A reinforcement learning approach for a lot sizing and production scheduling problem with energy consideration

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

Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 56-2 (2023) 11141–11147

A reinforcement learning approach for a lot sizing and production scheduling problem with energy consideration

Mohamed Habib Jabeur*. Sonia Mahjoub. Cyril Toublanc***. Veronique Cariou******

** Oniris, INRAE, STATSC, 44300 Nantes, France (Tel: +33617619991; e-mail: mohamed-habib.jabeur@oniris-nantes.fr).*

***Oniris, Nantes université, LEMNA, CS 82225, 44322 Nantes, France (e-mail: sonia.mahjoub@oniris-nantes.fr)*

****Oniris, Nantes université, CNRS, GEPEA, UMR 6144, F-44000 Nantes, France, (e-mail: cyril.toublanc@oniris-nantes.fr)}*

***** Oniris, INRAE, STATSC, 44300 Nantes, France (e-mail: veronique.cariou@oniris-nantes.fr)*

Abstract: with climate change, many companies are looking to reduce their carbon footprint and ensure a sustainable manufacturing. To meet this challenge, one of the alternatives is to replace carbon intensive processes with low-carbon processes involving electrical and/or renewable energies. Within this scope, a novel scheduling approach is proposed to take into account the introduction of onsite renewable energy. In particular, a lot sizing and production-scheduling problem in flexible flow line with renewable energy integration is formulated as a versatile optimization model. With regard to associated complexity issues, a multi-agent reinforcement learning approach is advocated to solve the lot sizing and scheduling problem. Finally, the approach is evaluated with a benchmark case and other numerical experiments.

Keywords: flexible flow line, renewable energy, energy optimization, multi-agent reinforcement learning

1. INTRODUCTION

With climate change issue, sustainable manufacturing has gaining significant momentum in achieving the challenging target of zero carbon emission. The fast depletion of fossil fuels and their related environmental impacts are gradually prompting industries to turn to more sustainable renewable energy sources. In comparison with traditional energies, renewable energy is a zero carbon energy resource with the potential to secure energy supplies and reduce emissions (Raihan and Tuspekova, 2022). Within this context, several studies have been conducted over the past few decades to integrate onsite renewable energy sources (RES) into manufacturing energy supply systems. Li et al. (2017) designed a carbon-free power model to plan for the integration of onsite renewable energy into manufacturing plants. Duarte et al. (2020) proposed a novel approach to achieve low carbon emissions through the integration of onsite RES and Energy Storage System (ESS) into a multi-process production system. Golari et al. (2017) developed a low-carbon production-inventory solution by incorporating onsite RES into a multi-factory manufacturing system. Wang et al. (2020) presented a scheduling model for flow shops adapted to match RES with ESS and main grid power. The production scheduling problem with energy consideration was also studied by Masmoudi et al. (2017). These authors explored a single item capacitated lot sizing problem in a flow shop system. The objective of the study was to determine the quantities of items produced by each machine, which optimize production and energy costs. A two heuristics were developed to address the complex lot sizing problem. Masmoudi et al. (2016) formulated a mixed integer linear program (MILP) to discuss a multi-item capacitated lot sizing and scheduling problem in a flow shop system, which includes an energy management approach. A fix-and-relax (RF) algorithm was proposed to overcome computational complexity. This algorithm was also applied by Rodoplu et al. (2020) to address the problem of single-item lot sizing for flow shops under multiple energy constraints. Li et al. (2018) studied the hybrid flow shop scheduling problem with setup energy consideration. An energy-aware multi-objective optimization algorithm was proposed (EA-MOA) to optimize the makespan and energy consumptions.

As demonstrated in the literature, integrating energy efficiency into the lot sizing and production scheduling problem in flexible flow shop is strongly NP-hard. As a result, various methods including heuristic and meta-heuristic optimization algorithms have been developed to solve these problems in a reasonable time frame (Yan et al., 2022). In particular, several artificial intelligence approaches have been advocated to achieve the optimal balance between maximizing the efficiency of the optimization and minimizing the time required for execution. Among those we can cite; reinforcement learning (RL) algorithms (Yan et al. 2022). For instance, Waschneck et al. (2018) proposed a deep Q network agent RL algorithm to solve production scheduling in a semiconductor system. Xue et al. (2018) applied Q-learning approach to minimize the total makespan of an automated guided vehicles (AGV) scheduling problem. Wang et al.

10.1016/j.ifacol.2023.10.832

(2022) formulated resource preemption environment RPE as a decentralized Markov Decision Process (MDP) and proposed to apply a multi-agent reinforcement learning (MARL) approach. Wang (2020) presented a scheduling strategy using a multi-agent reinforcement learning approach to deal with real-time job insertions in job shop scheduling environment. A weighted Q-learning algorithm was used to optimize makespan and penalties for lateness.

