(04-010) - Development of an artificial intelligence algorithm for indoor air quality optimization and industrial production management

Cebolla-Alemany, Joaquim [1]; Macarulla Martí, Marcel [1]; Viana, Mar [2]; Gassó-Domingo, Santiago [1]

[1] Universitat Politècnica de Catalunya, [2] IDAEA-CSIC

Nanoparticle modelling allows concentration simulations in industrial settings to estimate the effect of different air extraction strategies in scenarios with nanoparticle emitting processes. Several artificial intelligence-based techniques can evaluate these strategies to find the optimal one. Moreover, they can simultaneously minimise the energy cost of the extraction process by coordinating the industrial activity with the hourly grid energy cost fluctuation. Consequently, two artificial intelligence algorithms are proposed based on genetic algorithms and reinforcement learning. For the first, a population generator manages system's restrictions based on real operative scenarios and then individuals change through time imitating natural selection, reproduction and mutation processes. For the second, a meta-heuristics policy is designed from state space and actions consisting on different heuristic strategies to explore potential solutions. Preliminary results evaluating the energy cost performance show that both algorithms reach similar solutions, registering the expected population features curve for the genetic algorithm but not illustrating a clear learning curve for the reinforcement learning study.

Keywords: Energy; nanoparticles; industry; air extraction; reinforcement learning; genetic algorithm

Desarrollo de un algoritmo de inteligencia artificial para optimización de calidad de aire interior y gestión de producción industrial

La modelización de nanopartículas permite realizar simulaciones de concentración en entornos industriales para estimar el efecto de distintas estrategias de extracción forzada de aire en escenarios con procesos que emiten nanopartículas. Muchas técnicas basadas en inteligencia artificial pueden evaluar estas estrategias para encontrar la óptima. Además, pueden minimizar simultáneamente el coste energético del proceso de extracción mediante la coordinación de la actividad industrial con el coste fluctuante horario de la energía de la red. En consecuencia, se proponen dos algoritmos basados en inteligencia artificial: un algoritmo genético y uno basado en aprendizaje por refuerzo. Para el primero, un generador de individuos gestiona las limitaciones físicas del sistema en condiciones reales operativas para que, posteriormente, la población evoluciona con el tiempo imitando los procesos de selección natural, reproducción y mutación. Para el segundo, una política meta-heurística se diseña a partir de un espació de estados y acciones definidas por distintas estrategias heurísticas para explorar potenciales soluciones. Los resultados preliminares evaluando el rendimiento en la minimización del coste energético muestran que ambos algoritmos alcanzan soluciones similares, mostrando la esperada curva de población para el algoritmo genético, pero sin alcanzar una clara curva de aprendizaje en el aprendizaje por refuerzo.

Palabras clave: Energía; nanopartículas; industria; extracción de aire; aprendizaje por refuerzo; algoritmo genético

Correspondencia: Joaquim Cebolla Alemany joaquim.cebolla@upc.edu

Agradecimientos: This research is part of a LIFE-funded project (LIFE20 ENV/ES/000187). It was also supported by the Spanish Ministry of Science and

Innovation (Project CEX2018-000794-S) and by AGAUR (projects 2017 SGR41 and 2021 SGR 00341). Finally, the first author gratefully acknowledges the Universitat Politècnica de Catalunya for the financial support of his predoctoral grant FPU-UPC, with the collaboration of Banco de Santander.

1. Introduction

Industrial production management plays a pivotal role in shaping efficiency, sustainability, and competitiveness of manufacturing operations across various sectors. As industries navigate through an era of rapid technological advancement, globalization, and evolving market dynamics, the landscape of industrial production management is undergoing a significant transformation towards smart automated decision-making algorithms implementation to optimise energy consumption, processing time, energy cost, or use of renewable energy, among others.

One of the prominent trends in industrial production management is the adoption of advanced technologies to enhance productivity, quality, and agility in manufacturing processes. Automation, robotics, and digitalization are revolutionizing traditional production methods, enabling seamless integration of data-driven insights into decision-making processes. Smart factories equipped with Internet of Things sensors, artificial intelligence (AI), and predictive analytics capabilities are driving efficiencies, reducing downtime, and enabling predictive maintenance strategies to optimize asset utilization and minimize operational costs.

Moreover, the rise of Industry 4.0 concepts is reshaping the manufacturing landscape, fostering the convergence of physical and digital systems to create interconnected, intelligent production ecosystems. Industry 4.0 principles emphasize the use of cyber-physical systems, cloud computing, and real-time data analytics to enable autonomous decision-making, adaptive manufacturing processes, and customized production solutions tailored to individual customer needs. As a result, industrial production management is evolving towards more flexible, responsive, and customer-centric approaches, enabling companies to meet the demands of a rapidly changing marketplace.

Furthermore, the globalization of supply chains and increasing market volatility are driving companies to rethink their production strategies and supply chain management practices. The COVID-19 pandemic highlighted the vulnerabilities of global supply chains, prompting organizations to prioritize resilience, agility, and localization in their production networks. As a result, there is a growing emphasis on supply chain digitization, risk mitigation strategies, and the development of robust contingency plans to ensure business continuity and mitigate disruptions.

