	10,089	2.3 %
	L5	9929	10,246	11,982	9767	9767	1.6 %
Average		9958	10,265	11,835	9619	9619	2.64 %

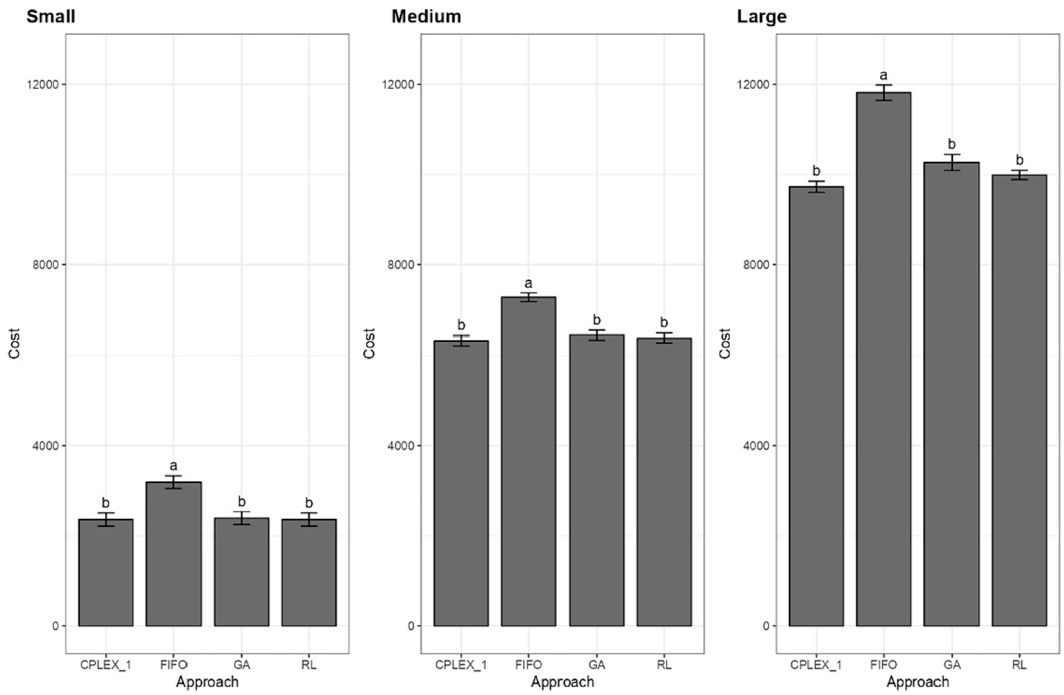

Fig. 23. Mean plots of costs (€) for the different approaches with LSD 95% confidence intervals.

Table 17
Average execution times.

	RL execution time	FIFO execution time	GA execution time	CPLEX_1	CPLEX_2
Small instances	2.3 s	0.14 s	4.73	856 s	135 s
Medium instances	4.8 s	0.18 s	13.52 s	+1000 s	+1000 s
Large instances	7.6 s	0.23 s	26.47 s	+ 1000 s	+1000 s

enables industries to optimize their operations and ultimately enhance their financial performance. Moreover, the integration of on-site renewable energy sources and energy storage systems provides additional advantages by reducing reliance on conventional grid power and lowering greenhouse gas emissions. Second, this study promotes governmental policies for sustainable development, by supporting in particular the adoption of renewable energy technologies through policy initiatives and regulatory frameworks. The findings highlight the potential benefits of integrating renewable energy sources into production processes, thereby contributing to the overall sustainability of the industry. Finally, based on the computational results, the proposed model and its implementation including a reinforcement learning strategy demonstrate high computational efficiency together as well as the ability to provide best solutions with minimal execution time. This suggests that such an approach could be integrated into either commercial or open-source software tools in order to offer more comprehensive and efficient solutions.

7. Conclusion

This study addresses the integrated lot sizing and production scheduling problem in a flexible flow line while considering energy policies. A novel mixed-integer mathematical model is formulated to solve this complex problem. The main objective of such a model is to optimize the overall cost, which includes setup costs, inventory costs, ESS utilization costs, RES costs, and conventional energy consumption costs. To address the complexity of the problem, a multi-level approach based on a dynamic Multi-Agent System architecture is developed. The MAS architecture is designed to simultaneously solve the lot sizing and production scheduling problem by incorporating multiple agents that interact and make cooperative decisions. The MAS utilizes the Q-learning algorithm to obtain good solutions within a reasonable timeframe. Numerical experiments were conducted to evaluate the performance of the proposed model and the multi-level resolution method. The results demonstrate that the mathematical model is accurate and effective, enabling a good tradeoff between production and energy performance. Moreover, the comparative analysis highlights the robustness and effectiveness of the MAS architecture in solving Integrated Lot Sizing and Production Scheduling (ILSPS) problems. This architecture outperforms other methods in terms of cost reduction and energy consumption optimization. Future work in this area is needed to account for the uncertainty in electricity prices and product demand in this first model. This can be achieved by enhancing the MAS to handle uncertain outcomes through different techniques such as stochastic modeling, scenario analysis, or robust optimization.

