Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 955–960

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Conceptual framework of scheduling applying discrete event simulation as an environment for deep reinforcement learning

Kristina Eriksson[a,*], Sudha Ramasamy[a], Xiaoxiao Zhang[a], Zhiping Wang[b], and Fredrik Danielsson[a]

[a]*Division of Production Systems, Department of Engineering Science, University West, Trollhättan, SE-461 86, Sweden*
[b]*Research and Technology Development, Volvo Group Trucks Operations, Gothenburg, SE-405 08, Sweden*

* Corresponding author. Tel.: +46721600145; *E-mail address:* kristina.eriksson@hv.se

Abstract

Increased environmental awareness is driving the manufacturing industry towards novel ways of energy reduction to become sustainable yet stay competitive. Climate and environmental challenges put high priority on incorporating aspects of sustainability into both strategic and operational levels, such as production scheduling, in the manufacturing industry. Considering energy as a parameter when planning makes an already existing highly complex task of production scheduling even more multifaceted. The focus in this study is on inverse scheduling, defined as the problem of finding the number of jobs and duration times to meet a fixed input capacity. The purpose of this study was to investigate how scheduling can be formulated, within the environment of discrete event simulation coupled with reinforcement learning, to meet production demands while simultaneously minimize makespan and reduce energy. The study applied the method of modeling a production robot cell with its uncertainties, using discrete event simulation combined with deep reinforcement learning and trained agents. The researched scheduling approach derived solutions that take into consideration the performance measures of energy use. The method was applied and tested in a simulation environment with data from a real robot production cell. The study revealed opportunities for novel approaches of studying and reducing energy in the manufacturing industry. Results demonstrated a move towards a more holistic approach for production scheduling, which includes energy usage, that can aid decision-making and facilitate increased sustainability in production. We propose a conceptual framework for scheduling for minimizing energy use applying discrete event simulation as an environment for deep reinforcement learning.

Keywords: Reinforcement learning; Discrete event simulation; Energy optimal scheduling; Inverse scheduling; Industry 4.0

1. Introduction

Environmental awareness is driving manufacturing companies towards novel ways of reducing energy [1,2]. With challenges in climate and the environment, it is of the highest priority to incorporate sustainability into strategic and operational work in manufacturing industry [3]. Considering energy when planning shop floor production makes the already existing highly complex task of scheduling even more multifaceted [4]. Thus, there is a need to find methods for more sustainable production planning and control as the traditional scheduling objectives i.e., meeting due dates and satisfying customers, must be extended to minimizing energy use and greenhouse gas emissions [5]. The way a production schedule meets its objectives is evaluated through performance measurements [6]. While makespan or maximum completion time i.e., the time at which the last task of a schedule is finished, is a major performance measure to be minimized, other indices like throughput, stability, robustness, and energy use may also be tried to be optimized in parallel with e.g., makespan. Thus, shaping scheduling as a multi-objective optimization problem which is not guaranteed to have a feasible solution. Moreover,

10.1016/j.procir.2022.05.091

some of the performance measures may move in the opposite direction of each other, i.e., optimization of one may require actions that recede others. For instance, to increase stability, idle or slack times are typically inserted into schedules, which in turn reduces energy efficiency of the production [7]. Further, production is affected by disturbances e.g., variations in processing times, machine breakdowns, or fluctuations in demands, meaning scheduling has a stochastic nature [8]. There are several operations research (OR) approaches to solve scheduling problems, e.g., linear programming [6]. However, for realistically sized scheduling problems with multiple performance measures and uncertainties, other methods may be applied. E.g., methods within the range of artificial intelligence (AI) and machine learning (ML) such as reinforcement learning (RL) can be used in scheduling. RL has its roots in dynamic programming and for RL data is not provided in advance, rather machines collect it through interaction [9]. One of the main contributions of RL is that it can solve stochastic decision-making problems even if the way the environment changes is unknown [9]. It is therefore useful to create a simulation environment and to train the RL agents inside that environment. A common tool for addressing improvements of production flows is discrete event simulation (DES), which can be extended to consider and measure energy [10].