However, there exist diverse levels of permeability of these technologies to the industry depending on the sector. In this context, special focus should be placed on industrial workshops dedicated to repairing and enhancing mechanical components from other industries such as chemical, pharmaceutical, or energy plants. These workshops exhibit high flexibility in task scheduling to meet their clients' requirements and schedules, resulting in stochastic flexible production with potential optimisation approaches involving energy consumption (Zhao et al., 2022). Several studies have examined different scheduling strategies for workshops (Sang et al., 2021) aimed at minimising tardiness (Luo, 2020; Qiu & Lau, 2013; Xiong et al., 2017). Consequently, industrial workshops offer significant potential for deploying other flexibility approaches due to their high adaptability to client requirements.

Among the various reparation techniques, thermal spraying stands out as a surface engineering process that enables selective repairs and coating with highly efficient raw material usage with the potential for further reuse and recycling (Kuroda et al., 2008; Lashmi et al., 2020; Viswanathan et al., 2021). This process finds applications in industries such as aerospace, automotive, energy, and manufacturing, with techniques including high-velocity oxy-fuel (HVOF) spraying, atmospheric plasma spraying (APS), flame spraying, arc spraying, and detonation gun spraying. According to the Thermal Spraying Coating Market Outlook (Mordor Intelligence, n.d.), the global thermal spray market size was $10.91 billion in 2023, projected to reach $13.41 billion by 2028. Consequently, the number of workers and companies involved in thermal spraying is expected to increase proportionally in the coming years.

Nonetheless, thermal spraying processes emit incidental nanoparticles (INPs) in large quantities, posing potential health hazards (Ajith et al., 2022; Ghosh, 2019; Kreyling et al., 2006; Oberdörster, 2001; Sonwani et al., 2021; Stone et al., 2017). Therefore, implementing risk management measures such as mechanical air extraction is necessary to improve indoor air quality (IAQ) in these environments. Such scenarios extend beyond thermal spraying to include various highly energetic and mechanical processes involved in materials transformation (Biswas & Wu, 2005; Gwinn & Vallyathan, 2006; Hämeri et al., 2009; Wake et al., 2002) in industrial workshops and other types of industries. However, INPs present a legislative gap within the European Union, with only national recommendations, such as the nano reference value in the Netherlands (Hendrikx & van Broekhuizen, 2013).

In this context, AI algorithms emerge as a potential tool for not just energy, cost, or time management, but also for risk management measures policies design. Consequently, a single algorithm could couple both criteria and design optimal scheduling strategies. In this regard, meta-heuristics have already been used in literature to solve complex optimisation problems (Huang et al., 2021; Lin et al., 2022; Ruiz & Stützle, 2007; Yu et al., 2023). These algorithms combine different heuristic strategies to “destroy” and “construct” potential solutions iteratively.

In this paper, we introduce the ongoing development of two AI meta-heuristics to minimise energy costs due to INP extraction in a thermal spraying booth as a function of the fluctuating hourly energy price for a week of work. Moreover, cost minimisation indirectly increases the share of renewable energy consumption due to the descend in the grid energy price while variable renewable energy sources such as solar or wind are generating.

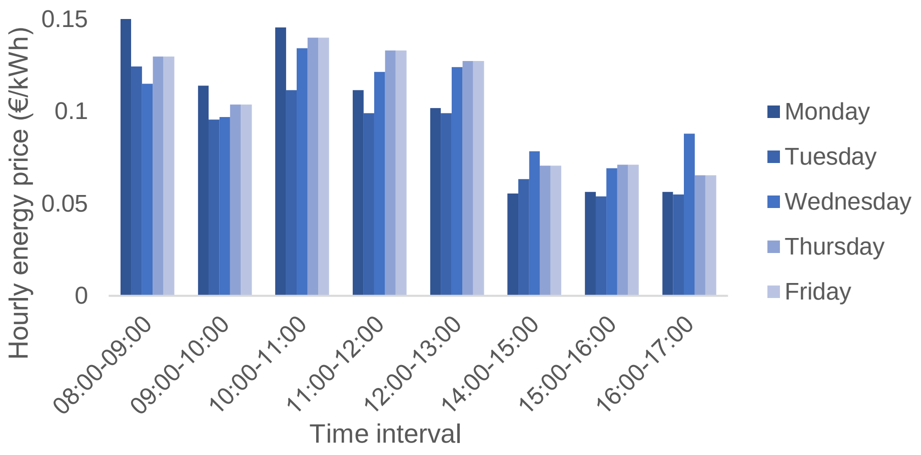
2. Algorithm design

2.1 Case study

The tackled scenario in this paper falls under the category of the classic optimisation single-machine scheduling problem. It involves scheduling a set of jobs to be processed on a single machine, subject to certain constraints and objectives, such as minimizing total completion time or minimising total tardiness. In the single-machine scheduling problem, each job has a processing time and a due date, and the goal is to determine the sequence in which the jobs should be processed on the single machine to optimize the chosen objective function.

