Collective Learning for Energy-centric Flexible Job Shop Scheduling

Narayanan Arun, Pournaras Evangelos, Nardelli Pedro H. J.

DOI: 10.1109/ISIE51358.2023.10228029

Collective Learning for Energy-centric Flexible Job Shop Scheduling

Arun Narayanan
Department of Electrical Engineering
LUT University
Lappeenranta, Finland
arun.narayanan@lut.fi

Evangelos Pournaras
School of Computing
University of Leeds
Leeds, United Kingdom
e.pournaras@leeds.ac.uk

Pedro H. J. Nardelli
Department of Electrical Engineering
LUT University
Lappeenranta, Finland
pedro.nardelli@lut.fi

***Abstract*—Manufacturing industries can reduce their energy consumption by exploiting the flexibility in manufacturing processes, such as machine availability, flexible jobs, and resource usage. In this paper, we exploit the flexibility inherent in some jobs, their schedules, and their energy consumption to model an energy-centric flexible job shop scheduling problem. We assume that there is limited electrical energy to power machines, and we attempt to match the schedules of machine jobs to the available energy with the objective of minimizing the energy consumption. We propose that collective learning, i.e., a form of decentralized (and unsupervised) learning where autonomous agents coordinate their decision-making to collectively learn and manage tasks that can be efficiently performed by coordination, can be employed to achieve this. We present a methodology that combines a plan-generation algorithm with a collective-learning tool—Iterative Economic Planning and Optimized Selections (I-EPOS)—to solve this problem with near optimal-solutions. We apply the methodology to a practical dataset comprising 3 machines and 12 jobs and show that the energy consumption decreases by approximately 5% when they choose and schedule the jobs, instead of acting independently or without any coordination. We also show that it is possible to scale this method to a large number of jobs to obtain reasonable solutions quickly, and that coordination always outperforms uncoordinated and independent actions by increasing the energy savings.**

***Index Terms*—collective learning, flexible job shop scheduling, manufacturing industry, energy consumption, energy efficiency**

I. Introduction

Environmental pollution and climate change are well-known problems today, and substantial research efforts are ongoing to mitigate their impacts. A recognized pathway to reduce their impacts is to utilize renewable energy resources to meet energy demands and to improve the energy efficiency of energy-production and delivery processes [1]–[3]. Managing energy consumption and ensuring high energy efficiency is especially important in the industrial sector. In particular, manufacturing industries often use highly energy-intensive processes that consume tremendous amounts of energy. In 2021, the industrial sector accounted for 33% of the total energy consumption in the US, and manufacturing alone consumed nearly 81% of the annual industrial energy consumption [4]. Moreover, industrial activity was directly responsible for nearly 25% of the global emissions in 2021 [5]. Therefore, in order to achieve "net-zero" emissions by 2050, it is imperative that manufacturing industries adopt energy-efficiency measures [5].

To improve the energy efficiencies of manufacturing processes, manufacturing industries can try to exploit any flexibility available in the various processes that create the intended product [6]. For example, machines and workers could be selected based on their availability in such a way that idle time is avoided; jobs and processes could be rescheduled based on the availability of resources; and machines could be switched on/off according to the requirements [7]. Similarly, another approach could be to match machines to job schedules based on the available energy. This is especially pertinent when the energy is limited, for example, when renewable energy is used to supply electricity. In this paper, we consider such a problem where the machines involved in the manufacturing processes are powered by battery energy storage systems (BESS) alone during the night. This represents a scenario in which a manufacturer wishes to maximize renewable energy usage, and hence, uses BESS as the backup supply. These machines are required to perform jobs that have a number of sub-processes, i.e., operations, that must be performed sequentially or non-sequentially. Further, we also assume that some of the operations can be performed by more than one machine, but different machines will use different amounts of energy and time, depending, for example, on their model, manufacture, and age. Hence, given a set of machines, it is necessary to schedule all the operations required to produce an output in such a way that the available stored energy is used as efficiently as possible. Thus, in this paper, we consider the problem of *energy-efficient scheduling* of flexible manufacturing jobs, processes, and operations.