The aim of this study is to propose a conceptual framework, for production scheduling that incorporates uncertainties, applying discrete event simulation as an environment combined with deep reinforcement learning and trained agents, focusing the performance measures energy use and makespan.

In the following sections related work, concerning the possibilities of shop floor scheduling applying DES and RL coupled with measuring energy usage, is addressed. Subsequently, the development of the conceptual framework is described followed by results, discussion, and conclusion.

2. Related work

This section describes related research in the areas of RL and DES in relation to production scheduling. Next, incorporation of energy use as a performance measure is addressed.

2.1. RL and DES for production scheduling

Production planning and control (PPC) and job shop scheduling (JSS) in particular, are considered difficult and often are so called NP-hard problems [11]. Thus, approaches using RL have been applied, for example a multi-agent RL framework is presented to solve scheduling [12]. Further, DES can be used as a simulation environment for RL for training the RL agents. An example is a DES framework developed in Python to simulate an autonomous cart fleet, where the model includes stochastic parameters such as passengers and cart behavior [13]. Another example is the creation of a stochastic DES model in Matlab and its use as an environment to interact with an agent defined in Matlab [14]. Both examples show that DES enhances the environment as it is easy to tune parameters and test courses of events.

2.2. Production scheduling and energy use

Sustainability is increasingly being focused on many aspects and areas in manufacturing today. Though, modelling energy can be challenging, as the nature of energy should be defined, then accordingly a measurement can be done. How to integrate this measurement into a simulation is the next challenge. An overview of different types of simulation (continuous, discrete) is presented and the advantages and limitations of each approach is outlined in relation to modelling and energy measurement [15]. There are examples of how to optimize robotic path planning while also measure energy use [16,17,18,19]. In addition to the need for energy optimal robot trajectories, the significance of having energy optimal production schedules has been increasingly studied in OR [20]. One example is a simulation-based energy scheduling of robot stations with a job shop setup where it is possible to identify slack and to adjust the robot operation times to reduce energy adjusting the schedule while keeping the same total cycle time [16]. Further, data requirements for DES of energy use in productions systems has been investigated by [10]. Another example of applying DES when studying energy use is focused on scheduling of a gantry crane in a car engine factory [21]. A DES of energy consumptions for a manufacturing production line has been investigated by [22]. A model applying genetic algorithm to minimize energy use and enhance schedule efficiency has been developed for flexible manufacturing systems [23]. Energy-efficient rescheduling in job shop problems has been studied by [24] where a memetic algorithm is proposed for finding a schedule that minimizes energy use. Game theory is applied to consider the environmental impact of real-time multi-objective flexible job shop scheduling [25].

As exemplified PPC is complex and difficult to solve optimally and many approaches have been studied, often with single performance measures of e.g., meeting due dates or minimizing makespan. The complexities and uncertainties of PPC mean challenges of development of RL as a tool for multi-objective scheduling. Further, though there are examples of studying environmental factors in relation to scheduling, the performance measure of energy use needs attention [23]. It is imperative for manufacturing industry to address energy in many areas and adding energy use as a parameter to minimize, makes scheduling harder. Thus, this area of research needs further attention for increased sustainability.

3. The development of the conceptual framework

While scheduling is extensively addressed by OR and heuristic methods, both paradigms face challenges when the size of the problem grows, and uncertainties are present. Moreover, they fail as real time controllers at the presence of production disturbances. Here we explain the proposed approach of developing a conceptual framework for studying a JJS problems with the performance measure of energy use. The framework combines RL and agents with a DES environment.

3.1. The Case and DES as a simulation environment for RL

As complexity of a system to be optimized increases, a simulation model can be desired for interacting with the RL agent. Simulation results could then be analyzed to convert actions into inputs of the simulation and observations as measurements taken from the simulation. DES allows to easily set and obtain values from a model, as it is based on events occurring in the system [26], and it makes it possible to conduct experiments that cannot be performed on the real systems. It also allows for the possibility to model random variations present in most manufacturing systems. In this work Tecnomatix Plant Simulation DES software was applied.