For our case study, jobs consist of sets of pieces that have to be processed during a week with processing times between 1 and 4 hours (6 sets of 1 hour, 6 sets of 2 hours, 2 sets of 3 hours, and 1 set of 4 hours). The first development stage of the algorithms focuses on the minimisation of energy cost without considering INP concentration variations. For this purpose, it requires the hourly energy grid prices during working hours, starting at 8:00 and finishing at 17:00. The schedule includes a lunch break between 13:00 and 14:00. As boundary conditions, sets of pieces have to be processed without interruptions and a gap of one hour has to be considered between sets representing the time to prepare and move the sets from and to the booth. The system assumes a constant electric motor power of 22 kW for the extraction system and the hourly energy prices represented in Figure 1.

Figure 1: Hourly energy prices (in €/kWh) during working hours for each day of the week considered for the case study



2.2 Genetic algorithm

A genetic algorithm (GA) is a type of optimization algorithm inspired by the principles of natural selection and genetics. It is used in AI and computational optimization to find solutions to complex optimization and search problems. In a GA, a population of candidate solutions (often called individuals or chromosomes) evolves over successive generations. Each individual represents a potential solution to the optimization problem. The algorithm starts with an initial population of random solutions and iteratively evolves this population over multiple generations.

The simulated phenomena that cause the evolution of the initial population between epochs are selection and recombination, with the former including reproduction and mutation. For selection, the fitness of each individual is evaluated using a function that measures how well the individual solves the optimisation problem. Then, this fitness is used as a probabilistic criterion to define the survival of each individual. Reproduction creates offspring by applying a crossover of existing individuals, combining genetic material from the two parents. Mutation implies a random modification of part of the individual through epochs.

To design the individuals, each set of pieces is represented by an integer representing the hours of processing it requires, and zeros are added to represent inactivity time until reaching the 40 expected hours of work for a week.

To initialise the population, an array containing all the sets of pieces is randomly shuffled as many times as individuals, determining the population size. Then, zeros are added to the array to fulfil the boundary conditions of the problem (1-hour break for lunch, 1-hour break between sets, and 8:00-17:00 schedule without interruptions within a set).

The calculation of the fitness is conducted considering the energy cost of operative hours. Then, weights are assigned to each individual based on their fitness. These weights are used by a random selection algorithm that increases the probability of survival between epochs proportionally to each individual's fitness. This process is applied resulting in a final population smaller than the initial one.

After selection, the remaining individuals are randomly iteratively selected by pairs to generate child individuals with part of both parents. This process is called reproduction or crossover. Parents are split at day shifts or lunch breaks to facilitate the management of the child. For example, if it is randomly decided that the breakpoint of the new individual is Tuesday at lunch break, Monday and Tuesday morning will come from one parent, and Tuesday afternoon and the remaining three days of the week will come from the other parent. If a child does not represent a valid solution because it does not contain the proper sets of pieces, it is modified to fit the boundary conditions of the experiment.

Then, there exists a probability for an individual to randomly change one of its chromosomes. That is, to present a mutation. For the case study, this is applied as a random permutation between two mornings or two afternoons or a shuffle of a morning or an afternoon. These four options are randomly chosen for a random day if mutation activates for an individual. This interpretation of mutation is designed so small changes can be applied to the schedules without the need to repair potential unviable outcomes.

Regarding the properties of the algorithm for its deployment, a population of 1000 individuals is designed to evolve over 100 generations. Moreover, the mutation and elitism rates are both defined at 0.4.

2.3 Reinforcement learning

Reinforcement learning (RL) is a branch of machine learning in which an agent (the entity to control) learns to make sequential decisions by interacting with an environment (the space with which the agent interacts) to maximise a cumulative reward. In this paradigm, the agent acquires knowledge through trial and error, obtaining feedback in the form of rewards or penalties based on its interactions within the environment changing its state. The goal of the agent is to learn a policy (defined through a mapping from states to actions) that optimises the cumulative reward obtained over time. A state describes the characteristics of the environment at a given moment, and its set represents all possible situations in which the agent will have to decide which action to take.

Rewards are the key point of this type of learning and tell the agent whether the action it has previously taken has led it to a "good" or "bad" state, which we define based on our objectives. As more and more actions are taken in different states, the agent will have scores for each possible action to take at each moment and will decide how to behave based on this.

Among RL strategies, Q-learning (QL, where the *Q* stands for *quality*) emerges as a model-free algorithm applied in finite Markov decision processes (i.e., in situations where outcomes are partially stochastic and partly subject to the control of a decision-maker). In QL, the agent learns an action-value function *Q(s, a)*, which represents the expected cumulative reward that the agent will receive by taking action *a* in state *s* and subsequently following the optimal policy. A Q-value is assigned and updated to each possible state-action pair as the agent interacts with the environment, initialising each Q-value at zero. All state-action pairs are grouped in a matrix named Q-table, with as many rows as possible states and a column for each possible action.