Funding

The first author is a PhD student whose thesis benefits from funding by the French Ministry of Agriculture and Food Sovereignty (MASA) and the Pays-de-la-Loire Region as part of the 2021–2026 State-Region Plan Contract (CPER).

8. Declarations

Ethics approval: not applicable.
Consent to participate: not applicable.
Consent for publication: not applicable.

CRediT authorship contribution statement

Mohamed Habib Jabeur: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Resources, Software, Visualization, Writing – original draft, Writing – review & editing, Validation. **Sonia Mahjoub:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing, Resources, Visualization. **Cyril Toublanc:** Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Supervision, Validation, Visualization, Writing – review & editing. **Veronique Cariou:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Resources, Supervision, Validation, Writing – review & editing, Visualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

Alves, F. F., Nogueira, T. H., De Souza, M. C., & Ravetti, M. G. (2021). Approaches for the joint resolution of lot-sizing and scheduling with infeasibilities occurrences. *Computers & Industrial Engineering, 155*, Article 107176. https://doi.org/10.1016/j.cie.2021.107176

Babaei, M., Mohammadi, M., & Ghomi, S. M. T. F. (2014). A genetic algorithm for the simultaneous lot sizing and scheduling problem in capacitated flow shop with complex setups and backlogging. *The International Journal of Advanced Manufacturing Technology, 70*, 125–134. https://doi.org/10.1007/s00170-013-5252-y

Basán, N. P., Cóccola, M. E., Dondo, R. G., Guarnaschelli, A., Schweickardt, G. A., & Méndez, C. A. (2020). A reactive-iterative optimization algorithm for scheduling of air separation units under uncertainty in electricity prices. *Computers & Chemical Engineering, 142*, Article 107050. https://doi.org/10.1016/j.compchemeng.2020.107050

Carvalho, D. M., & Nascimento, M. C. V. (2022). Hybrid matheuristics to solve the integrated lot sizing and scheduling problem on parallel machines with sequence-dependent and non-triangular setup. *European Journal of Operational Research, 296*, 158–173. https://doi.org/10.1016/j.ejor.2021.03.050

Cheng, L., Tang, Q., Zhang, L., & Yu, C. (2022). Scheduling flexible manufacturing cell with no-idle flow-lines and job-shop via Q-learning-based genetic algorithm. *Computers & Industrial Engineering, 169*, Article 108293. https://doi.org/10.1016/j.cie.2022.108293

Duarte, J. L. R., Fan, N., & Jin, T. (2020). Multi-process production scheduling with variable renewable integration and demand response. *European Journal of Operational Research, 281*, 186–200.

Eurostat. Energy. Transport and environment statistics - 2019 edition. 2019th ed. Luxembourg: Publications Office of the European Union; 2019. https://doi.org/10.2785/660147.

Golari, M., Fan, N., & Jin, T. (2017). Multistage stochastic optimization for production-inventory planning with intermittent renewable energy. *Production and Operations Management, 26*, 409–425. https://doi.org/10.1111/poms.12657

Gomes, E. R., & Kowalczyk, R. (2009). Dynamic analysis of multiagent Q-learning with -greedy exploration 8. *In Proceedings of the 26th annual international conference on machine learning*.

Kelley, M. T., Baldick, R., & Baldea, M. (2019). Demand response operation of electricity intensive chemical processes for reduced greenhouse gas emissions: Application to an air Separation unit. *Sustainable Chemistry & Engineering, 7*(2), 1909–1922. https://doi.org/10.1021/acssuschemeng.8b03927

Kelley, M. T., Baldick, R., & Baldea, M. (2020). Demand response scheduling under uncertainty: Chance- constrained framework and application to an air separation unit. *AIChE Journal*. https://doi.org/10.1002/aic.16273

Kelley, M. T., Pattison, R. C., Baldick, R., & Baldea, M. (2018). An MILP framework for optimizing demand response operation of air separation units. *Applied Energy, 222*, 951–966. https://doi.org/10.1016/j.apenergy.2017.12.127

Lei, K., Guo, P., Zhao, W., Wang, Y., Qian, L., Meng, X., & Tang, L. (2022). A multi-action deep reinforcement learning framework for flexible job-shop scheduling problem. *Expert Systems with Applications, 205*, Article 117796. https://doi.org/10.1016/j.eswa.2022.117796

Li, J., Sang, H., Han, Y., Wang, C., & Gao, K. (2018). Efficient multi-objective optimization algorithm for hybrid flow shop scheduling problems with setup energy consumptions. *Journal of Cleaner Production, 181*, 584–598.

