IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 28 July 2023, accepted 27 August 2023, date of publication 6 September 2023, date of current version 13 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3312557

RESEARCH ARTICLE

Joint Production and Maintenance Scheduling for Total Cost and Machine Overload Reduction in Manufacturing: A Genetic Algorithm Approach

BRUNO MOTA, PEDRO FARIA, (Member, IEEE), AND CARLOS RAMOS, (Senior Member, IEEE)

Intelligent Systems Associate Laboratory (LASI), GECAD—Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), Polytechnic Institute of Porto (ISEP/IPP), 4200-072 Porto, Portugal

Corresponding author: Pedro Faria (pnf@isep.ipp.pt)

The present work has received funding from European Regional Development Fund through COMPETE 2020 - Operational Programme for Competitiveness and Internationalisation through the P2020 Project MUWO (ANI|P2020 POCI-01-0247-FEDER-069417), and has been developed under the EUREKA - ITEA3 Project MUWO (ITEA-19022), we also acknowledge the work facilities and equipment provided by GECAD research center (UIDB/00760/2020) to the project team. Pedro Faria is supported by the Grant CEECIND/01423/2021, from FCT.

ABSTRACT The European manufactory sector has been greatly impacted in recent times by the COVID-19 pandemic and the Russo-Ukrainian conflict, which have made energy prices soar to all-time highs reducing European companies' competitiveness in the global market. To remain competitive in these crises, manufacturing companies need to start optimizing not only their energy costs but also maintenance-derived costs, by better planning maintenance activities, through joint production and maintenance scheduling, and by improving machine longevity by way of reducing overload in single machines. Accordingly, the premise of the present paper is to propose an intelligent joint production and maintenance scheduling system to minimize total costs, that is, energy and maintenance costs, as well as minimize single-machine overload by balancing tasks between machines, while also allowing for imposed constraints in the production schedule. This is achieved through a Genetic Algorithm to solve the scheduling problem in flexible job shop manufacturing layouts. To reduce total costs, retailer energy price volatility, generated renewable energy resources availability and surplus selling, and maintenance stipulated hours prices are considered and benefited from as much as possible. Overload in single machines is reduced by minimizing the machine occupation rate standard deviation in the production schedule. A baseline scenario with real-production data from a work in the literature is used to validate the proposed scheduling system. Obtained results show that the proposed system is able to reliably reduce energy costs by 11.3% up to 15.4%, and single machine overload by 32.3% up to 52.7%.

INDEX TERMS Genetic algorithm, machine degradation optimization, production and maintenance scheduling, renewable energy resources, total cost optimization.

I. INTRODUCTION

The Russo-Ukrainian conflict and the COVID-19 pandemic have affected the European economies with sharp increases in energy prices and supply instability, crippling the European manufacturing sector [1], [2]. This has resulted in higher manufacturing expenses due to high energy costs, leaving many European companies on the brink of insolvency because of being unable to compete in the global market [3], [4]. One way for manufacturing companies to reduce the impact of high energy prices is the investment in Renewable Energy Resources (RERs) to increase grid stability and reduce dependability on external retailers [5], [6]. Moreover, companies can participate in the electricity market by selling their surplus RERs' energy, further mitigating the impact of high energy prices [7], [8]. Another advantage of RERs is their ability to replace, to some degree, fossil fuels, helping in fighting climate change [9]. Accordingly, it is essential

The associate editor coordinating the review of this manuscript and approving it for publication was Christian Pilato.

that companies take advantage of RERs through intelligence production scheduling algorithms that utilize Artificial Intelligence (AI) to obtain optimized schedules [10], [11].

While intelligent production scheduling in manufacturing is essential to optimize shop floors, primarily during unstable times, intelligent maintenance scheduling is also crucial to reduce the total expenses in a company, by optimizing maintenance activities to be more effective and efficient [12], [13]. According to [14] maintenance expenses can range between 15% and 70% of the cost of the product, leaving a big room for improvement. Nevertheless, when considering production and maintenance scheduling, there will always be trade-offs to consider, from maintenance postponement leading to an increase in machine failures to more time being spent repairing machines than manufacturing products [15]. This can be mitigated through preventive maintenance planning, which can be accomplished using AI techniques to obtain production and maintenance schedules that focus on minimizing the total costs (i.e., energy and maintenance costs) and overload in single machines (i.e., improving machine longevity) [16].

It is worth noting that, most works that address production and maintenance scheduling are directed towards more simplistic manufacturing layouts (e.g., flow shop or job shop), which can decrease execution times of algorithms, but have lower flexibility to be applied in real manufacturing environments [17]. Flexible Job Shop (FJSP) is an extension of the job-shop problem allowing for additional flexibility in scheduling but being more complex and NP-hard [18], [19].