This problem is essentially a variation of the job shop scheduling problem (JSSP), a well-established combinatorial optimization problem [8]. In the JSSP, n different jobs—$J = \{J_1, J_2, ..., J_n\}$—have to be scheduled on m machines with different capabilities such as power ratings, speeds, efficiencies, or processing power. Further, each job J_i consists of a set of k operations—$O_i = \{O_{i1}, O_{i2}, ..., O_{ik}\}$—that must be performed in a given sequence. Here, each operation can be processed by only one specific machine, and moreover, only one operation in a job can be processed at a time. In the *flexible* JSSP (FJSSP) variant, some, or all, the operations can be processed on any machine [9]. Traditionally, the JSSP and FJSSP aimed to schedule the tasks on the machines while minimizing

the time needed to complete all the jobs (i.e., the makespan), the cost required, or other production efficiency measures. More recently, researchers have attempted to minimize energy consumption, maximize energy efficiency, or meet some other energy-related objectives (e.g., minimization of carbon footprint, CO_2 emissions, etc.) in the JSSP and FJSSP, leading to an energy-centric FJSSP that emphasizes energy saving. To do this, many researchers have proposed to exploit the flexibility available in some manufacturing processes and machines, e.g., by machine selection, job sequencing, distributed scheduling, decision making by switching off/on the machines, or worker availability [10]–[14].

FJSSP and energy-centric FJSSP can be modeled as a mixed-integer linear programming (MILP) problem and solved exactly, as in, for example, [8]–[10], [12], [13], [15], [16]. However, the MILP-based approach is computationally infeasible and inefficient when the number of jobs increases. This is because the JSSP and FJSSP are NP-hard problems whose solution spaces increase tremendously with the number of jobs, with the number of possible job–machine combinations being of the order of $(n!)^m$ [11]. As a result, heuristics that give near-optimal solutions have been used to solve many of these problems [11], [17], for example, genetic algorithms [9], [11], [18], [19], evolutionary algorithms [7], [9], [17], and swarm intelligence methods such as particle swarm optimization [20], while, more recently, deep learning has been employed [21]. However, these heuristics and methods are also limited in terms of scalability and fail to give solutions in reasonable time for a large number of jobs; swarm-intelligence methods, for example, can be computationally intensive in terms of both memory and time [22].

In this paper, we propose a highly scalable "collective learning"-based heuristic based on the decentralized coordination of agents' choices. Collective learning has been recently proposed as a way to solve combinatorial optimization problems that involve coordination [23], [24]. Collective learning is a form of decentralized (and unsupervised) learning where autonomous agents coordinate their decision-making to collectively learn and manage tasks that can be efficiently performed by coordination, such as reducing electric power peaks [23], [24]. It leads to a type of "collective intelligence" that is based on the idea that in a network of connected agents, collective or coordinated actions among the agents can lead to better systemic performance than their independent non-cooperative actions [25]. The basic concept is as follows. Consider a network consisting of a set of n agents, $A = \{a_1, a_2, ..., a_n\}$, that can coordinate with each other if needed. Further, assume that every agent a_i ($\forall i = 1, 2, ..., n$) performs individual tasks that are associated with a certain cost c_i ($\forall i = 1, 2, ..., n$) so that the set of costs is $C = \{c_1, c_2, ..., c_n\}$. The idea is that in some networks, the total cost of the network when the agents coordinate, C_{coll}, is at most the sum of the costs when the agents do not coordinate, given as follows:

$$C_{\text{coll}} \leq \sum_{i=1}^{n} c_i \tag{1}$$

Thus, if the agents exchange information and coordinate among themselves, the system cost, or the global cost, can decrease. Note that the agents' individual costs C, or the local costs, might also change and become worse. Thus, the system performance improves, often at the cost of individual performances. Swarm intelligence algorithms such as ant colony optimization and bee swarm optimization are examples of the application of the concept of collective intelligence.

In this paper, we show how the collective-learning approach can potentially be used to solve the energy-centric FJSSP and obtain reasonable solutions quickly. In particular, we use the collective learning approach proposed by [23] wherein the authors have set up collective learning as a fully decentralized approach for coordinated multi-objective decision-making in multi-agent systems. We use their general-purpose decentralized collective-learning algorithm and tool—Iterative Economic Planning and Optimized Selections (I-EPOS). This collective-learning approach and I-EPOS have been previously applied to multi-agent systems and scheduling and resource allocation problems, for example, for appliance scheduling [26], bike sharing [24], energy demand-response [23], distributed sensing with drones [27], and prosumer flexibility [28], yielding near-optimal solutions. Collective-learning-based methods are able to achieve effective coordination and scalability through parallel processing, and hence, they can be especially useful for industrial applications in which machines can operate at the same time, in parallel, choosing their operations from a set of discrete options while constrained by some parameters like energy consumption. Here, we model the machines as independent agents that can choose to perform the operations independently or in cooperation with other machines, with both these choice leading to different system-level outcomes. We use a practical example of energy-centric FJSSP to demonstrate the potential applicability of the I-EPOS heuristic, and further, reasonable solutions can be reached quickly even after scaling the problem to large number of jobs, despite the large solution space.