The DES model is built with data from a real case of a robotic production cell in automotive industry. The case encompasses a production cell consisting of three robots and seven processes i.e., four operative processes, Spot Welding I, Gluing, Hemming and Spot Welding II, and three transport processes. Further the model includes three buffers, one for modelling the infeed, one for modelling the outfeed of the robotic cell, and one buffer to model the main line. The concept proposed is to create a DES model that can include parameters such as failure rates and varying processing times, coupled with RL. The purpose of this is to design and implement a proof of concept using DES as a simulation environment for RL, which in the next step can be updated with data incorporating uncertainties from a real production environment.

3.2. Energy measurement and signature

Energy signatures of robotics operations are extracted following the approach introduced in [27]. The method suggests conducting a series of experiments and measure energy use levels when the same operation is repeated with different speeds. The energy profile, parameterized by operation time, is then approximated by a piecewise polynomial function satisfying a set of conditions extracted from the empirical data. In this study, data were collected using a commercial simulation software, ABB RobotStudio, where an operation is repeatedly simulated with different speeds. A fourth order polynomial function is used as an approximator to parametrize the performance curve with nominal operation time and nominal energy. The energy profile is discretized into four regions: fastest, fast, optimal, and slow. The discretized form is used to introduce the notions of operation code and schedule code. The schedule code is created based on the logic i.e., the number of jobs is added as the first bit to the array, and resulted number is interpreted in base 10. Assume that the number of jobs are 8, the schedule code can be explained as following:

$$\text{scheduleCode} = [\text{nJobs}, \text{opBit}_{O_{W1}}, \text{opBit}_{O_G}, \text{opBit}_{O_H}, \text{opBit}_{O_{W2}}]$$

$$\text{scheduleCode} = [8,1,2,0,3] = (81203)_{10}$$

These codes become useful for representing capacity curves of production processes. Moreover, RL agents working with discrete state or action spaces can be trained using discretized form of the energy signature. To discretize the energy profile, we select five discretization points resulting in four intervals. One may increase the resolution by introducing more points, but four regions were enough for the purpose of analysis in this study. In Fig. 1 can be seen that also an operation bit is assigned

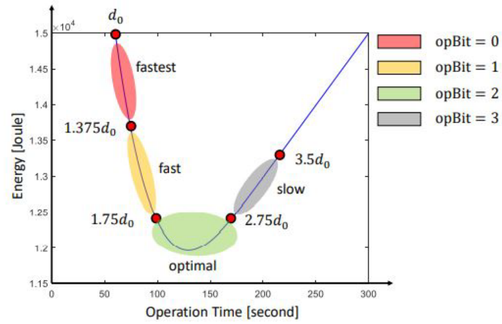

Fig. 1. Discretized energy signature

to each operation, starting from 0 for the first interval. The operation bits are assigned depending on different the operations, for example Spot Welding I, Gluing, Hemming and Spot Welding II.

3.3. Inverse scheduling

The desire to meet production rate while minimizing energy is formulated as an optimization problem according to inverse scheduling. Meaning a reverse scheduling problem where an input demand is specified as a desired capacity. The capacity is defined as number of jobs divided by the makespan. Thus, a combination of number of jobs and process duration times should be found to meet that. The capacity threshold ϵ is defined as the difference between capacity demand k and schedule capacity c. The schedule capacity c is obtained by inverse scheduler. For a given capacity demand and capacity threshold pair (k,ϵ), we define the set of all schedules realizing k bounded by ϵ as the solution set, and denoted by $\mathcal{S}_{k,\epsilon}$, see Fig. 2.

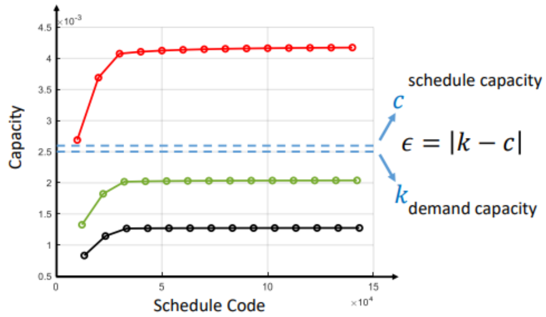

Fig. 2. Solution set associated to a capacity demand, capacity threshold pair.