The equation to update a Q-value $Q(s_t, a_t)$ at an instant *t* after performing an action is (1), where α is the learning rate, r_{t+1} is the reward associated with the resulting state after performing the action a_t, γ is the discount factor, and *a'* is every possible action in the resulting state.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\right] \tag{1}$$

The two constants of the equation determine the nature of the learning policy getting a value between 0 and 1. α determines the update speed of the Q-table per iteration and usually gets values near to 0. γ evaluates the projection towards the future of the taken action. I.e., the lower the value it has, the higher importance will be given to the immediate effect of the action.

The application of the algorithms works as follows:

1. The script identifies the current state of the system.
2. The algorithm decides which action to take.

3. The agent executes the action.
4. The new state is identified.
5. The result of the action is evaluated.
6. The Q-table is updated.

To avoid bias during the training, the agent starts deciding randomly which action to perform (exploration) and slowly increases the use of the Q-values in the decision-making policy (exploitation). This procedure allows the evaluation of each possible state-action pair and, consequently, avoids missing potential “good” actions.

For the specific case study of this research, possible actions are low-level heuristics. These strategies are:

- Permutation of morning schedules
- Permutation of afternoon schedules
- Permutation of full-day schedules
- Permutation of the activities in a day
- Permutation of the activities in a morning
- Permutation of the activities in an afternoon

To define the states, three options were considered: former activity, number of actions used, and routine energy cost. In the first option, a square Q-table matrix would define an order to apply each low-level heuristic to reach the fastest possible way true optimum of the system. In the second alternative, the Q-table is followed sequentially from top to bottom, given a maximum number of iterations, to shorten the time to obtain the best permutation. Finally, total energy cost defines the possible states. To avoid potential closed loops affecting the performance of the algorithm, the first option is discarded. Moreover, to not restrain the use of the q-table across their axis, the third option is chosen, and thus, states of the system are defined in 45 cost intervals between 54 and 65 €.

The iteration limit for each episode is set at 500 low-level heuristic actions, and an alternative finishing condition is defined in case the solution does not improve after a few actions, showing convergence in the result (that is not stored in case the solution is worse than the schedule prior to the application of the heuristic.

3. Results and discussion

3.1 Genetic algorithm

Applying the GA as described in Section 2.2, the resulting fitness curve representing the evolution over generations of the algorithm population can be seen in Figure 2, reaching a final energy cost of 52.27 € for the week. The initial population has at least a member near the true minimum, although the mean energy cost is far from this value. However, the mean experiences a swift decrease during the first generations, to then experience a stabilisation phase between the 10th and 30th generations, when it descends again almost until the 40th generation, to then slowly converge with the minimum energy cost, that is, the found optimal solution. In other words, all population is formed by the fittest individual, completely overlapping before the 80th generation. Simultaneously, the algorithm is able to generate the optimum after 20 generations, showing a fast decrease in the first 10 iterations. After finding the true optimum, it doesn’t disappear from the population due to mutation or pseudo-stochastic selection. The found optimal schedule is represented in Figure 3.

Figure 2: Population fitness curve evolution over 100 generations for the GA with the mean (in red) and minimum (in blue) weekly energy cost, in €

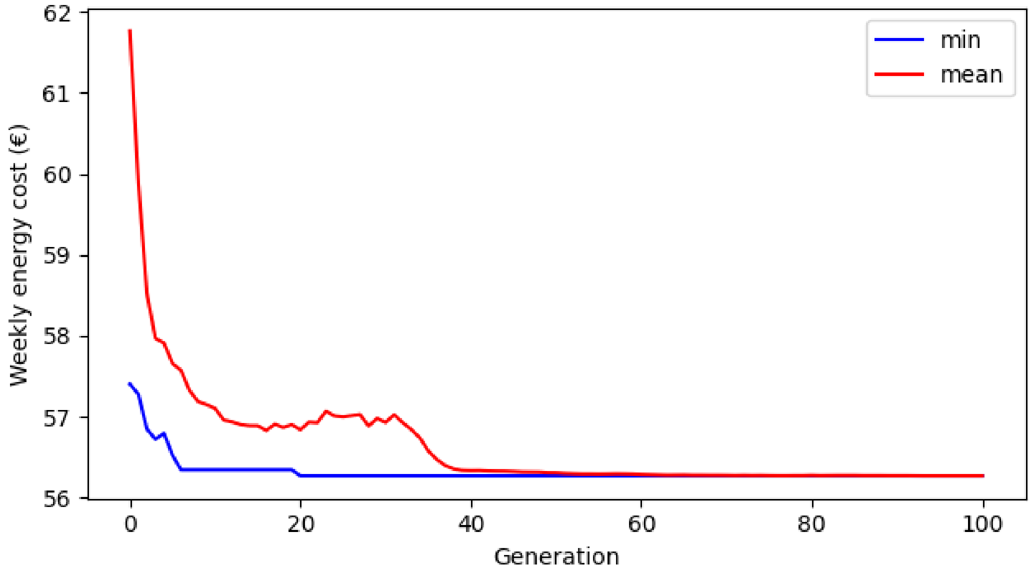


Figure 3: Optimal schedule found through the GA deployment to minimise energy cost for the weekly work demand of the case study

	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-09:00			2	2	2
09:00-10:00		4			
10:00-11:00					
11:00-12:00	2		2		
12:00-13:00				1	1
13:00-14:00	lunch break				
14:00-15:00	3	3	2	1	1
15:00-16:00					
16:00-17:00				1	1

3.2 Q-learning

Applying QL as indicated in Section 2.3, the number of iterations conducted on each episode to reach stabilisation or the maximum number of actions is represented in Figure 4(a), resulting in a final schedule (see Figure 5) with an energy cost of 52.00 € per week. However, observing the convergence value for each episode in Figure 4(b), we can observe that the algorithm does not improve through time, but behaves like a noisy signal, presenting even an apparent first stabilisation phase during the initial 700 episodes, where exploration is prevalent over exploitation.