The remaining paper is organized as follows. In Section II, we present our problem model with a mathematical formulation. We also describe our approach to solve the problem using I-EPOS. In Section III, we apply the approach to a practical use case and we discuss the obtained results. Finally, in Section IV, we conclude the paper by describing the limitations of this approach and possible further improvements.

II. Methodology

A. Problem Formulation

A complete mathematical MILP formulation of the FJSSP as well as the energy-centric FJSSP is given in detail in [15] and [12], respectively. In this paper, we consider a smaller problem that considers energy supplied by BESS to the machines. Consider a set of m machines $M = \{M_1, M_2, ..., M_m\}$, a set of n jobs $J = \{J_1, J_2, ..., J_n\}$, and a set of k operations for each job J_i, $O_i = \{O_{i1}, O_{i2}, ..., O_{ik}\}$. Every operation O_{ij} ($\forall\ i = 1, ..., n$ and $j = 1, ..., k$) can be processed by employing a subset of eligible machines

$M_{ij} \subseteq M$. In this paper, for simplicity, we assume that every operation can be processed by *all* the machines, i.e., the subset of eligible machines M_{ij} is the set M: $M_{ij} = M$. We also assume that the sequence of operations is not important for completing a job. Finally, we assume that all the machines are either equipped with or powered by their own BESS so that the total number of BESS is the same as the total number of machines, m. The set of the energy capacities in the BESS at a time t is given by $B = \{B_1(t), 2B_2(t), ..., B_m(t)\}$.

In addition, similar to the traditional JSSP and FJSSP formulations, we make the following assumptions:

1) All the machines are available in the beginning (time 0).
2) The operations of different jobs are independent.
3) All the machines process only one operation at a time.
4) An operation cannot be interrupted once it is started.
5) All machining data such as processing time, processing power, etc., are known in advance.
6) BESS is charged to its maximum capacity in the beginning (time 0), and this available energy is known.

In our energy-centric FJSSP, the aim is to choose the appropriate machine for each operation and determine the sequence of operations—i.e., the operational schedule—for every machine with the objective of minimizing the total energy consumption. The total energy consumption depends on the machine selected for a particular operation. Therefore, the objective function is as follows:

$$\min \left(\sum_{i=1}^{n} \sum_{j=1}^{o} \sum_{k=1}^{m} (x_{ijk} \times E_{ijk}) \right) \tag{2}$$

Here, the binary decision variable x_{ijk} is 1 if a machine k is selected for an operation O_{ij} and 0, otherwise. E_{ijk} is the energy consumed by a machine k for performing an operation O_{ij}, i.e., the j^{th} operation of the i^{th} job. We then have the following constraints:

$$\sum_{k \in M} x_{ijk} = 1 \qquad (\forall i \in J, \quad \forall j \in O_i) \tag{3}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{o} (x_{ij1} \times E_{ij1}) \leq B_1(0) \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{o} (x_{ij2} \times E_{ij2}) \leq B_2(0) \tag{5}$$

$$\vdots$$

$$\sum_{i=1}^{n} \sum_{j=1}^{o} (x_{ijm} \times E_{ijm}) \leq B_m(0) \tag{6}$$

Constraint 3 ensures that only one machine performs an operation O_{ij} and that every operation is performed. Constraints 4–6 ensures that the total energy consumed by a machine k does not exceed the capacity, at time 0, of the BESS that is supplying power to it.

B. Collective-learning-based Method

In this paper, we apply collective-learning principles and specifically the collective-learning tool, I-EPOS, to solve this problem. The mechanism of the I-EPOS algorithm is described in detail with rigorous analyses in [23] and [24], and we only describe its basic idea briefly here. Consider a network comprising a set of n agents, $A = \{a_1, a_2, ..., a_n\}$, that can coordinate with each other if needed. Every such agent creates a finite set of *possible plans*, where a possible plan p_i is a vector x of size d with real values. Every possible plan p_i is associated with a preference, often given by the cost of carrying out the plan. For example, consider a city that is implementing bike sharing. A citizen plans to take a bike from a station i and deposit it at a station j, and this plan can be encoded as a vector of integers where taking and depositing a bike are encoded by negative decrements and positive increments, respectively. In other words, a citizen traveling from station 5 to station 1 and then from station 1 to station 3 may have the following possible plan: (0, 0, 1, 0, -1, ...), and this could associated with a cost 0.9, expressing their preference for this plan. Similarly, a citizen making trips from station 1 to station 3 and then from station 2 to station 4 and from station 4 to station 5 will have the following plan: (-1, -1, 1, 0, 1, ...). The citizen, who is the agent here, makes several such plans, each with a different preference (or cost).