This paper addresses the simultaneous problem of lot sizing and production scheduling in a flexible flow line (FFL) with a focus on energy efficiency. To the best of author's knowledge, there is a gap in the research on integrated lot sizing and sustainable production scheduling for FFLs with energy consideration. In response, we propose a novel multi-products and serial multi-process optimization model that integrates renewable energy and energy consumption optimization into the FFL system. A cooperative multi-agent system based on Q-learning methodology is developed to solve the proposed model. The Q-agents are trained in the FFL environment with the objective of meeting customer's demand while optimizing energy and setups costs. Two level training algorithms are designed to achieve the global cost optimization objective.

The rest of the paper is organized as follows. Section 2 presents the problem formulation. Section 3 details the proposed multi-agent architecture for solving the problem. In section 4, the effectiveness of the proposed approach is evaluated on the basis of numerical experiments. Finally, the conclusion is fully outlined in section 5.

2. PROBLEM DESCRIPTION

2.1 Context

This paper explores an integrated sustainable lot sizing and production scheduling problem in a FFL system powered by different energy sources: the conventional power grid, on-site PV solar panels and an ESS. This production system in question consists of several successive processes, each of which consists of one or more non-identical parallel machines.

2.2 Model assumptions

The proposed model is based on the assumptions listed below:

- A two-level time scale is considered. The scheduling horizon is split into T macro-periods, each divided into the same a priori number of micro-periods F with a consistent duration across all micro-periods.
- All demands are satisfied at each macro-period with no backlog.
- There is no buffer between stages, but instead a queue may appear between processes. It is worth to note that waiting costs and time are not taken into consideration herein.
- All items must go through all processes to obtain the final product.
- Lot splitting is not allowed at any process; an item can only be manufactured by one of the parallel machines.
- A product made at a process will be available at the next micro-period.
- At the beginning of the scheduling horizon, each machine is setup for a specific item i_{empty}.
- A setup occurs only at the beginning of a micro-period.
- Only one setup is scheduled during micro-period.
- Setup times are sequence dependent and may vary from one product to another.
- Machines consume energy during setup times and waiting times.

2.3 Notations

Sets and indices:

T, t :	Set and index of macro-periods in production horizon.
F, f:	Set and index 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.

Parameters:

$D_{i,t_{\|F\|}}$	Demand for product i at the end of macro-period t.
$\tau_{m,p}^{i}$	Processing time of product i with machine m of process p.
$s_{m,p}^{i,j}$	Setup time from product i to product j with machine m of process p.
$C_{m,p}^{i,j}$	Setup cost from product i to product j with machine m of process p.
d	Micro-period duration.
M	A big real number.
s_{max}	Maximum storage capacity of the ESS.
s_{min}	Minimum storage capacity of 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.
η^+/η^-	ESS charging and discharging efficiency.
$E_{i,m,p}^{on}$	Power required to produce one unit of product i with machine m of process p.
$E_{m,p}^{off}$	Consumed power in one time unit when machine m of process p is inactive.
G_{t_f}	Available renewable energy at micro-period f from macro-period t.
g_{t_f}	Conventional energy price at micro-period f from macro-period t.
r_{t_f}	MWh PV power cost at micro-period f from macro-period t.

Variables:

x_{m,p,t_f}^{i}	Quantity of product i produced by machine m of process p at micro-period f of macro-period t.
XT_{p,t_f}^{i}	Amount of product i produced in process p at micro-period f of macro-period t and transferred to the next process.
XS_{p,t_f}^{i}	Amount of product i produced in process p at micro-period f of macro-period t and waiting for production at next process.
Q_{p,t_f}^{i}	Amount of product i waiting for production at process p in micro-period f of macro-period t.

QT^{i}_{p,t_f} Amount of product i in queue at process p and progress to next process $p+1$ at micro-period $f+1$.

QS^{i}_{p,t_f} Amount of product i in queue at process p and still waiting in queue at process p at the next micro-period.

$I^{i}_{t_f}$ Inventory level of final product i at micro-period f of macro-period t.

$y^{i,j}_{m,p,t_f}$ = 1, if a setup from product i to product j occurs for machine m of process p at the beginning of micro-period f, otherwise = 0.

w^{i}_{m,p,t_f} = 1, if machine m of process p is setup for product i in the micro-period f of macro-period t, 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 micro-period f of macro-period t.

er_{t_f} Renewable energy consumed at micro-period f of macro-period t.

ec_{t_f} Grid conventional energy consumed at micro-period f of macro-period t.