In this case, the algorithm required 1268.85 seconds (i.e., more than 21 minutes) to calculate the resulting Q-table in Table 1 with the same conditions described in Section 3.1. This table shows how the low-level heuristic 1 is the one improving the most the schedule for most states. Adversely, low-level heuristics 5 and 6 show little effect on the final weekly price. Moreover, states below 56 € are never reached, showing that they are not feasible, at least, through the designed actions.

Figure 4: (a) Required number of loop iterations until solution converges for each episode; and (b) minimum found weekly energy cost for each episode

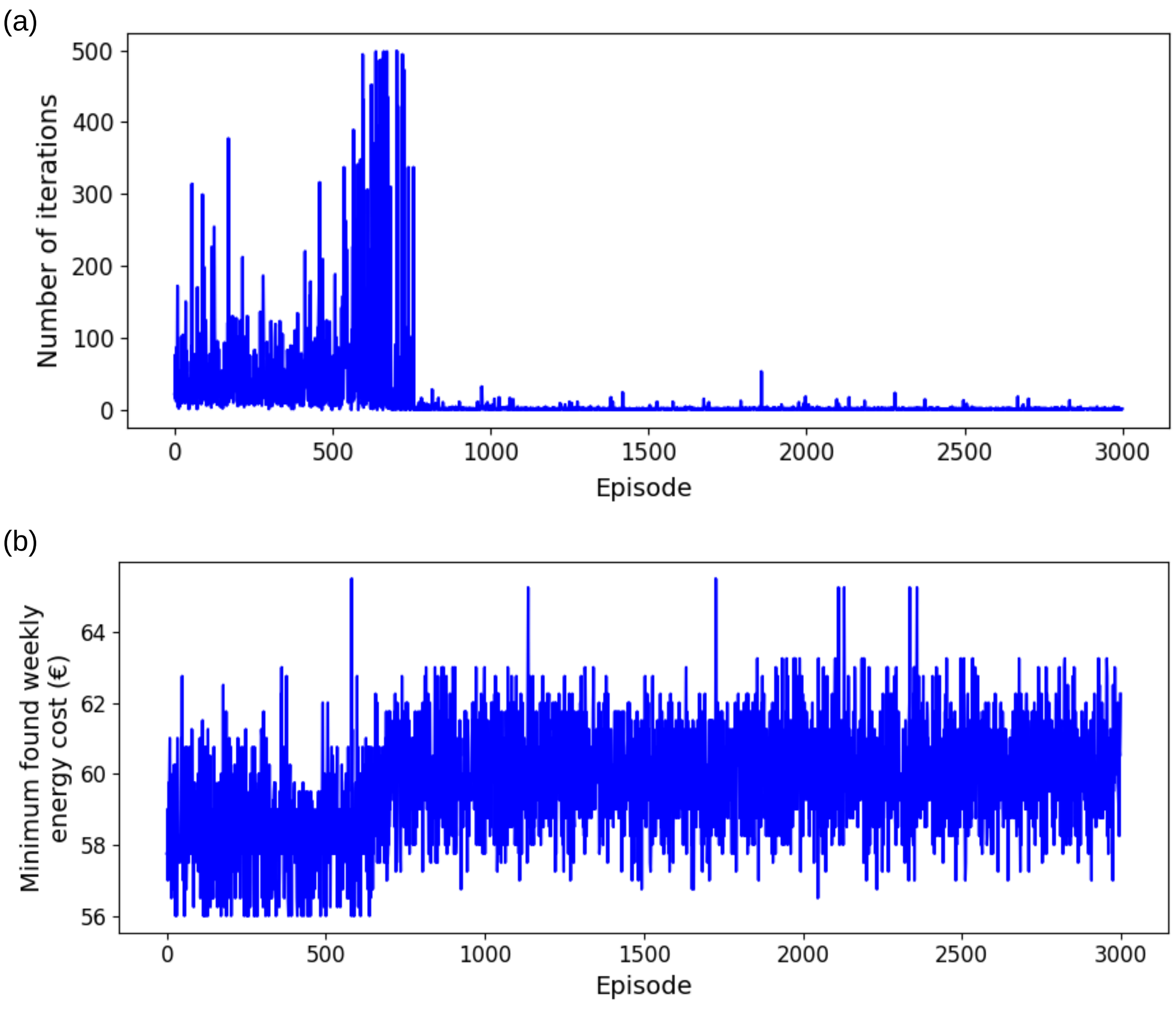
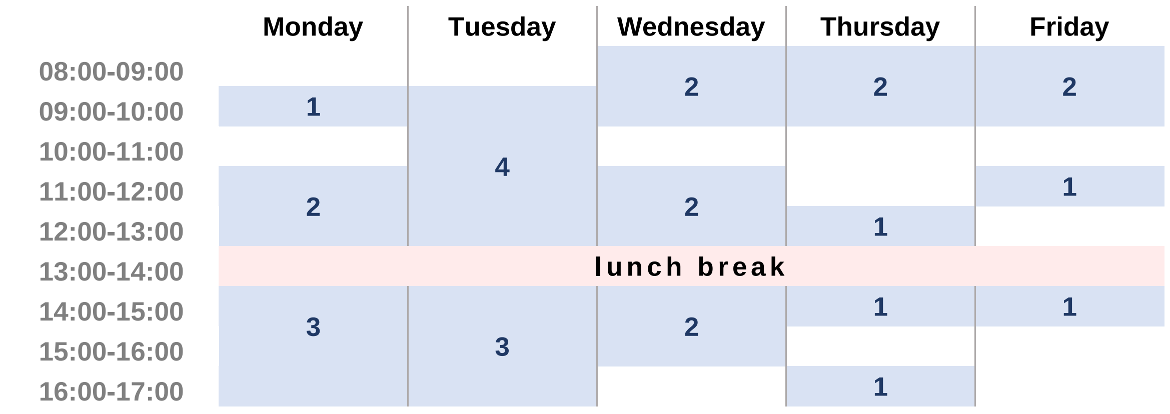