Every schedule within $\mathcal{S}_{k,\epsilon}$ has the same capacity, but a different combination of number of jobs n and duration times D. Thus, by changing n and D will result in different solutions sets i.e., $\mathcal{S}_{k,\epsilon} = \mathcal{S}_{k,\epsilon}(n,D)$. Therefore, we need to find the best combination of number of jobs and duration times (n_*,D_*), in such a way that energy use is minimized over the elements of $\mathcal{S}_{k,\epsilon}$ (n,D), $(n_*,D_*) = argmin\, \mathcal{S}_{k,\epsilon}(n,D)$. Every schedule within the solution set proposed by the inverse scheduler should minimize the makespan for its selected number of jobs and duration times. The different combination of function value $\mathcal{S}_{k,\epsilon} = \mathcal{S}_{k,\epsilon}(n,D)$ is plotted with red, green and black color response in Fig.2 to find the solution set.

3.4. RL, Agent, and Artificial Neural Network

In RL an agent takes actions and interacts with an environment to maximize the reward [9]. Therefore, there are two main components in a RL problem, agent, and environment. The agent's goal is to find the optimal scheduling to the problem. Hence, the agent considers policy mapping observations received from the environment (state of the system) into actions to be applied on the environment, and a RL algorithm improves the policy considering the reward received from actions taken. Further, we combine reinforcement learning with Artificial Neural Networks (ANNs) as function approximators. This is called deep reinforcement learning. The RL agent has two major components, the policy, and the learning algorithm. The policy is implemented as a neural network function approximator using Matlab, which sends the desired schedule [n,D] to the scheduler in the Matlab environment, which is connected to the DES model of the Plant Simulation software. The agent is trained for PPC, where the trained RL agent is used to create tools for scheduling (production planning) and rescheduling (production control). The two applications, namely the RL agent and the operation code scheduler, are created with the help of Matlab App Designer. The inverse scheduler is interpreted as a decision maker, and the proposed training is a learning process founded by trial and error, which is based on random numbers, one class of learning methods for RL. For the inverse scheduling problem, since the state space is decided to be continuous, it is no longer feasible to model the policy as a simple look-up table. For this purpose, we have employed neural networks as function approximators. See Fig. 3 for an overview of the proposed RL framework for inverse scheduling.

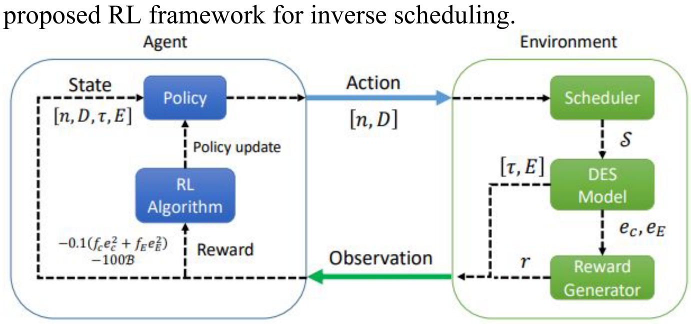

Fig. 3. Proposed RL framework for inverse scheduling.

The agent was trained employing Deep Q-Network (DQN) algorithm. The DQN algorithm is a critic-based RL method. It has a value function Q that estimates the long-term reward $\nu \in \mathbb{R}$ the agent could collect by applying actions $A \in \mathbb{S}_A$ while being in state $S \in \mathbb{S}_S$. Like other value-based algorithms, the policy is implicit in DQN. Assuming that the critic is parameterized by $\overline{\theta_Q}$, i.e. $\nu\ Q(S,A;\overline{\theta_Q})$ DQN algorithm adjusts elements of $\overline{\theta_Q}$ in such a way that the agent can collect maximum possible reward. While training, DQN agent explores the action space using epsilon greedy approach. This allows the agent to make a balance between exploitation and exploration. Further, DQN suggests having a target critic $Q'(S,A;\overline{\theta_{Q'}})$ with the same structure and parametrization as Q. It also collects past experiences of the agent in an experience buffer and sample from that to tune $\overline{\theta_Q}$. In depth explanation of the algorithm can be found in [28]. The environment has two major objectives. The first one is to update its state as a response to received actions, and to report the new state back to the agent. The second one is to generate the reward signal in each iteration of training. The reward generator inside the environment is responsible for that.