From such a finite set of plans, the agent selects one and only one plan—the *selected plan*—to determine its operation. Similarly, all the agents in the network choose their own selected plans. Then, we can obtain an aggregated response by summing up (element-wise) the selected plans and the cost of the selected plans of every agent. We thereby obtain a global response vector with an associated global cost. I-EPOS then aims to minimize the global cost. In this manner, agents together cooperatively select plans that lead to a minimum global cost. This cooperation is particularly effective when agents' choices depend on each other. Note that with I-EPOS, it is also possible to balance agents' and system's preferences by controlling a local parameter β that makes the algorithm prefer agents' individual preferences.

To solve the energy-centric FJSSP presented in Section II-A using I-EPOS, we make a small simplification: we consider that every job consists of only 1 operation so that $n = o$. Thus, we have k machines and n *flexible* jobs that can be performed by any of the k machines. More complicated setups and algorithms will be considered in the future. We consider that each machine is an agent and we generate a finite set of possible plans for every machine. Each plan corresponds to a set of jobs and is associated with the energy consumed while performing the jobs. For example, one possible plan for a machine, M_i, could be $1.256 : \{J_{i1}, J_{i3}, J_{i7}, J_{i9}\}$, implying that M_i performs the jobs 1, 3, 7, and 9 using 1.256 kWh. Plan generation is key for achieving optimized solutions, and we use the plan-generation algorithm given in Algorithm 1 to generate a finite set of possible plans for all the machines.

First, a machine k is selected randomly from the set

of machines. The machine is selected randomly instead of sequentially because otherwise, in every plan, the first machine will get the highest priority in choosing the jobs, the second machine the second-highest priority, and so on. This will make the plans unbalanced in favor of the first machines in the list, and therefore, random selection is used to promote diversity of plans and thereby increase the solution space. Then, an empty plan set P_k is initialized for the selected machine M_k: $P_k = \{\}$. A set of feasible jobs, F, is constructed from the set of jobs J, based on the abovementioned energy constraints 4–6 (line 12, where E_k and E_{jk} refer to the energy accumulated by a machine k and the energy used by the machine k for an job j, respectively). From this feasible set, a job is randomly selected and added to the machine plan P_k. This job is also removed from J so that it is not performed by another machine, thereby satisfying constraint 3 given earlier. The feasible set is reconstructed and additional jobs are added to P_k until there are no feasible jobs left. In this manner, the path of the selected machine is fully constructed.

Subsequently, another machine is selected and its plan is constructed using the remaining jobs. This is continued until all the jobs are selected. Note that it is not necessary that all the machines are needed for performing all the jobs. It is possible that all the jobs are selected but only a subset of the set of machines is used. In other words, the algorithm priorities job completion and not machine usage. We then generate p set of plans for the machines such that each set satisfies constraints 3–6. These plans are then combined together by I-EPOS to achieve a set of plans with the minimum total consumed energy, thereby meeting the objective specified in Eq. 2.

Algorithm 1 Plan generation algorithm for energy-centric flexible job shop scheduling problem

1: **Inputs**: (1) The set of machines, $M = \{M_1, M_2, M_3, ..., M_m\}$; (2) The set of jobs that need to be performed $J = \{J_1, J_2, J_3, ..., J_n\}$; (3) Energy consumed by all the machines for performing the operations, E_{jk} $(\forall k = 1, ..., m; j = 1, ..., n)$; (4) Battery energy for each machine: $B_1(0)$, $B_2(0)$,,...,$B_m(0)$.
2: Initialize a counter for the machines used $r = 0$
3: **while** $J \neq \emptyset$ and $r <= m$ **do**
4: Randomly select a machine M_k, $\forall\ k = 1, 2, ..., m$
5: Increment the machine counter: $r = r + 1$
6: Initialize plan of machine M_k: $P_k = \{\}$
7: Initialize machine M_k's energy used: $E_k = 0$
8: Initialize a Boolean variable to check if feasible solutions exist: feas $= 1$
9: **while** feas $== 1$ **do**
10: Initialize feasible set of operations $F = \{\}$
11: **for** j = 1:j++:|J| **do**
12: **if** $E_k + E_{jk} <= B_k(0)$ **then**
13: $F = F \cup \{J_j\}$
14: **end if**
15: **end for**
16: **if** $|F| > 0$ **then**
17: Randomly select a job from F: J_i
18: Update machine M_k's plan P_k: $P_k = P_k \cup J_i$
19: Update set of jobs J: $J = J \setminus J_i$
20: **else**
21: feas $= 0$
22: **end if**
23: **end while**
24: **end while**