Lin, J., Li, Y.-Y., & Song, H.-B. (2022). Semiconductor final testing scheduling using Q-learning based hyper-heuristic. *Expert Systems with Applications, 187*, Article 115978. https://doi.org/10.1016/j.eswa.2021.115978

Mahdieh, M., Bijari, M., & Clark, A. (2011). Simultaneous lot sizing and scheduling in a flexible flow line. *Journal of Industrial and Systems Engineering, 5*, 107–119.

Masmoudi, O., Yalaoui, A., Ouazene, Y., & Chehade, H. (2017). Lot-sizing in a multi-stage flow line production system with energy consideration. *International Journal of Production Research, 55*, 1640–1663.

Mazyavkina, N., Sviridov, S., Ivanov, S., & Burnaev, E. (2021). Reinforcement learning for combinatorial optimization: A survey. *Computers & Operations Research, 134*, Article 105400. https://doi.org/10.1016/j.cor.2021.105400

Van Moffaert, K. V., & Nowe, A. (2014). Multi-objective reinforcement Learning using sets of Pareto dominating policies. *The Journal of Machine Learning Research, 15*, 3483–3512.

Özdamar, L., & Barbarasoglu, G. (1999). Hybrid heuristics for the multi-stage capacitated lot sizing and loading problem. *Journal of the Operational Research Society, 50*, 810–825.

Peinado-Guerrero, M. A., & Villalobos, J. R. (2022). Using inventory as energy storage for demand-side management of manufacturing operations. *Journal of Cleaner Production, 375*, Article 134213. https://doi.org/10.1016/j.jclepro.2022.134213

R Core Team. (2021). *R: A language and environment for statistical computing*. Vienna, Austria: R Foundation for Statistical Computing. https://www.R-project.org/.

Rodoplu, M., Arbaoui, T., & Yalaoui, A. (2020). A fix-and-relax heuristic for the single-item lot-sizing problem with a flow-shop system and energy constraints. *International Journal of Production Research, 58*, 6532–6552. https://doi.org/10.1080/00207543.2019.1683249

Shrouf, F., Ordieres-Meré, J., García-Sánchez, A., & Ortega-Mier, M. (2014). Optimizing the production scheduling of a single machine to minimize total energy consumption costs. *Journal of Cleaner Production, 67*, 197–207. https://doi.org/10.1016/j.jclepro.2013.12.024

Sun, Z., Li, L., Fernandez, M., & Wang, J. (2014). Inventory control for peak electricity demand reduction of manufacturing systems considering the tradeoff between production loss and energy savings. *Journal of Cleaner Production, 82*, 84–93. https://doi.org/10.1016/j.jclepro.2014.06.071

Trevino-Martinez, S., Sawhney, R., & Sims, C. (2022). Energy-carbon neutrality optimization in production scheduling via solar net metering. *Journal of Cleaner Production, 380*, Article 134627. https://doi.org/10.1016/j.jclepro.2022.134627

Valledor, P., Gomez, A., Priore, P., & Puente, J. (2018). Solving multi-objective rescheduling problems in dynamic permutation flow shop environments with disruptions. *International Journal of Production Research, 56*, 6363–6377. https://doi.org/10.1080/00207543.2018.1468095

Wang, L., Hu, X., Wang, Y., Xu, S., Ma, S., Yang, K., Liu, Z., & Wang, W. (2021). Dynamic job-shop scheduling in smart manufacturing using deep reinforcement learning. *Computer Networks, 190*, Article 107969. https://doi.org/10.1016/j.comnet.2021.107969

Wang, S., Mason, S. J., & Gangammanavar, H. (2020). Stochastic optimization for flow-shop scheduling with on-site renewable energy generation using a case in the United States. *Computers & Industrial Engineering, 149*, Article 106812.

Yan, Q., Wang, H., & Wu, F. (2022). Digital twin-enabled dynamic scheduling with preventive maintenance using a double-layer Q-learning algorithm. *Computers & Operations Research, 144*, Article 105823. https://doi.org/10.1016/j.cor.2022.105823

Xiao, J., Yang, H., Zhang, C., Zheng, L., & Gupta, J. N. D. (2015). A hybrid lagrangian-simulated annealing-based heuristic for the parallel-machine capacitated lot-sizing and scheduling problem with sequence-dependent setup times. *Computers & Operations Research, 63*, 72–82. https://doi.org/10.1016/j.cor.2015.04.010

Zou, F., Yen, G. G., Tang, L., & Wang, C. (2021). A reinforcement learning approach for dynamic multi-objective optimization. *Information Sciences, 546*, 815–834. https://doi.org/10.1016/j.ins.2020.08.101