3.5. Overview of the conceptual framework

The goal is to with the same capacity find a schedule that minimizes energy use, but increasing the number of jobs and increasing the makespan. To do that, the agent is allowed to select different number of jobs (within a defined range) and different durations for the four main processes (machines). Transport operations have fixed process times. These different number of jobs and durations will lead to different schedules. A scheduler in the RL environment finds the schedule that minimizes the energy use for a fixed capacity, given the number of jobs and process times, and sends it to the DES model. The energy was discretized so that every process had four different possible values: fastest, fast, optimal and slow. The RL algorithm presented in the agent can choose between these four values (operations codes) for every process, which results in different energy use. Knowing the energy profile for each operative process, it can be integrated into the DES model, so that every time an operative process is completed, the energy is computed. Every schedule results a value of the energy use, which can be measured from the simulation model. The agent is given a reward based on the error in the production rate i.e., how far the scheduled production rate is from the demanded production rate, and on the error in the energy use i.e., how far the energy use is from the optimal energy use, assumed to be known by the energy signatures. See Fig. 4 for an overview of the conceptual framework.

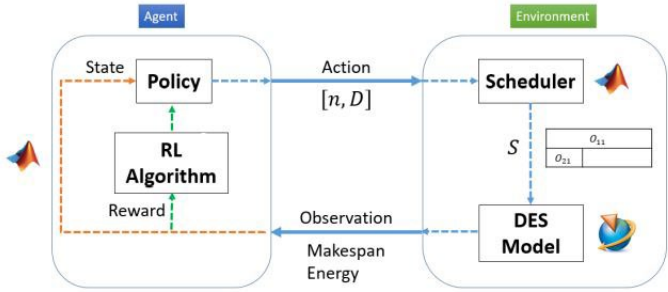

Fig. 4. Overview of the conceptual framework.

4. Results and analysis

The simulation experiment setup applied data from a real scenario from a real case in the automotive industry. Experiments were run in the simulation environment, according to the conceptual framework setup. Tests were run to compare the current schedule of the cell, its makespan and energy use, to the improved schedule found by the agent to fulfil the same demand production rate. Given a production rate demand, the trained agent will get the schedule that meets the demand, as close as possible, and with close to minimal energy use. The current schedule of the cell, composed by seven machines, and assumed to be at the highest possible speed for every operation is given in Fig. 5. The makespan for one cycle is 736 seconds, three jobs are produced every cycle, and the energy use per job produced is 100.99 kJ. Knowing the number of jobs and makespan of the schedule, the production rate can be obtained according to:

$$production\ rate = number\ of\ jobs/makespan = 3/736 = 40{,}76e-4$$

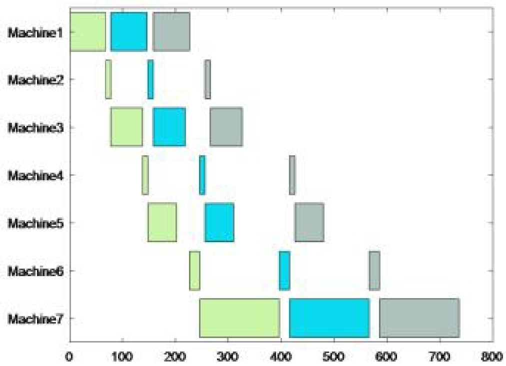

Fig. 5. Original job shop schedule where the x-axis shows time in seconds, and the y-axis is number of the machines.

This production rate can thereafter be used as input for the trained agent, and a schedule meeting that production rate with close to minimal energy use will be found. This new schedule is presented in Fig. 6 and see Table 1 for an overview of the experimental results. The makespan for that schedule per cycle is 1748 seconds, and the number of jobs produced is seven. Calculating the production rate according to:

$$production\ rate = 7/1748 = 40{,}04e - 4$$

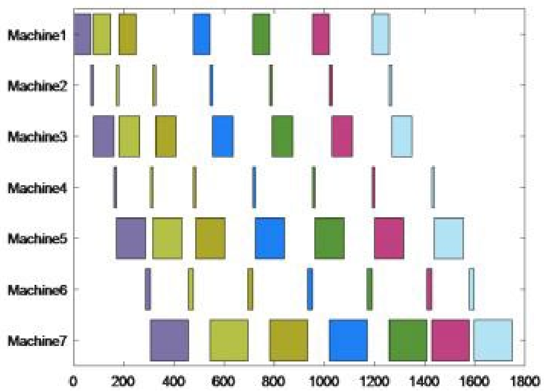

Fig. 6. Schedule found by the trained RL Agent, where the x-axis shows time in seconds, and the y-axis is number of the machines.