2.4 Mathematical model

$$C_{total} = \sum_{f=1}^{F}\sum_{t=1}^{T}\sum_{p=1}^{P}\sum_{m=1}^{M_p}\sum_{I=1}^{N}\sum_{j=1}^{N} C^{i,j}_{m,p}.y^{i,j}_{m,p,t_f} + \sum_{f=1}^{F}\sum_{t=1}^{T} g_{t_f}.ec_{t_f} + r_{t_f}.er_{t_f} + C^{+}.r^{+}_{t_f} + C^{-}.r^{-}_{t_f} \quad (1)$$

$$\sum_{m=1}^{M_p} x^{i}_{m,|P|,t_{|F|}} + I^{i}_{t_{|F|-1}} - I^{i}_{t_{|F|}} \geq D_{i,t_{|F|}}, \forall i \in N,\ \forall t \in T \quad (2)$$

$$x^{i}_{m,p,t_f} = XT^{i}_{p,t_f} + XS^{i}_{p,t_f}, \forall i \in N,\ \forall m \in M_p,\ \forall p \in P,\ \forall t \in T, \forall f \in F \quad (3)$$

$$Q^{i}_{p,t_f} = QT^{i}_{p,t_f} + QS^{i}_{p,t_f}, \forall i \in N,\ \forall p \in P,\ \forall t \in T, \forall f \in F \quad (4)$$

$$x^{i}_{m,p,t_f} = XT^{i}_{p-1,t_{f-1}} + QT^{i}_{p-1,t_{f-1}},\ \forall i \in N,\ \forall m \in M_p,\ \forall p \in \{2,\dots,|P|\},\ \forall t \in T, \forall f \in F \quad (5)$$

$$Q^{i}_{p,t_f} = QS^{i}_{p,t_{f-1}} + XS^{i}_{p,t_f} \quad , \quad \forall p \in P, \forall i \in N, \forall t \in T, \forall f \in F \quad (6)$$

$$I^{i}_{t_f} = I^{i}_{t_{f-1}} + x^{i}_{m,|P|,t_{f-1}}, \forall m \in M_{|p|}, \forall i \in N, \forall t \in T, \forall f \in \{2,\dots,|F|-1\} \quad (7)$$

$$\tau^{i}_{m,p}.x^{i}_{m,p,t_f} + \sum_{j=1}^{N} s^{j,i}_{m,p}.y^{j,i}_{m,p,t_f} \leq d,\ \forall i \in N,\ \forall m \in M_p, \forall p \in P, \forall t \in T, \forall f \in F \quad (8)$$

$$x^{i}_{m,p,t_f} \leq M.w^{i}_{m,p,t_f}, \forall i \in N,\ \forall m \in M_p,\ \forall p \in P,\ \forall t \in T,\ \forall f \in F, \quad (9)$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N} y^{i,j}_{m,p,t_f} \leq 1,\ \forall m \in M_p,\ \forall p \in P,\ \forall t \in T,\ \forall f \in F \quad (10)$$

$$\sum_{i=1}^{N} w^{i}_{m,p,t_f} = 1, \forall m \in M_p,\ \forall p \in P, \forall t \in T, \forall f \in F \quad (11)$$

$$w^{i_{empty}}_{m,p,t_0} = 1, \forall m \in M_p,\ \forall p \in P, t = 1 \quad (12)$$

$$y^{i,j}_{m,p,t_f} + 1 \geq w^{i}_{m,p,t_{f-1}} + w^{j}_{m,p,t_f}, \forall i \in N,\ \forall j \in N, \forall m \in M_p, \forall t \in T,\ \forall p \in P, \forall f \in F \quad (13)$$

$$y^{i_{empty},j}_{m,p,t_1} + 1 \geq w^{j}_{m,p,t_1} + w^{i_{empty}}_{m,p,t_0},\ \forall j \in N\ \forall m \in M_p,\ \forall p \in P,\ \ t = 1 \quad (14)$$

$$y^{i,j}_{m,p,t_1} + 1 \geq w^{i}_{m,p,t-1_{|F|}} + w^{j}_{m,p,t_1},\ \forall i \in N,\ \forall j \in N, \forall m \in M_p, \forall t \in \{2,\dots,|T|\},\ \forall p \in P \quad (15)$$

$$S_{min} \leq ESS_{t_f} \leq S_{max} \quad , \forall t \in T, \forall f \in F \quad (16)$$

$$ESS_{t_f} = ESS_{t_{f-1}} + \eta^{+}.r^{+}_{t_f} - r^{-}_{t_f}\frac{1}{\eta^{-}} \quad , \forall t \in T, \forall f \in F \quad (17)$$