III. Results and Discussions

To demonstrate our proposed methodology, we first applied it to a practical energy-intensive additive manufacturing process (3D printing) introduced in [29] and further explored in [30]. The authors have presented the following scenario. Three different fused deposition modeling (FDM) machines manufacture four different parts, with each manufacturing process consisting of three stages—pre-heating, printing, and bed cooling. Further, they have given actual measured power consumption of the 3 stages (equivalent to operations in our paper) used by the 3 machines for constructing each of the 4 parts (equivalent to jobs). Thus, 3 machines perform 4 jobs each of which has 3 operations. In this paper, we consider the energy consumption for the jobs collectively (obtained by adding the energy consumption for their operations) so that we have 3 machines performing 4 jobs; the energy demand for the 3 machines to perform these 4 jobs are given in Table I. Note that similar to [30], we have multiplied all the energy demand values by 10 to scale up the problem.

Further, we consider that the objective is to produce 3 parts of each type, equalling 12 products, with the minimum energy consumption. Thus, we have a set of machines $M = \{M_1, M_2, M_3\}$ ($m = 3$ machines) and 3 sets of jobs—$J_1 = \{J_{1_1}, J_{1_2}, ..., J_{1_{12}}\}$ ($n_1 = 12$ jobs); $J_2 = \{J_{2_1}, J_{2_2}, ..., J_{2_{12}}\}$ ($n_2 = 12$ jobs); and $J_3 = \{J_{3_1}, J_{3_2}, ..., J_{3_{12}}\}$ ($n_3 = 12$ jobs)—corresponding to machines 1, 2, and 3, respectively. In [30], the authors considered a BESS energy of 3 kWh for the 4 jobs. In this paper, we similarly assume that each of the machines is powered by a BESS of $3 \times 3 = 12$ kWh since we have 12 jobs. Thus, $B_1(0) = B_2(0) = B_3(0) = 12$ kWh. The plans generated with Algorithm 1 were then combined using I-EPOS such that all 12 jobs were performed by the 3 machines with minimum energy consumption. We ran I-EPOS 10 times independently with the following parameters: number of simulations = 100; number of iterations = 20; and number of plans $p = 100$. The plan with the minimum cost, i.e., the minimum energy consumption was selected.

If the 3 machines, m_1, m_2, and m_3, performed all 12 jobs independently by themselves, then their energy consumption values were 22.85, 22.07, and 24.20 kWh, respectively. If they performed the jobs together without any coordination, with the machines randomly choosing to perform any available job, the machines together consumed an energy of 22.541 kWh. However, when the machines coordinated among themselves, their total energy consumption was 21.264 kWh, a savings of 5% over the best performing machine and 2% over the uncoordinated case. The following set of jobs

TABLE I: Energy demand (kWh) for 3 different fused deposition modeling machines to perform 4 different jobs (Dataset adapted from [30]).

Jobs	Machine 1 (kWh)	Machine 2 (kWh)	Machine 3 (kWh)
1	1.7967	1.99	2.292
2	2.2597	2.1203	2.351
3	2.4567	2.2053	2.1287
4	1.104	1.0423	1.295

were chosen for machines 1, 2, and 3, respectively: $[1, 5, 9]$, $[2, 4, 6, 8, 10, 12]$, and $[3, 7, 11]$. Note that this also implies that instead of the input total BESS energy values of 36 kWh $(B_1(0) + B_2(0) + B_3(0))$, BESS energy of 21.264 kWh is sufficient to power all 3 machines together to perform the 12 jobs if they follow the determined optimum schedule.