Figure 5: Optimal schedule found through the QL deployment to minimise energy cost for the weekly work demand of the case study



	Monday	Tuesday	Wednesday	Thursday	Friday
08:00-09:00			2	2	2
09:00-10:00	1	4			
10:00-11:00					
11:00-12:00	2		2		1
12:00-13:00				1	
13:00-14:00	lunch break				
14:00-15:00	3	3	2	1	1
15:00-16:00					
16:00-17:00				1	

Table 1: Q-table with price ranges as states (rows) and low-level heuristics as actions (columns) after training with higher Q-values in blue and lower Q-values in red

State	LLH1	LLH2	LLH3	LLH4	LLH5	LLH6
54.00	0	0	0	0	0	0
54.25	0	0	0	0	0	0
54.50	0	0	0	0	0	0
54.75	0	0	0	0	0	0
55.00	0	0	0	0	0	0
55.25	0	0	0	0	0	0
55.50	0	0	0	0	0	0
55.75	0	0	0	0	0	0
56.00	0.01	0.01	0.01	0.01	-0.02	0.01
56.25	1.93	0.01	0.01	0.01	-0.11	0.01
56.50	0.01	0.01	0.01	0.01	0	0.01
56.75	0.01	0.01	0.01	0.01	0.01	0.01
57.00	0.01	0.01	0.01	0.01	0.01	0.01
57.25	0.61	0.01	0.01	0.01	0.01	0.01
57.50	0.01	0.01	0.01	0.01	-0.06	0.01
57.75	2.55	0.01	0.01	0.01	-0.02	0.01
58.00	1.38	0.01	0.01	0.01	0.01	0.01
58.25	0.01	0.01	0.01	0.01	0.01	0.01
58.50	3.76	0.01	0.01	0.01	-0.04	0.01
58.75	2.02	0.01	0.01	0.01	0.01	0.01
59.00	0.01	0.01	0.01	0.01	-0.13	0.01
59.25	2.29	0.01	0.01	0.01	0.01	0.01
59.50	3.17	0.01	0.01	0.01	0.01	0.01
59.75	1.07	0.01	0.01	0.01	-0.37	0.01
60.00	1.26	0.01	0.01	0.01	0.01	0.01
60.25	2.05	0.01	0.01	0.01	0.01	0.01
60.50	0.01	0.01	0.01	0.01	0.01	0.01
60.75	1.26	0.01	0.01	0.01	0.01	0.01
61.00	0.27	0.01	0.01	0.01	0.01	0.01
61.25	0.01	0.01	0.01	0.01	0.01	0.01
61.50	6.89	0.01	0.01	0.01	0.01	0.01
61.75	0.01	0.01	0.01	0.01	0.01	0.01
62.00	4.1	0.01	0.01	0.01	0.01	0.01
62.25	0.01	0.01	0.01	0.01	0.01	0.01
62.50	0.01	0.01	0.01	0.01	0.01	0.01
62.75	5.66	0.01	0.01	0.01	0.01	0.01
63.00	2.54	0.01	0.01	0.01	0.01	0.01
63.25	9.53	0.01	0.01	0.01	0.01	0.01
63.50	6.43	0.47	0.51	0.43	0.42	0.48
63.75	0.16	6.98	0.14	0.14	0.18	0.13
64.00	2.65	2.57	8.03	1.46	1.31	1.43
64.25	5.75	0	9.73	0	1.67	2.37
64.50	4.29	10.05	3.15	4.75	4.53	0
64.75	10.75	0	0	0	0	0
65.00	7.85	0	0	0	0	0

3.3 Discussion

Comparing both algorithms, QL can reach a better solution than the GA. Monday and Friday slightly diverge, while the other three days of the week present the same order of piece processing. However, both algorithms seem to have potential for improvement, so an even better solution could be found with modifications to the algorithm.