$$r^{+}_{t_f} \leq R^{+} \quad , \forall t \in T, \forall f \in F \quad (18)$$

$$r^{-}_{t_f} \leq R^{-} \quad , \forall t \in T, \forall f \in F \quad (19)$$

$$\sum_{i=1}^{N}\sum_{m=1}^{M_p}\sum_{p=1}^{P}(E^{on}_{m,p}.x^{i}_{m,p,t_f}.\tau^{i}_{m,p} + \sum_{j=1}^{N} y^{i,j}_{m,p,t_f}.s^{i,j}_{m,p}.E^{off}_{m,p}) + \sum_{i=1}^{N}\sum_{m=1}^{M_p}\sum_{p=1}^{P}(d - (x^{i}_{m,p,t_f}.\tau^{i}_{m,p} + \sum_{j=1}^{N} y^{i,j}_{m,p,t_f}.s^{i,j}_{m,p}.E^{off}_{m,p})) = ec_{t_f} + er_{t_f} + r^{-}_{t_f}, \forall t \in T, \forall f \in F \quad (20)$$

$$er_{t_f} + r^{+}_{t_f} \leq G_{t_f}, \forall t \in T, \forall f \in F \quad (21)$$

$$x^{i}_{m,p,t_f}, XT^{i}_{p,t_f}, XS^{i}_{p,t_f}, Q^{i}_{p,t_f}, QT^{i}_{p,t_f}, QS^{i}_{p,t_f}, I^{i}_{t_f}, ESS_{t_f}, r^{+}_{t_f}, r^{-}_{t_f}, ec_{t_f}, er_{t_f} \geq 0, \forall i \in N, \forall p \in P, \forall m \in M_p, \forall t \in T, \forall f \in F \quad (22)$$

$$y^{i,j}_{m,p,t_f}, w^{i}_{m,p,t_f} \in \{0,1\} \quad \forall i \in N,\ \forall j \in N, \forall p \in P, \forall m \in M_p,\ \forall t T, \forall f \in F \quad (23)$$

The objective function (1) aims to minimize the costs of setup and energy consumption while constraint (2) presents the flow balance constraints. Constraints (3) and (4) model the items production and waiting queue in process p at micro-period f. Constraints (5) and (6) express the flow balance in process p at micro-period f. Meaning that if an amount of a product i is transferred from process p to the next one, but there is no available machine for processing or the transferred quantity cannot be processed in one micro-period, a part of the whole quantity will be processed in the current micro-period and the rest will wait for the next micro-period. Constraint (7) ensures that the final products are stored in the final inventory awaiting shipment. The production capacity of machine m is given by constraint (8). Constraint (9) ensures that an item i can only be processed on a machine m if it is configured for that item. Constraint (10) ensures that there is at most one machine setup per micro-period while constraint (11) ensures that only one setup state can be defined in a micro-period. The initial setup is specified by constraint (12), which sets the configuration of machines to product i_{empty} before the start of processing. Constraints (13) to (15) express the link between binary variables. The energy level at ESS is bounded by S_{max} and S_{min} as shown in (16). Constraint (17) updates the state of the ESS after charging and discharging. Constraints (18) and (19) model the charging and discharging capacities. Constraint (20) ensures that the energy needs of the FLL are met by On-site PV energy, ESS and conventional energy. Constraint (21) models the on-site PV power supply capacity. Non-negativity and binary requirements are modeled in (22) and (23)

3. REINFORCEMENT LEARNING APPROACH

The model detailed in section 2 is formulated as a multi-agent system (MAS) in which cooperative agents interact sequentially in the same environment in order to obtain a near optimal schedule and product sequence. The FFL environment defines the number of agents, their relationships, the number of products, processing and setup time and the energy consumed by the machines. To improve solution quality, agents share experiences and knowledge with each other. In situations where decision-makers have partial control over outcomes, the interaction process can be modeled using a Markov Decision Process (MDP). The proposed reinforcement learning approach in this study is modeled as a factored m-agent Decentralized Markov decision Process (DEC-MDP) consisting of a tuple of $(A_g, S, U, T, R, \delta, O)$. $A_g = A_p \times A_m$ represents the set of agents. It is worth noting that FFL is divided into two problems: routing problem

(assigning a product to one of the parallel machines) and the sequencing problem. Therefore, A_g is also divided into two parts: |P| agents associated with processes and $|M_p|$ agents associated to machines. Since this problem involves two different types of agents, a Q-learning approach is associated to each type. The m-factored DEC-MDP is characterized by a factored state space $S = S_m \times S_p$, where S_m represents the states of the machine agents and S_p represents the state of the process agents . Each agent a $\in A_g$ is capable of selecting an action u from the action space $U = U_m \times U_p$, where U_m is the action space of the machine agents and U_p is the action space of the process agents. Each agent has an observation o $\in O$ according to the observation function $\delta(agent, u)$: $A_g \times U \rightarrow O$. R represents agent's reward function. The transition function $T(s, u, s')$ enables us to determine the probability that the system will change its state from s to s' after performing action u.