We also performed additional experiments to demonstrate the scalability of I-EPOS for this problem. Unfortunately, large industrial datasets are not publicly available, and simply repeating the jobs in Table I does not lead to useful or insightful results. Hence, instead, we sampled random energy values for the number of chosen jobs from a uniform random distribution between [1 2.5] kWh, a range similar to that of Table I. The energy consumption values used by us for our first experiment with 3 machines and 10 jobs are listed in Table II. We obtained the BESS energy by proportionally increasing the value of 3 kWh for the 4 jobs to n jobs: $B_m(0) = \text{floor } (n_m \times 3/4) \ \ \forall\, m = 1, 2, 3$. Here, *floor* refers to the floor function: $\lfloor x \rfloor = \max\{m \in \mathbb{Z} \mid m \leq x\}$ where $x \in \mathbb{R}$, and $m \in \mathbb{Z}$. We use this to set a stronger additional limit to guide the algorithm to determine optimal schedules.

For the energy consumption values listed in Table II, we found that when the machines 1, 2, and 3 coordinated, they selected operations $[1, 4]$, $[2, 6, 7, 8]$, and $[3, 5, 9, 10]$, respectively, using only 16.11 kWh. On the other hand, when m_1, m_2, and m_3 independently did the jobs, they required 19.25, 18.22, and 18.71 kWh, respectively. In the case of an uncoordinated approach, the machines together consumed an energy of 16.58 kWh. Thus, there were savings of 12% and 3% over the best performing machine as well as the uncoordinated case, respectively. We repeated this experiment with 3 machines and 50, 100, 500, 1000, 5000, and 10000 jobs and compared the energy consumption when the machines coordinated with each other with two cases: (1) the machines acted independently and alone (we took the best performing machine, i.e., the machine with the lowest energy consumption) and (2) the machines acted together but without coordination. As shown in Fig. 1, coordination always lowered the energy consumption and led to energy savings, and the energy savings increased with the number of jobs.

It is important to note here that since the problem is NP-hard, our proposed methodology does not guarantee an optimum solution especially when the problem is scaled up to a large number of jobs. Instead, our methodology is able to give a reasonable solution quickly. Using a laptop with Intel

TABLE II: Energy demand (kWh) for three different machines to perform 10 jobs (Randomly created dataset).

Jobs	Machine 1 (kWh)	Machine 2 (kWh)	Machine 3 (kWh)
1	1.82	2.19	2.47
2	2.07	1.79	2.2
3	1.9	1.85	1.69
4	1.82	2.39	2.17
5	1.64	1.11	1.18
6	1.97	1.13	1.96
7	1.66	1.03	1.22
8	2.34	2.25	2.42
9	2.45	2.17	1.78
10	1.58	2.31	1.62

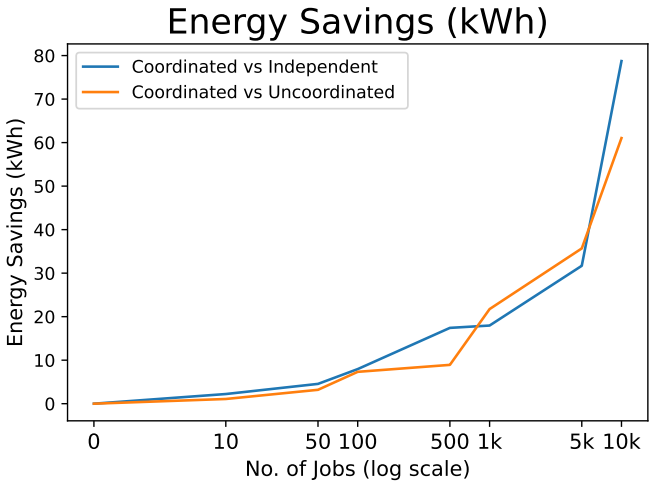

Fig. 1: Energy savings, given by the difference in the energy consumed when the 3 machines were coordinated versus when (1) the machines acted independently and alone and (2) the machines acted together but without any coordination, as the number of jobs increased from 10 to 10000.

Core(TM) i5-8400H CPU @ 2.50GHz and 32 GB RAM, the proposed methodology found the solution for the case with 50, 100, 500, 1000, 5000, and 10000 jobs in 2, 2.5, 6.5, 10.5, 84, and 230 min, respectively. Note that for 50, 100, 500, and 1000 jobs we ran I-EPOS 10 times with the number of plans $p = 100$. But, for 5000 and 10000 jobs, we increased p to $p = 1000$ and ran I-EPOS only once in order to obtain a quick, but reasonable, result despite the huge number of jobs. This has practical importance for many industries since they can save significant costs by obtaining planning solutions quickly and efficiently.