Despite its performance, QL does not show a clear learning curve, which can help us conclude that it finds a better schedule than the GA because of the number of attempts or the quality of the low-level heuristics deployed. This last factor can be affected as well for the codification of the schedules within the algorithm and the management of the boundary conditions, which can result in missing better schedules.

Regarding calculation time, the QL algorithm takes more than 40 times more to finish than the GA. This phenomenon can be explained mainly by the difference in learning performance between algorithms. GA's convergence allows it to finish swiftly, contrasting with the incapability of QL to consistently find better schedules through iterations, thereby resulting in entrapment within a loop until activation of convergence criteria due to the absence of evident enhancements.

Observing the Q-table, its information can be useful to assess the performance of the low-level heuristics and, consequently, think about deploying new actions. Additionally, the states' definition seems to be the main cause of the malfunctioning of the algorithm, since most of the

Q-values remain near zero and it is not possible to identify a proper policy to generate good quality schedules consistently.

4. Conclusions and future perspectives

Two single-machine scheduling AI algorithms have been designed and compared in this work to study incidental nanoparticle generation industrial scenarios to minimise the energy cost of mechanical air extraction systems with real operative boundary conditions for a hypothetic weekly workload defined by different sets of pieces with diverse processing times. The design algorithms consist of a GA and RL, specifically QL.

Regarding the GA, the boundary conditions of the case study were used to determine the structure of the individuals of the population and to deploy the crossover and mutation phenomena without obtaining non-viable solutions. Regarding RL, the algorithm was based on a space defined by total cost ranges and an agent affecting the system through six different low-level heuristics.

For each of the algorithms, schedules' structure and low-level heuristics were designed, as well as the necessary information to evaluate their performance. While the GA showed an evident learning curve, QL was able to obtain a lower final weekly energy cost schedule based on the use of six different low-level heuristics despite not presenting a clear improvement through learning episodes.

To improve the GA, different mutation and elitism rates could be studied. Moreover, as previously mentioned, an alternative individual's design could be explored to facilitate the deployment of new low-level heuristics to reach better solutions like in the QL case.

To solve the learning issue of QL, discarded options when considering the design of the algorithm could be deployed and compared, or a new magnitude to define the states of the system could be proposed in future work.

Finally, the aim of the research that frames the development of the presented work is, as previously stated, energy optimisation in incidental nanoparticle generation scenarios. Hence, the scheduling algorithm will require the incorporation of nanoparticle concentration simulations, not affecting the codification of schedules, but adding a new level of complexity in their evaluation and simulation.

5. References

Ajith, S. A., Mohamed, O., Sabouni, R., Husseini, G., Karami, A., & Bai, R. G. (2022). Toxicological impact of nanoparticles on human health: A review. *Materials Express*, *12*(3), 389–411. https://doi.org/10.1166/mex.2022.2161

Biswas, P., & Wu, C.-Y. (2005). Nanoparticles and the Environment. *Journal of the Air & Waste Management Association*, *55*(6), 708–746. https://doi.org/10.1080/10473289.2005.10464656

Ghosh, S. (2019). *Nanomaterials Safety: Toxicity And Health Hazards*. De Gruyter. https://books.google.es/books?hl=ca&lr=&id=HBiBDwAAQBAJ&oi=fnd&pg=PR7&dq=Nanoparticles+and+occupational+health+hazards+ghosh&ots=p_Lbd58nFf&sig=myNPjb3bZHBRL7VEy-DlxoW34FU#v=onepage&q=Nanoparticles and occupational health hazards ghosh&f=false

Gwinn, M. R., & Vallyathan, V. (2006). Nanoparticles: Health effects - Pros and cons. *Environmental Health Perspectives*, *114*(12), 1818–1825. https://doi.org/10.1289/ehp.8871

Hämeri, K., Lhde, T., Hussein, T., Koivisto, J., & Savolainen, K. (2009). Facing the key workplace challenge: Assessing and preventing exposure to nanoparticles at source. *Inhalation Toxicology*, *21*(SUPPL. 1), 17–24. https://doi.org/10.1080/08958370902942525

Hendrikx, B., & van Broekhuizen, P. (2013). Nano reference values in the Netherlands. *Gefahrstoffe Reinhaltung Der Luft*, *73*(10), 407–414. isi:000326913600002

Huang, Y. Y., Pan, Q. K., Huang, J. P., Suganthan, P. N., & Gao, L. (2021). An improved iterated greedy algorithm for the distributed assembly permutation flowshop scheduling problem. *Computers and Industrial Engineering*, *152*(December 2020), 107021. https://doi.org/10.1016/j.cie.2020.107021