A. LITERATURE REVIEW

The use of AI for complex production and/or maintenance scheduling is not a new concept, as there are already many AI algorithms explored in the literature. Within the scope of the present paper's problem, the most promising AI algorithms used in the literature are Particle Swarm Optimization (PSO) [20], Linear Programming (LP) [21], Simulated Annealing (SA) [22], Reinforcement Learning (RL) [23], and the most popular being the Genetic Algorithm (GA) [24], [25], [26], [27], [28]. Accordingly, in the present paper, a GA was chosen to be implemented due to being a well-documented algorithm in the literature for task/load scheduling, much faster when compared to more linear mathematical approaches, and being able to find solutions in large complex solution spaces. Other advantages of the GA are its execution time flexibility (less time can however result in worse solutions), crucial in uncertain environments such as shop floors, and its lower chance of getting stuck in a local optimum [29], [30].

An integrated flow shop production and preventive maintenance scheduling system using an adaptive local search Nondominated Sorting Genetic Algorithm II (NSGA-II) is proposed in [24]. The balance between production, maintenance activities and degradation is taken into account to minimize completion time and average idle time of machines, reducing production costs. Three algorithms are used for validation, adaptive local search NSGA-II, NSGA-II, and PSO, with the first cited algorithm outperforming the others in all performance metrics. The GA-implemented system in the work [25], focuses on a joint model for production and maintenance scheduling applied to a batch-deteriorating manufacturing system. It minimizes the total costs, while still complying with product demands, and degradation in the shop floor. The best maintenance strategy to mitigate degradation is found by following an imperfect preventive maintenance strategy. The proposed GA was compared to a SA algorithm, and even though the performance gap is small, the GA outperformed SA in all but one scenario. The paper in [26] also proposes a joint model for production and maintenance scheduling, as well as product quality control optimization for a serial-parallel multistage manufacturing system. It considers the usage and age of a machine as factors to determine a machine's condition, which affects product quality. As such, a constrained stochastic mathematical model is proposed to minimize the total costs by taking into account maintenance and quality control limits, as well as overall manufacturing time. The problem is solved using a GA and a Monte Carlo (MC) simulation-based approach. The obtained results of the cited work demonstrate that the combined maintenance strategy proposed outperforms monetarily more conventional maintenance strategies. Reference [27] proposes a scheduling system for joint production and maintenance optimization that focuses on minimizing makespan in a two-parallel machine environment. It achieves this by implementing and exploring three metaheuristics algorithms: GA, SA, and Tabu Search (TS). In the cited work, machine unavailability (i.e., interval-based preventative maintenance) and machine setup constraints are considered in the production schedule. Of the three metaheuristics algorithms, there was not a dominant algorithm during validation, each one being better in certain scenarios than in others. A real-time joint production and maintenance scheduling system is proposed in [28] for production and maintenance cost minimization as well as machine degradation optimization in FJSP manufacturing environments. To accomplish this, a real-time hybrid GA is implemented to solve a proactive-reactive optimization model. The proposed system was tested in real-time simulation scenarios, demonstrating cost savings averaging 27%, and up to 30% if the execution time of the algorithm was prolonged. A summary of the above-cited works, relating to the present paper's problem, is presented in Table 1.

B. CONTRIBUTIONS

The papers explored in the present paper's literature review (Section I-A) implement to some degree production and maintenance scheduling in manufacturing environments. However, they fail to further extend the complexity of the problem formulation to include more concepts that enable more reliable simulations of the real-world and efficient utilization of resources. For instance, the work in [24] is only adapted to flow shop manufacturing layouts and does not tackle costs directly, reducing its effectiveness in minimizing

TABLE 1. Summary of reviewed literature works.

Reference	Objective [a]	Method [b]	Application [c]
[20]	CM, PS, MS	PSO, SA, JA	FS, M
[21]	DM, PS, MS	LP	FS, M
[22]	CM, PS, MS	SA, GA	FS, M
[23]	CM, DM, PS, MS	RL	JSP, M
[24]	DM, PS, MS	NSGA-II	FS, M
[25]	CM, DM, PS, MS	GA	BP, M
[26]	CM, DM, PS, MS, PQ	GA, MC	SPMP, M
[27]	DM, PS, MS	GA, SA, TS	JSP, M
[28]	CM, DM, PS, MS	GA	FJSP, M