Table 1. Comparison between current schedule and the one by RL Agent.

Performance Index	Current schedule at factory	Best schedule found by the RL Agent
Number of jobs	$n = 3$	$n = 7$
Makespan	$\tau = 736$ sec	$\tau = 1748$ sec
Capacity	$c = 0.00407\left(\frac{1}{\mathrm{sec}}\right)$	$c = 0.00400\left(\frac{1}{\mathrm{sec}}\right)$
Operation Times	$\overline{D} = \begin{bmatrix} d_{W1} \\ d_G \\ d_H \\ d_{W2} \end{bmatrix} = \begin{bmatrix} 68\ (\mathrm{sec}) \\ 60\ (\mathrm{sec}) \\ 54\ (\mathrm{sec}) \\ 150\ (\mathrm{sec}) \end{bmatrix}$	$\overline{D} = \begin{bmatrix} d_{W1} \\ d_G \\ d_H \\ d_{W2} \end{bmatrix} = \begin{bmatrix} 68\ (\mathrm{sec}) \\ 82\ (\mathrm{sec}) \\ 116\ (\mathrm{sec}) \\ 150\ (\mathrm{sec}) \end{bmatrix}$
Operation Code	opCode = [0,0,0,0]	opCode = [0,1,2,0]
Energy	$E = 100.99\left(\frac{\mathrm{kJ}}{\mathrm{job}}\right)$	$E = 94.31\left(\frac{\mathrm{kJ}}{\mathrm{job}}\right)$

Looking at the values, we notice that the production rate is not the same, thus we calculate the absolute error accordingly:

$$error = |demand - current| = |0.0040761 - 0.0040046| = 0.715e - 4$$

The error is very small, converted into jobs/hour it is equivalent to 0.26 jobs/hour. Therefore, it can be accepted as a valid solution. Studying the energy use for this schedule we find that it is 94.3 kJ per unit produced. Compared to the initial schedule this corresponds to a reduction of 6.6 percent. It is observed that the agent managed to save energy by 6.6 percent through slowing down the second and third operations. It is demonstrated that it was possible for the agent to find a schedule that produces at the same rate but with lower energy use. After gathering results, the agent optimized schedule for 1073 different given demands, Fig. 7. shows the energy use for each possible demand computed by the agent. The demand is expressed in parts per hour, the red marks indicate the possible demands to meet by the agent exactly. The blue curve shows the tendency. Note that the demand is only available between 3 and 15 parts/hour. Based on the process times applied as the fastest and assuming the energy signatures, the demand that can be met, while moving towards energy optimization, range between these values.

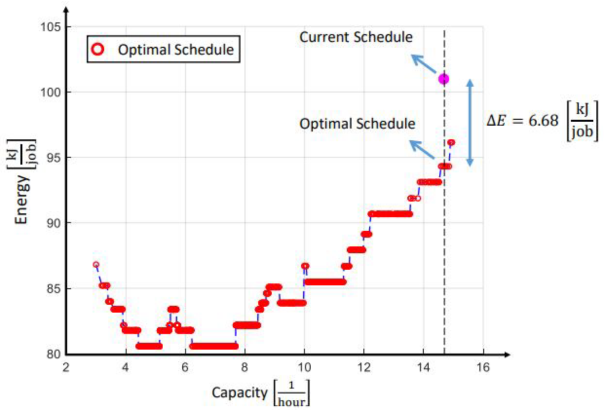

Fig. 7. Energy-capacity diagram found by the RL agent.