IV. Conclusions

In this paper, we addressed the flexible job shop scheduling problem with the objective of minimizing energy consumption. We showed that collective-intelligence and collective-learning principles can be effectively used to achieve coordination

among machines, and moreover, that this coordination can lead to efficient energy usage. We presented a plan-generation algorithm that can be used with the collective-learning tool I-EPOS to obtain optimal schedules even when there are a large number of operations. Using a relatively simple, but practical, scenario, we have shown the potential of collective-learning principles, methods, and tools to tackle energy-centric FJSSP.

The proposed methodology has a few limitations currently; it is relatively simple and does not consider time as a variable, leading to unbalanced schedules where some machines perform fewer jobs. We plan to extend the methodology to make it more generalized and practical. We will also apply the methodology to larger and more complex datasets to demonstrate its effectiveness in real-world scenarios. Further, we plan to match produced renewable energy with flexible job schedules, for example, by considering that solar panels supply power to the machines and the BESS, with the objective that the renewable energy is fully utilized. Other research directions include exploiting additional flexibility offered by manufacturing processes, such as distributed scheduling and machine idling and introducing a preferred sequence of operations as a constraint.

Acknowledgements

This research is partly supported by (1) Academy of Finland via: (a) FIREMAN consortium n.326270 part of CHIST-ERA-17-BDSI-003, (b) EnergyNet Fellowship n.321265/n.328869/n.352654, and (c) X-SDEN project n.349965; (2) Baltic-Nordic Energy Research programme via Next-uGrid project n.117766; (3) JAES Foundation via project STREAM. Evangelos Pournaras is supported by a UKRI Future Leaders Fellowship (MR/W009560/1): '*Digitally Assisted Collective Governance of Smart City Commons–ARTIO*'.

References

[1] O. Edenhofer, R. Pichs-Madruga, Y. Sokona, K. Seyboth, S. Kadner, T. Zwickel, P. Eickemeier, G. Hansen, S. Schlömer, C. von Stechow *et al.*, *Renewable energy sources and climate change mitigation: Special report of the intergovernmental panel on climate change*. Cambridge University Press, 2011.

[2] D. Gielen, F. Boshell, and D. Saygin, "Climate and energy challenges for materials science," *Nature materials*, vol. 15, no. 2, pp. 117–120, 2016.

[3] J. Twidell, *Renewable energy resources*. Routledge, 2021.

[4] US Energy Information Administration. Industry, publiher = IEA, Paris, year = 2022, url = https://www.iea.org/reports/industry, urldate = 2023-02-14,.

[5] International Energy Agency. Industry, year = 2023, url = https://www.eia.gov/energyexplained/use-of-energy/industry.php.

[6] A. Tristán, F. Heuberger, and A. Sauer, "A methodology to systematically identify and characterize energy flexibility measures in industrial systems," *Energies*, vol. 13, no. 22, p. 5887, 2020.

[7] Q. Luo, Q. Deng, G. Xie, and G. Gong, "A pareto-based two-stage evolutionary algorithm for flexible job shop scheduling problem with worker cooperation flexibility," *Robotics and Computer-Integrated Manufacturing*, vol. 82, p. 102534, 2023.

[8] H. Xiong, S. Shi, D. Ren, and J. Hu, "A survey of job shop scheduling problem: The types and models," *Computers & Operations Research*, p. 105731, 2022.

[9] J. Xie, L. Gao, K. Peng, X. Li, and H. Li, "Review on flexible job shop scheduling," *IET Collaborative Intelligent Manufacturing*, vol. 1, no. 3, pp. 67–77, 2019.

[10] L. Zhang, Q. Tang, Z. Wu, and F. Wang, "Mathematical modeling and evolutionary generation of rule sets for energy-efficient flexible job shops," *Energy*, vol. 138, pp. 210–227, 2017.

[11] J. Para, J. Del Ser, and A. J. Nebro, "Energy-aware multi-objective job shop scheduling optimization with metaheuristics in manufacturing industries: A critical survey, results, and perspectives," *Applied Sciences*, vol. 12, no. 3, p. 1491, 2022.

[12] L. Meng, C. Zhang, X. Shao, and Y. Ren, "MILP models for energy-aware flexible job shop scheduling problem," *Journal of cleaner production*, vol. 210, pp. 710–723, 2019.