3.1 Process agents

The set A_p consists of the FFL processes where an agent k is associated with each process p with parallel machines. This learning phase addresses the routing problem in terms of processing time, machine availability and machine energy consumption. As a result, each agent learns how to choose the most suitable machine for a product to perform its processing operation.

State representation: The space state S_p represents the local information of the actual scheduling environment at processes, including the remaining products to be processed. Each process agent has a limited local view as it only has information about its associated available machines and products waiting for production.

Action representation: The action space is the set of agents' behavior. Making an action for an agent involves selecting which machine from the available ones to process the product. Then, the action set U_p is defined as set of process machines.

Reward representation: Since the objective is to minimize the total energy and setups costs, the reward is related to a lower cost. For scheduling steps, the objective is to select the machine with the lowest corresponding energy consumption. The reward function is defined as follows:

$$R_p(m,i) = \begin{cases} (\Delta t + S^{j,i}_{m,p}).E^{off}_{m,p} + E^{on}_{i,m,p}.x^i_{m,p,t_f}, f + \tau^i_{m,p}.x^i_{m,p,t_f} \le |F| \\ big\ number\ M\ , \quad f + \tau^i_{m,p}.x^i_{m,p,t_f} > |F| \end{cases}$$

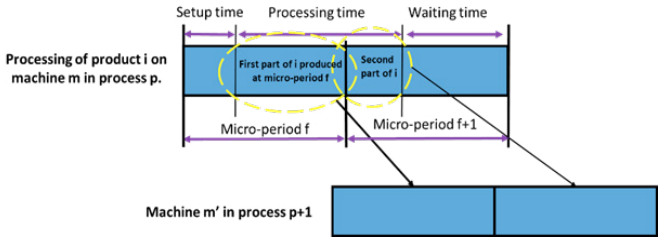

Fig. 1. Workflow between machines.

Here, m is the selected action (machine), i is the current product to be processed and j is the previous product processed on the chosen machine. If i is the first product to be processed on m, $C^{j,i}_{m,p} = 0$. Δt is the machine waiting time to process the next product at the next micro-period. In practice, a machine can only perform a changeover at the beginning of a micro-period. In some cases, product processing ends during the micro-period, which forces the machine to wait until the beginning of the next micro-period to select another product and to execute its changeover (Fig. 1).

3.2 Machine agents

The machine agents set A_m represents the production resources where an agent k is associated to each machine m. A machine agent cannot take action at every micro-period, but only after the current product has been processed (constraint 11). Additionally, the machine agent can only choose the next product to be processed at the beginning of the micro-period after the setup time has passed. Moreover, the agent is limited to processing at most one product during a micro-period due to the setup constraint. Machine agents aim to optimize the sequence of products; i.e., they have to determine the most cost-effective order in which to process the products

State and action representation: Machine agents make decisions about selecting available products selection. This means that they choose an action from the waiting and remaining products at the corresponding machines. Therefore, the space of states S_m is defined as the set of remaining products and the space of actions U_m corresponds to products selection.

Reward representation: The objective of the sequencing problem in FFL is to minimize energy and setup costs. These costs are dependent on the setup time and machine waiting time. Therefore, the reward function R_m for machine agents is defined as follows:

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

3.3 Agent Policy model

At each micro-period, an agent uses the observation as a direct input to its policy model to determine how to select an action in order to achieve the maximum cumulative rewards in the long run. As a result, comprehensive information about the agent can be gathered. The observation of an agent is represented as:

o_p = {available machines observations, products processing time, current micro-period, products to be produced and available quantities}.

o_m= {current product, product waiting for process, affected products, current micro-period}.

Where o_p is the process agent observation and o_m is the machine agent observation. The agent policy model is based on Q-learning system, which learns through an action-value function. Each state-action pair has an associated Q-value which is updated based on the reward received after action selection. The objective of this study is to minimize production costs and the update rule for the state action pair (s, u) is given as follows:

$$Q(s,u) = (1-\alpha)\, Q(s,u) + \alpha\, [r + \gamma * min_{u'}\, (Q(s',u') - Q(s,u))] \quad (24)$$

The input of the models consists of agent observation and action. When an agent reaches state S, it has two options: either exploit the previous experience by selecting the best action based on the associated Q-value, or select an action randomly (exploration). The action selection method considered in this paper is based on the $\varepsilon - greedy$ strategy due to its success in multi-agent environment (Gomes and Kowalczyk, 2009). The agent's interaction and the entire MARL approach are presented in Fig. 2.

The proposed Q-learning algorithms are summarized in algorithms 1 and 2.