[a]CM = Cost Minimization, PS = Product Scheduling, MS = Maintenance Scheduling, DM = Degradation Minimization, PQ = Product Quality.
[b]JA = Jaya Algorithm.
[c]FS = Flow Shop, M = Manufacturing, JSP = Job Shop, BP = Batch Production, SPMP = Serial-Parallel Multistage Production.

costs. Furthermore, the cited works [25] and [26] have a basic approach to the problem formulation, by not considering energy price volatility, constraints imposed in the production schedule, and job shop layouts. In addition, while the work in [27] takes into account imposed constraints (i.e., machine unavailability and setup), it lacks in cost minimization and problem complexity by only considering two machines. The work in [28] addresses these concerns to some extent by considering cost and machine degradation minimization, constraints (e.g., product deadlines), and FJSP layouts, hence being the most similar work found to the present paper. Nevertheless, it again falls short in problem formulation by not taking into account energy price volatility as well as RER usage and surplus selling. When taking into account Table 1, it is clear that there is a lack of work that considers FJSP layouts, a manufacturing environment that is becoming evermore common.

Accordingly, the innovation of the present paper is to jointly optimize production and maintenance, by proposing a GA scheduling system that considers the combination of the following features:

- **Total cost minimization** – to reduce both energy and maintenance-derived costs;
- **Machine task overload minimization** – to improve machine longevity by reducing machine failure rate;
- **Flexible job shop layouts** – to accommodate more complex job shop manufacturing environments;
- **Energy price volatility** – to more reliably simulate retailer energy prices;
- **Renewable energy resources** – to further reduce energy costs by covering energy consumption with RERs and turn a profit with surplus energy;
- **Constraints** – to better simulate limitations in the production schedule.

The main innovation of the paper is to propose a GA capable of reliably scheduling tasks and maintenance activities in FJSP layouts, by proposing a sorted list crossover approach in the GA. The proposed crossover approach is more appropriate for FJSP layouts and constraints imposed in the schedule.

To validate the proposed system, a scenario from the work in the literature [31], containing real-production data, is considered as a baseline. It takes into account the scheduling of 275 tasks in three machines during a working week, with imposed constraints.

C. PAPER STRUCTURE

The paper is divided into six main sections. The first section presents an introductory segment that contextualizes the reader to the problem at hand, the current state-of-the-art works in the literature, and the corresponding contributions of the present paper. Section II addresses the proposed methodology, while Section III describes the implementation in detail. The case study is described in Section IV and its corresponding results and discussion are presented in Section V. The final section sums up the main conclusions of the present paper and discloses future work.

II. PROPOSED METHODOLOGY

In the proposed methodology, a joint production and maintenance scheduling system is implemented and explored to optimize the total costs (i.e., energy and maintenance costs) on the shop floor, as well as improve machine longevity by minimizing overload in single machines. This is accomplished through an intelligent AI scheduling system, employing a GA, to schedule tasks and maintenance activities, at the same time, in FJSP manufacturing layouts. On one hand, total cost minimization is achieved by taking advantage of retailer energy price volatility (e.g., during low energy price times, demand response participation), generated RERs (e.g., utilizing solar energy during its peak hours and selling the surplus to the electricity market), and by respecting maintenance hours (e.g., scheduling maintenance activities in maintenance hours, for lower maintenance costs). On the other hand, machine longevity is maximized by balancing tasks between the different machines on the shop floor, minimizing task overload in single machines. Furthermore, to more reliably simulate real-world scenarios, constraints imposed on the shop floor are considered (e.g., task order, task collision, order deadline, maintenance hours, and time transitions).

A. DOMAIN MODEL

There are eight concepts considered for the formulation of the problem: task, maintenance activity, machine, product, manufacturing order, energy source, energy buyer, and constraint. The domain model of the proposed methodology for production and maintenance optimization is represented in Fig. 1.

The described domain model concepts are:

- **Task** – activity to manufacture a product;
- **Maintenance activity** – activity to repair/inspect a machine;