[13] L. Meng, C. Zhang, Y. Ren, B. Zhang, and C. Lv, "Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem," *Computers & Industrial Engineering*, vol. 142, p. 106347, 2020.

[14] G. Gong, R. Chiong, Q. Deng, W. Han, L. Zhang, W. Lin, and K. Li, "Energy-efficient flexible flow shop scheduling with worker flexibility," *Expert Systems with Applications*, vol. 141, p. 112902, 2020.

[15] C. Özgüven, L. Özbakır, and Y. Yavuz, "Mathematical models for job-shop scheduling problems with routing and process plan flexibility," *Applied Mathematical Modelling*, vol. 34, no. 6, pp. 1539–1548, 2010.

[16] M.-J. Park and A. Ham, "Energy-aware flexible job shop scheduling under time-of-use pricing," *International Journal of Production Economics*, vol. 248, p. 108507, 2022.

[17] K. Gao, Z. Cao, L. Zhang, Z. Chen, Y. Han, and Q. Pan, "A review on swarm intelligence and evolutionary algorithms for solving flexible job shop scheduling problems," *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 4, pp. 904–916, 2019.

[18] G. May, B. Stahl, M. Taisch, and V. Prabhu, "Multi-objective genetic algorithm for energy-efficient job shop scheduling," *International Journal of Production Research*, vol. 53, no. 23, pp. 7071–7089, 2015.

[19] W. Xu, H. Sun, A. Awaga, Y. Yan, and Y. Cui, "Optimization approaches for solving production scheduling problem: A brief overview and a case study for hybrid flow shop using genetic algorithms," *Advances in Production Engineering & Management*, vol. 17, no. 1, pp. 45–56, 2022.

[20] D. Tang, M. Dai, M. A. Salido, and A. Giret, "Energy-efficient dynamic scheduling for a flexible flow shop using an improved particle swarm optimization," *Computers in industry*, vol. 81, pp. 82–95, 2016.

[21] B. Cunha, A. M. Madureira, B. Fonseca, and D. Coelho, "Deep reinforcement learning as a job shop scheduling solver: A literature review," in *Hybrid Intelligent Systems: 18th International Conference on Hybrid Intelligent Systems (HIS 2018) Held in Porto, Portugal, December 13-15, 2018 18*. Springer, 2020, pp. 350–359.

[22] Y. Tan and K. Ding, "A survey on gpu-based implementation of swarm intelligence algorithms," *IEEE transactions on cybernetics*, vol. 46, no. 9, pp. 2028–2041, 2015.

[23] E. Pournaras, P. Pilgerstorfer, and T. Asikis, "Decentralized Collective Learning for Self-managed Sharing Economies," *ACM Transactions on Autonomous and Adaptive Systems*, vol. 13, no. 2, pp. 1–33, Nov. 2018.

[24] E. Pournaras, "Collective learning: A 10-year odyssey to human-centered distributed intelligence," in *2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS)*. IEEE, 2020, pp. 205–214.

[25] A. Narayanan, M. Korium, D. C. Melgarejo, H. M. Hussain, A. S. De Sena, P. Goria, D. Gutierrez-Rojas, M. Ullah, A. Esmaeelnezhad, M. Rasti *et al.*, "Collective intelligence using 5g: Concepts, applications, and challenges in sociotechnical environments," *IEEE Access*, 2022.

[26] F. Fanitabasi and E. Pournaras, "Appliance-level flexible scheduling for socio-technical smart grid optimization," *IEEE Access*, vol. 8, pp. 119 880–119 898, 2020.

[27] C. Qin, F. Candan, L. S. Mihaylova, and E. Pournaras, "3, 2, 1, drones go! a testbed to take off uav swarm intelligence for distributed sensing," *arXiv preprint arXiv:2208.05914*, 2022.

[28] A. Mashlakov, E. Pournaras, P. H. Nardelli, and S. Honkapuro, "Decentralized cooperative scheduling of prosumer flexibility under forecast uncertainties," *Applied Energy*, vol. 290, p. 116706, 2021.

[29] S. Karimi, S. Kwon, and F. Ning, "Energy-aware production scheduling for additive manufacturing," *Journal of Cleaner Production*, vol. 278, p. 123183, 2021.

[30] S. Karimi and S. Kwon, "Comparative analysis of the impact of energy-aware scheduling, renewable energy generation, and battery energy storage on production scheduling," *International Journal of Energy Research*, vol. 45, no. 13, pp. 18 981–18 998, 2021.