Algorithm 1. Process agent QL

Initialize:
$Q(s,u) = zeros\_matrix(products_{number}, machines_{number})$
for each episode **do**:
  Initialize:
    $\boldsymbol{S}$ = [list of] remaining products to be processed
    Possible_actions = [list of] available machines
  for micro-period = $\boldsymbol{f}$, product = $\boldsymbol{s}$ **do**:
    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 $(\boldsymbol{s},\boldsymbol{u})$
    $\boldsymbol{S = S'}$
    Return selected machine and process observations.
  end for
end for

Algorithm 2. Machine agent QL

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

4. COMPUTATIONAL RESULTS

4.1 Benchmark Input data

The proposed MILP is solved using PULP 2.4.1. The MARL approach is coded from scratch in python 3.9 and runs on a PC with a Core ™ i7-11850 H 2.5 GHz CPU with 32GB of RAM. The computation times are measured in CPU seconds. In this section, a FFL benchmark problem is conducted to explore the proposed model and the MARL method.

As shown in Fig. 3, the FFL production system consists of three processes where the first two processes have a single machine and the third one has three non-identical parallel machines. To apply the proposed approach, one process agent is required for the third process. However, process 1 and process 2 are modeled with their corresponding machines agents. On-site PV panels, ESS and the conventional grid meet the electrical demand of the studied FFL. The production horizon considered in this benchmark is composed of one macro-period; which is divided into 8 micro-periods. Three kinds of products are considered. As in Özdamar and Barbarasoglu's (1999), production model parameters are randomly generated as follows: products demand $D_{i,t_{|F|}}$ is obtained from a uniform distribution $U(40,90)$. Processing times $\tau^i_{m,p}$ (in time unit) are generated from $U(1,3)$. Setup times are proportional to total processing time: $s^{i,j}_{m,p} = \frac{S\sum_{j,t,m}\tau^j_{m,p} * D_{i,t_{|F|}}}{|T| * max_p(|M_P|)}$. Where the prarameter S is generated from $U(0.05,0.01)$. Tables 1, 2 and 3 present the demands, processing times, setup times and costs, respectively (Setup parameters of machines M3, M4 and M5 are not used in the presented solution and hence not presented herein). Energy parameters adjusted from Duarte et al. (2020), are presented in table 4. $s_{min} = 0.4\ MWh$, $s_{max} = 0.6\ MWh$ and $s_0 = 0.6\ MWh$. $R^+ = R^- = 0.5\ MWh$, $\eta^+ = \eta^- = 1$, and $c^+ = c^- = 30$ €. $g_{t_f} = 90$ € and $r_{t_f} = 50$ €. $E^{off}_{m,p}$ is set to 0.005 MWh/(time unit). The other energy parameters are presented in tables 4 and 5. The agent's learning parameters are set as follows: α = 0.2, γ = 0.5 and ε = 0.5.

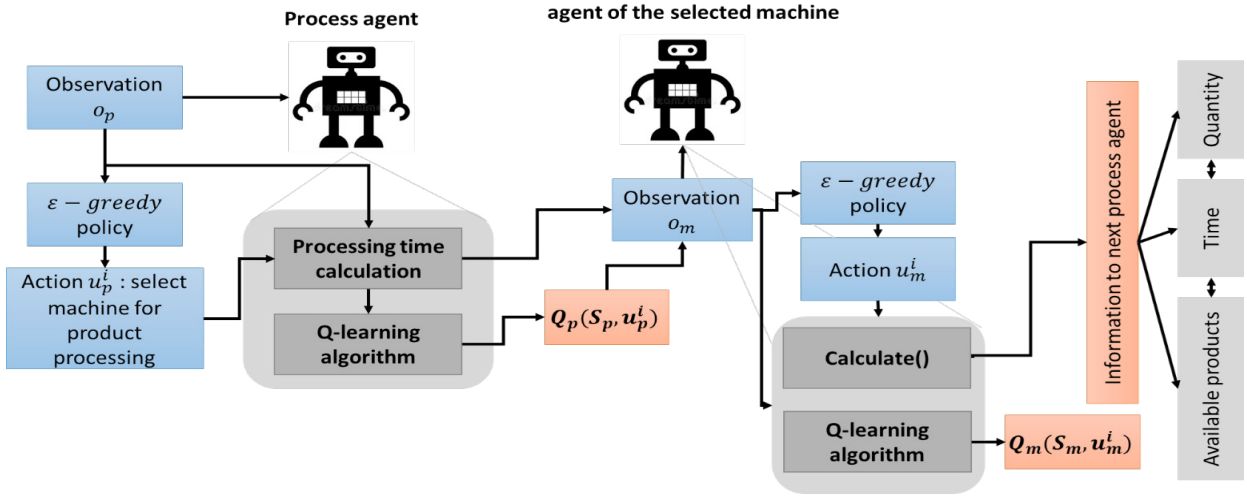

Fig. 2. The MARL model for the FFL scheduling problem.