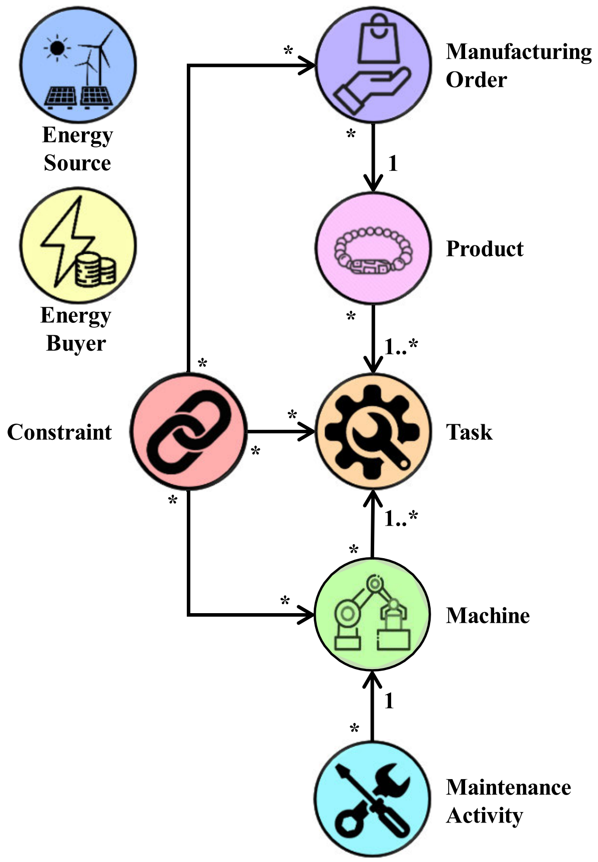

FIGURE 1. Domain model of the proposed methodology for joint production and maintenance scheduling.

- **Machine** – available manufacturing equipment, has a task compatibility (i.e., able to process) list;
- **Product**– task list to manufacture a specific commodity;
- **Manufacturing order** – product manufacturing request and its corresponding quantity;
- **Energy Source** – acquired energy that covers energy consumption from the shop floor, can represent paid energy (e.g., retailer or aggregator) or generated energy at no cost (e.g., RERs);
- **Energy Buyer** – energy buying surplus source (e.g., surplus RER for monetary compensation);

1) **Constraint** – production or maintenance limitation imposed on the production schedule. Five constraint types are considered:
   - *Task order* – sequence between two tasks (e.g., task β precedes task α);
   - *Task collision* – incompatible execution time between two tasks (e.g., task α and β cannot be executing at the same time);
   - *Order deadline* – completion time limit for a manufacturing order;
   - *Maintenance hours* – time intervals where maintenance activities are allowed (i.e., during maintenance hours), optionally with a monetary penalty for other times (i.e., outside maintenance hours);

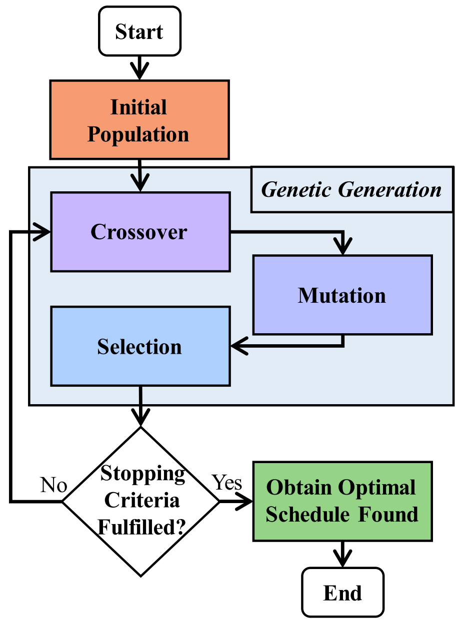

FIGURE 2. Flowchart of the Genetic Algorithm for joint production and maintenance scheduling.

   - *Time transitions* – production and maintenance un-schedulable times (e.g., the time between the shop floor closing and the next day's opening).

B. PROBLEM CONSIDERATIONS
To increase input flexibility in the proposed methodology, allowing for greater adaptability in different use cases, the following considerations are taken into account:

- **Time** – is represented in intervals of time (e.g., 30 in 30 seconds, 15 in 15 minutes, 1 in 1 hour) and labeled as periods, with a period corresponding to a unique interval of time. All periods have the same interval of time defined by the user in the input data. In addition, a period must be consistent throughout all the input data (i.e., production, energy, machine, maintenance, and price data);
- **Energy units** – are described as units of energy per period (e.g., Wh/period, kWh/period, MWh/period). The energy unit is defined by the user and needs to be consistent in all the input data.

III. GENETIC ALGORITHM IMPLEMENTATION
The proposed joint production and maintenance scheduling system is employed with a GA, described in Fig. 2, to find the optimal schedule that minimizes total costs and single-machine overload. It was developed and tested in the Python programming language.

The GA, as described in Fig. 2, can be divided into five main phases:

- **Initial population** – to generate a random population (i.e., set of GA solutions) composed of unique individuals (i.e., possible schedules) to initiate the GA;

- **Crossover** – to combine GA individual's characteristics (i.e., genetic information);
- **Mutation** – to insert diversity into the GA population, decreasing the chances of the GA getting stuck in a local optimum;
- **Selection** – to select which individuals inherit to the next genetic generation (i.e., procedure sequence crossover, mutation, and selection);
- **Obtain optimal schedule found** – to obtain, after a stopping criteria is fulfilled (e.g., GA execution time, number of completed genetic generations, fitness stagnation, or total cost reached) the best joint production and maintenance schedule found by the GA.