5. Discussion

The results show that RL could find solutions for the described scenario, nevertheless, using reinforcement learning for this application adds complexity to the problem. Defining a reward signal that converges to the solution is a difficult task, and there are plenty of parameters to tune such as, the training parameters, the agent configuration (neural networks), etc. Further, deployment of the agent (using the trained agent) is not easy, especially when neural networks are used for training. The motivation to use RL combined with DES to solve a scheduling problem was both to be able to be responsive to uncertainties and to have a trained agent to be used as a controller, being able to reschedule a cell if needed. The result demonstrates the concept of applying DES in combination with RL. Thus, the next part of extending the research intends to incorporate variations and uncertainties into the DES models. Additional extensions of the research could for one be to use the full potential of DES so that the simulation of the cell for every episode could be done for several days, meaning that the agent could realize the long-term effect of its actions. Further, the accuracy used for energy measurements can be a subject of discussion, as the assumption that all energy signatures followed the same pattern was made. For this purpose, it was enough, as the idea was to prove that the concept works, and that energy can be reduced if an agent can tune process times to meet a given demand.

6. Conclusion

Inverse scheduling is proposed to formulate scheduling as a control problem. In this paradigm, capacity and energy error signals are tried to be minimized by adjusting number of jobs and duration times as reference signals. The problem is translated into the framework of RL, and an agent is trained to solve it. The result is a trained policy that can meet a capacity demand while minimizing makespan and energy use. The motivation to use DES in a reinforcement learning framework was to integrate uncertainties and failures to the model, so that the energy consumption and makespan would not simply be obtained by a mathematical equation. Thus, the model could include the possibilities of variations and hence simulate a more realistic production environment. High complexity and uncertainties in PPC make multi-objective scheduling challenging, thus this study address this through outlining a conceptual framework of scheduling applying discrete event simulation as an environment for deep reinforcement learning. The study revealed opportunities for novel approaches of studying and reducing energy in manufacturing settings. The results demonstrate a move towards a more holistic approach for production scheduling, which includes energy usage, that can aid decision-making and facilitate increased sustainability in manufacturing.

Future work intends to extend this research to incorporate uncertainties and variations occurring during manufacturing processes. In many real manufacturing settings, there are for example uncertainties in processing times and machine failures. By using DES these uncertainties can be included, so that the RL agent gets trained within an environment closer to reality. The described case did not yet include uncertainties. However, with the application of DES as the simulation environment in the proposed concept, we will extend and explore this approach further in the future.

Acknowledgements

The work was carried out at the Production Technology Centre at University West, Sweden supported by the Swedish Governmental Agency for Innovation Systems (Vinnova) under the project SmoothIT and by the KK Foundation under the project Artificial and Human Intelligence through Learning (AHIL). Their support is gratefully acknowledged. Assistance provided by Master's students María Vincenta Vivo and Mohammadali Zakeriharandi was greatly appreciated.