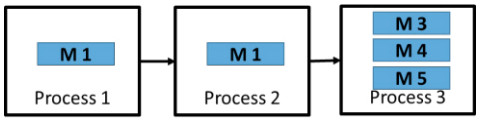

Fig. 3. FFL processes

Table 1. Demands

Macro-P \ products	P1	P2	P3
t = 1	57	80	44

Table 2. Processing times.

Machine \ products	P1	P2	P3
M1	2	1,3	1,35
M2	1,8	1,3	1,3
M3	2,5	1,8	2
M4	1,7	2	1,2
M5	2,2	1,3	2,2

Table 3. Setup times and costs for two machines.

Machine	Setup time (minutes)			Cost (€)			
M1		P1	P2	P3	P1	P2	P3
	P1	0	7	3	0	80	44
	P2	8	0	6	86	0	62
	P3	6	9	0	60	94	0

Machine	Setup time (minutes)			Cost (€)			
M2		P1	P2	P3	P1	P2	P3
	P1	0	9	5	0	99	58
	P2	9	0	3	93	0	32
	P3	9	11	0	102	115	0

Table 4. Machines energy consumption (MWh/unit)

Machines\ product	P1	P2	P3
M1	0.02	0.015	0.014
M2	0.019	0.013	0.014
M3	0.025	0.019	0.02
M4	0.017	0.019	0.013
M5	0.023	0.015	0.024

Table 5. PV energy availability

f	1	2	3	4	5	6	7	8
G_{t_f}(MW)	0	0.23	0.52	0.4	0.32	0.36	0.21	0

4.2 Results

Based on the parameters mentioned above, the learning process of the QL and the resulting scheduling scheme are presented in Fig. 4 and 5, respectively. In Fig 4, it is apparent that the total cost, set to 1097 euros, reached after only nine learning episodes. The total execution time is 0.7 second. In Fig. 5, the Gant chart displays the scheduling scheme for processing P1, P2 and P3. It can be verified that there is no overlap between the different processing steps of the same product, and the next processing step is executed only when the previous processing step is completed. Since only machine M1 and machine M2 are involved in the sequencing problem, the chosen sequence is P2-P3-P1 which leads to a lowest set setup cost of 256€ and waiting cost of 69.3€. For process 3, each machine will process only one product. In order to avoid a high set up cost, the process agent 3 assigns P3 to M3 at $f = 5$ and $f = 6$, P1 to M4 at $f = 7$ and $f = 8$ and P2 to M5 at $f = 3$ and $f = 4$. The choice of this scheduling solution is based on energy consumption and waiting cost criteria. According to the proposed scheduling solution, the energy consumption plan is illustrated in Fig 6. It is evident that all generated PV power in the scheduling horizon is consumed which contributes in the optimization of the total energy cost. Based on the above performance evaluation, both the high speed and efficiency of FFL scheduling using QL have been demonstrated. Thus, the proposed QL algorithms can be applied to different problems with different magnitudes.

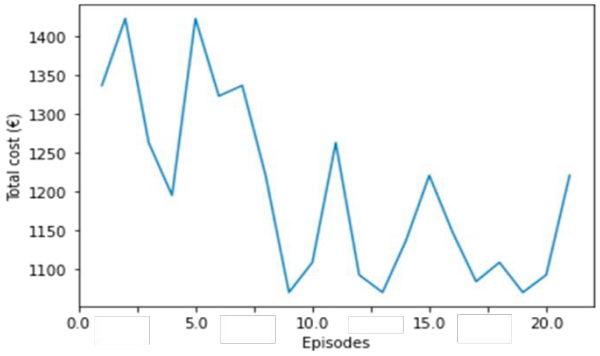

Fig. 4. Benchmark case learning

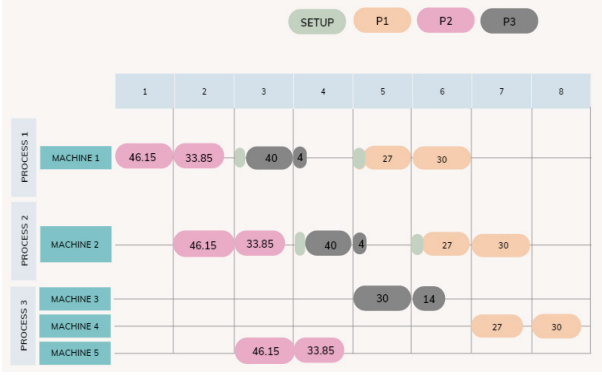

Fig. 5. Gant diagram for obtained solution.