Kreyling, W. G., Semmler-Behnke, M., & Möller, W. (2006). Health implications of nanoparticles. *Journal of Nanoparticle Research*, *8*(5), 543–562. https://doi.org/10.1007/s11051-005-9068-z

Kuroda, S., Kawakita, J., Watanabe, M., & Katanoda, H. (2008). Warm spraying - A novel coating process based on high-velocity impact of solid particles. *Science and Technology of Advanced Materials*, *9*(3). https://doi.org/10.1088/1468-6996/9/3/033002

Lashmi, P. G., Ananthapadmanabhan, P. V., Unnikrishnan, G., & Aruna, S. T. (2020). Present status and future prospects of plasma sprayed multilayered thermal barrier coating systems. *Journal of the European Ceramic Society*, *40*(8), 2731–2745. https://doi.org/10.1016/j.jeurceramsoc.2020.03.016

Lin, J., Li, Y. Y., & Song, H. B. (2022). Semiconductor final testing scheduling using Q-learning based hyper-heuristic. *Expert Systems with Applications*, *187*(September 2021), 115978. https://doi.org/10.1016/j.eswa.2021.115978

Luo, S. (2020). Dynamic scheduling for flexible job shop with new job insertions by deep reinforcement learning. *Applied Soft Computing Journal*, *91*, 106208. https://doi.org/10.1016/j.asoc.2020.106208

Mordor Intelligence. (n.d.). *Thermal Spray Market Size & Share Analysis - Growth Trends & Forecasts (2023-2028)*. Retrieved December 6, 2023, from https://www.mordorintelligence.com/industry-reports/thermal-spray-market

Oberdörster, G. (2001). Pulmonary effects of inhaled ultrafine particles. *International Archives of Occupational and Environmental Health*, *74*(1), 1–8. https://doi.org/10.1007/s004200000185

Qiu, X., & Lau, H. Y. K. (2013). An AIS-based hybrid algorithm with PDRs for multi-objective dynamic online job shop scheduling problem. *Applied Soft Computing Journal*, *13*(3), 1340–1351. https://doi.org/10.1016/j.asoc.2012.07.033

Ruiz, R., & Stützle, T. (2007). A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. *European Journal of Operational Research*, *177*(3), 2033–2049. https://doi.org/10.1016/j.ejor.2005.12.009

Sang, Y., Tan, J., & Liu, W. (2021). A new many-objective green dynamic scheduling disruption management approach for machining workshop based on green manufacturing. *Journal of Cleaner Production*, *297*, 126489. https://doi.org/10.1016/j.jclepro.2021.126489

Sonwani, S., Madaan, S., Arora, J., Suryanarayan, S., Rangra, D., Mongia, N., Vats, T., & Saxena, P. (2021). Inhalation Exposure to Atmospheric Nanoparticles and Its Associated Impacts on Human Health: A Review. *Frontiers in Sustainable Cities*, *3*(August), 1–20. https://doi.org/10.3389/frsc.2021.690444

Stone, V., Miller, M. R., Clift, M. J. D., Elder, A., Mills, N. L., Møller, P., Schins, R. P. F., Vogel, U., Kreyling, W. G., Jensen, K. A., Kuhlbusch, T. A. J., Schwarze, P. E., Hoet, P., Pietroiusti, A., de Vizcaya-Ruiz, A., Baeza-Squiban, A., Teixeira, J. P., Tran, C. L., & Cassee, F. R. (2017). Nanomaterials versus ambient ultrafine particles: An opportunity to exchange toxicology knowledge. *Environmental Health Perspectives*, *125*(10), 1–17. https://doi.org/10.1289/EHP424

Viswanathan, V., Katiyar, N. K., Goel, G., Matthews, A., & Goel, S. (2021). Role of thermal spray in combating climate change. *Emergent Materials*, *4*(6), 1515–1529. https://doi.org/10.1007/s42247-021-00307-1

Wake, D., Mark, D., & Northage, C. (2002). Ultrafine aerosols in the workplace. *Annals of Occupational Hygiene*, *46*, 235–238. https://doi.org/10.1093/annhyg/46.suppl-1.235

Xiong, H., Fan, H., Jiang, G., & Li, G. (2017). A simulation-based study of dispatching rules in a dynamic job shop scheduling problem with batch release and extended technical precedence constraints. *European Journal of Operational Research*, *257*(1), 13–24. https://doi.org/10.1016/j.ejor.2016.07.030

Yu, H., Gao, K. Z., Ma, Z. F., & Pan, Y. X. (2023). Improved meta-heuristics with Q-learning for solving distributed assembly permutation flowshop scheduling problems. *Swarm and Evolutionary Computation*, *80*(April), 101335. https://doi.org/10.1016/j.swevo.2023.101335

Zhao, N., Fu, Z., Sun, Y., Pu, X., & Luo, L. (2022). Digital-twin driven energy-efficient multi-crane scheduling and crane number selection in workshops. *Journal of Cleaner Production*, *336*(December 2021), 130175. https://doi.org/10.1016/j.jclepro.2021.130175

Communication aligned with the Sustainable Development Goals