References

[1] Seow Y, Rahimifard S. A framework for modelling energy consumption within manufacturing systems. CIRP Journal of Manufacturing Science and Technology 2011;4:258-264.
[2] Thiede S, Seow Y, Andersson J, Johansson B. Environmental aspects in manufacturing system modelling and simulation - State of the art and research perspectives. CIRP Journal of Manufacturing Science and Technology 2013;6:78-87.
[3] Siemieniuch CE, Sinclair MA, Henshaw MJdeC. Global Drivers, sustainable manufacturing and systems ergonomics. Applied Ergonomics 2015;51:104-119.
[4] Mouzon G, Yildirim MB, Twomey J. Operational methods for minimization of energy consumption of manufacturing equipment. International Journal of Production Research 2007;45;18-19:4247-4271.
[5] Zarte M, Pechmann A, Nunes IL. Decision support systems for sustainable manufacturing surrounding the product and production life cycle - A literature review. Journal of Cleaner Production 2019;219:336-349.
[6] Pinedo ML.Planning and Scheduling in Manufacturing and Services. 2nd ed. New York: Springer-Verlag; 2009.
[7] Sundström N, Wigström O, Lennartson B. Conflict between energy, stability, and robustness in production schedules. IEEE Transactions on Automation Science and Engineering 2017;14;2:658–668.
[8] Bokrantz J, Skoogh A, Ylipää T, Stahre J. Handling of production disturbances in the manufacturing industry. Journal of Manufacturing Technology Management 2016;27;8:1054–1075.
[9] Sutton RS, Barto AG. Reinforcement learning: An introduction. MIT press; 2018.
[10] Skoogh A, Johansson B, Hanson L. Data requirements for simulation of energy consumption in productions systems. 44th CIRP Conference on Manufacturing Systems, Madison, WI, USA, 2011.
[11] Hanen C. Study of a NP-hard cyclic scheduling problem: The recurrent job-shop. European journal of operational research 1994;72;1:82–101.
[12] Gabel T, Riedmiller M. Distributed policy search reinforcement learning for job-shop scheduling tasks. International Journal of Production Research 2012;50;1:41–61.
[13] Hashimoto N, Boyali A, Kato S, Otsuka T, Mizushima K, Omae M. Stochastic Discrete Event Simulation Environment for Autonomous Cart Fleet for Artificial Intelligent Training and Reinforcement Learning Algorithms. Technical Report 2018. DOI: 10.13140/RG.2.2.29769.24160.
[14] Creighton DC, Nahavandi S. Optimising discrete event simulation models using a reinforcement learning agent. Proceedings of the Winter Simulation Conference 2002:1945-1950.
[15] Garwood TL, Hughes BR, Oates MR, O'Connor D, Hughes R. A review of energy simulation tools for the manufacturing sector. Renewable and Sustainable Energy Reviews 2018;81:895-911.
[16] Hovgard M, Lennartson B, Bengtsson K. Simulation based energy optimization of robot stations by motion parameter tuning. IEEE 15th Int Conf on Automation Science and Engineering (CASE), Vancouver, 2019.
[17] Ramasamy S, Eriksson K, Perumal B, Peralippatt S,Danielsson F. Optimized path planning by Adaptive RRT* algorithm for constrained environments considering energy. IEEE 26th International Conference on Emerging Technologies and Factory Automation (ETFA) 2021.
[18] Riazi S, Wigström O, Bengtsson K, Lennartson B. Energy and peak power optimization of time-bounded robot trajectories. IEEE Transactions on Automation Science and Engineering 2017;14;2:646–657.
[19] Glorieux E, Danielsson F, Svensson B, Lennartsson B. Constructive cooperative coevolutionary optimisation for interacting production stations. The International Journal of Advanced Manufacturing Technology 2015;80:673–688.
[20] Dietmair A, Verl A. A generic energy consumption model for decision making and energy efficiency optimisation in manufacturing. International Journal of Sustainable Engineering 2009;2;2:123–133.
[21] Eriksson K, Juste A, Junefjäll J. Reducing energy consumption through production flow simulation: A case study at a car engine manufacturer. 8th International EurOMA Sustainable Operations and Supply Chain Forum. La Rochelle Business School, France, 22nd – 23rd March 2021.
[22] Cataldo A, Taisch M, Stahl B. Modelling, simulation and evaluation of energy consumptions for a manufacturing production line. IECON 39th Annual Conference of the IEEE Ind. Electronics Society, Vienna, 2013.
[23] Zhang L, Li X, Gao L, Zhang G. Dynamic rescheduling in FMS that is simultaneously considering energy consumption and schedule efficiency. Int. J. Adv. Manuf. Technol. 2016;87:5–8:1387–1399.
[24] Salido MA, Escamilla J, Barber F, Giret A. Rescheduling in jobshop problems for sustainable manufacturing systems. J. Cleaner Prod. 2017; 162:S121–S132.
[25] Zhang Y, Wang J, Liu Y. Game theory based real-time multiobjective flexible job shop scheduling considering environmental impact. J. Cleaner Prod. 2017;167:665–679.
[26] Banks J, Nelson JS, Nelson BL, Nicol D. Discrete-Event System Simulation. 5th ed. Prentice Hall, 2010.
[27] Sundström N. Robust and energy efficient scheduling. PhD Thesis. Chalmers University of Technology, Gothenburg, 2017.
[28] Mnih V. et al. Playing atari with deep reinforcement learning. arXiv preprint arXiv:1312.5602, 2013.