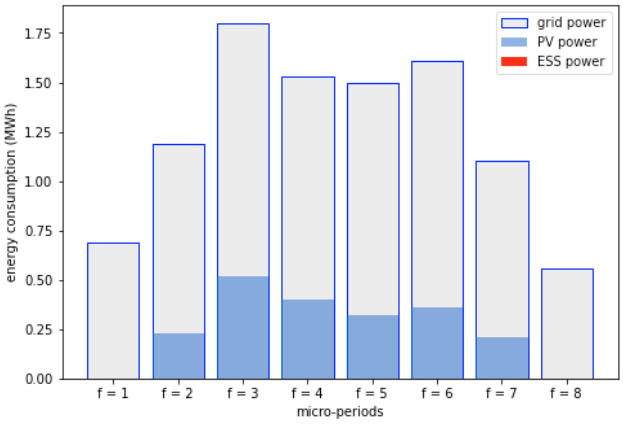

Fig. 6. Energy consumption

4.3 Other experiment results

Inspired from Özdamar and Barbarasoglu (1999), several problem instances are generated to evaluate the efficiency of the RL algorithm. For each problem size (N, P, M_p), it is assumed that T= 1 and F= 8. The performance of our RL approach is compared with the performance of the CEPLEX Solver and the well Known genetic algorithm (GA) in terms of solution quality and CPU time. The GAP criteria computed as: $\frac{re - op}{op} * 100$, where re is the obtained cost by the RL approach and op is the optimal cost of the problem, is considered in this comparative study.

The results of our experiments are reported on table 6. Then, it can be observed that all simulations carried out by the RL approach are not exceed 20 s while the CEPLEX solver reach the optimal solution within 1000 s. In addition, it can provide near optimal solutions for all instances. Specifically, for small problems, the algorithm can reach the optimal solution. As seen in table 6, the GAP increases as the problem size grows. Compared, to the GA, the RL approach is capable of obtaining competitive results. The GA seeks to find a good solution from randomly generated ones, while the RL approach employs a sequential learning process to accelerate the convergence of solutions. Therefore, for large problems, the RL approach can rapidly obtain better solution whose gap is equal to 1.8%.

Table 6. Comparative study

Instance	RL cost (€)	GA cost (€)	Optimum cost (€)	RL Gap (%)	GA Gap (%)	CPU_{RL} (s)	CPU_{ceplex} (s)
Small (4,2,2)	915.2	915.2	915.2	0	0.0	2.3 s	856 s
Medium (8,3,3)	3387.1	3382.9	3358.7	0.84	0.72	7.8 s	+1000 s
Large (12,4,4)	8533.4	8665.7	8381.9	1.8	3.38	15 s	+1000 s

5. CONCLUSIONS

In this paper, we proposed a mathematical model for lot sizing and production scheduling in FFL with the aim of minimizing energy consumption without compromising production constraints. Since the proposed MILP is NP-Hard, a multi-agent reinforcement learning approach was adapted to the proposed model and evaluated based on a benchmark test. The effectiveness and efficiency of the proposed QL algorithm for FFL scheduling were demonstrated in this study. In future work, we aim to integrate uncertainties related to PV power generation into our approach.

.

REFERENCES

Gomes, E.R., Kowalczyk, R., 2009. Dynamic Analysis of Multiagent Q-learning with -greedy Exploration 8.

Li, B., Tian, Y., Chen, F., Jin, T., 2017. Toward net-zero carbon manufacturing operations: an onsite renewables solution. Journal of the Operational Research Society 68, 308–321.

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.

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.

Masmoudi, O., Yalaoui, A., Ouazene, Y., Chehade, H., 2016. Multi-item capacitated lot-sizing problem in a flow-shop system with energy consideration. IFAC-PapersOnLine 49, 301–306.

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

Raihan, A., Muhtasim, D.A., Pavel, M.I., Faruk, O., Rahman, M., 2022. Dynamic impacts of economic growth, renewable energy use, urbanizatio, ans tourism on carbon dioxide emissions in Argentina. Environmental processes. 9, 38.

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.

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

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, 106812.

Wang, X., Zhang, L., Lin, T., Zhao, C., Wang, K., Chen, Z., 2022. Solving job scheduling problems in a resource preemption environment with multi-agent reinforcement learning. Robotics and Computer-Integrated Manufacturing 77, 102324.

Wang, Y.-F., 2020. Adaptive job shop scheduling strategy based on weighted Q-learning algorithm. J Intell Manuf 31, 417–432.

Waschneck, B., Reichstaller, A., Belzner, L., Altenmüller, T., Bauernhansl, T., Knapp, A., Kyek, A., 2018. Optimization of global production scheduling with deep reinforcement learning. Procedia CIRP 72, 1264–1269.

Xue, T., Zeng, P., Yu, H., 2018. A reinforcement learning method for multi-AGV scheduling in manufacturing, in: 2018 IEEE International Conference on Industrial Technology (ICIT).

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, 105823. https://doi.org/10.1016/j.cor.2022.105823