It is worth mentioning that because no available GA libraries allowed for the incorporation of the proposed constraints and were not adapted to FJSP manufacturing layouts, the GA implementation in Python was done without the use of any GA library.

<table>
<tr><th rowspan="2">Machine Identifier</th><th colspan="6">Period</th></tr>
<tr><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th></tr>
<tr><td>MAQ100</td><td>T5</td><td></td><td colspan="2">T7</td><td></td><td></td></tr>
<tr><td>MAQ200</td><td colspan="5">T4</td><td>M1</td></tr>
<tr><td>MAQ300</td><td>M2</td><td></td><td colspan="2">T3</td><td colspan="2">M3</td></tr>
<tr><td>MAQ400</td><td></td><td colspan="2">T1</td><td colspan="2">M4</td><td>T9</td></tr>
<tr><td>MAQ500</td><td>T6</td><td>T8</td><td colspan="3">T2</td><td></td></tr>
</table>

FIGURE 3. Genetic Algorithm individual example of a joint production and maintenance schedule ("T" and a numerical portray a unique task, "M" and a numerical describe a unique maintenance activity, and colors are to better differentiate tasks and maintenance activities).

A. INITIAL POPULATION

The initial population begins with the generation of random unique individuals and adding them to the GA population pool, according to the tasks and maintenance activities requested by the user in the scheduler's input. Population size is defined by the user in the input data as a GA optimization parameter. To maintain a unique population, duplicated individuals are removed from the population and substituted by another generated random individual that is unique. Moreover, if constraints are imposed on the production schedule and a generated schedule is invalid (i.e., does not respect all imposed constraints) the algorithm tries to repair the schedule, by shifting tasks and maintenance activities in order to comply with the constraints. However, if the repairing process fails, another random unique individual is generated and repaired if needed.

A GA individual describes a possible joint production and maintenance schedule, portrayed as a matrix with rows representing machine plans and columns specific periods, as shown in the example in Fig. 3. An individual can be easily navigated, to validate constraints or obtain task/maintenance information, by associating the x coordinate with the index of a machine plan and the y coordinate with a specific period.

B. CROSSOVER

Every genetic generation iteration starts with the crossover procedure, detailed in Fig. 4. If it is the first genetic generation iteration, then the crossover uses the initial population generation, otherwise, for subsequent iterations, it is used the resulting population from the previous genetic generation, as portrayed in Fig. 2.

After the corresponding population is obtained, individuals are randomly paired and applied with the crossover procedure. From an individual pair, two resulting new individuals can be obtained. For each pair, a list of tasks and maintenance activities is created, sorted by decreasing task processing time or maintenance expected time, and for equal times, increasing task/maintenance machine compatibility (maintenance activities always have the lowest compatibility due to only being associated with a machine in need of a repair/inspection). A new individual can be obtained by assigning the tasks and maintenance activities in order of the sorted list, according to the initial pair positions, following an alternative pattern. For instance, considering Fig. 4 as a crossover example starting with individual 1, following the sorted list, task "T4" is first assigned from individual 1 task position, then "T2" with individual 2, individual 1 again with "T7", maintenance "M3" with individual 2, and so on. As a result, one of the possible resulting individuals from a crossover pair can be obtained by beginning the assignment following the sorted list with individual 1 and the other with individual 2. In case an assignment is impossible (e.g., a task/maintenance is already assigned in that position) then the assignment is alternated to the other individual. However, if not possible, then it is assigned semi-randomly. In the worst-case scenario, where the task is impossible to assign (e.g., space or compatibility issues), the resulting individual is never added to the GA population pool. In addition, if a resulting individual is a duplicate, then it is also never added to the population.

The sorted list crossover approach was chosen instead of more conventional crossover procedures (e.g., cutting points or uniform crossover) because it is more appropriate for the problem at hand (i.e., FJSP layouts, individuals represented as matrixes, and constraints imposed). It allows tasks/maintenance activities with limited assignment flexibility to be prioritized above more flexible tasks/maintenance activities, reducing the chances of the crossover resulting in an invalid individual.

C. MUTATION

The mutation procedure utilizes the population obtained from the crossover to insert randomness into the population. To achieve this, each individual is assessed whether it is going to be subjected to a mutation or not, according to the probability defined by the user in the input data as a GA optimization parameter. For individuals that got selected

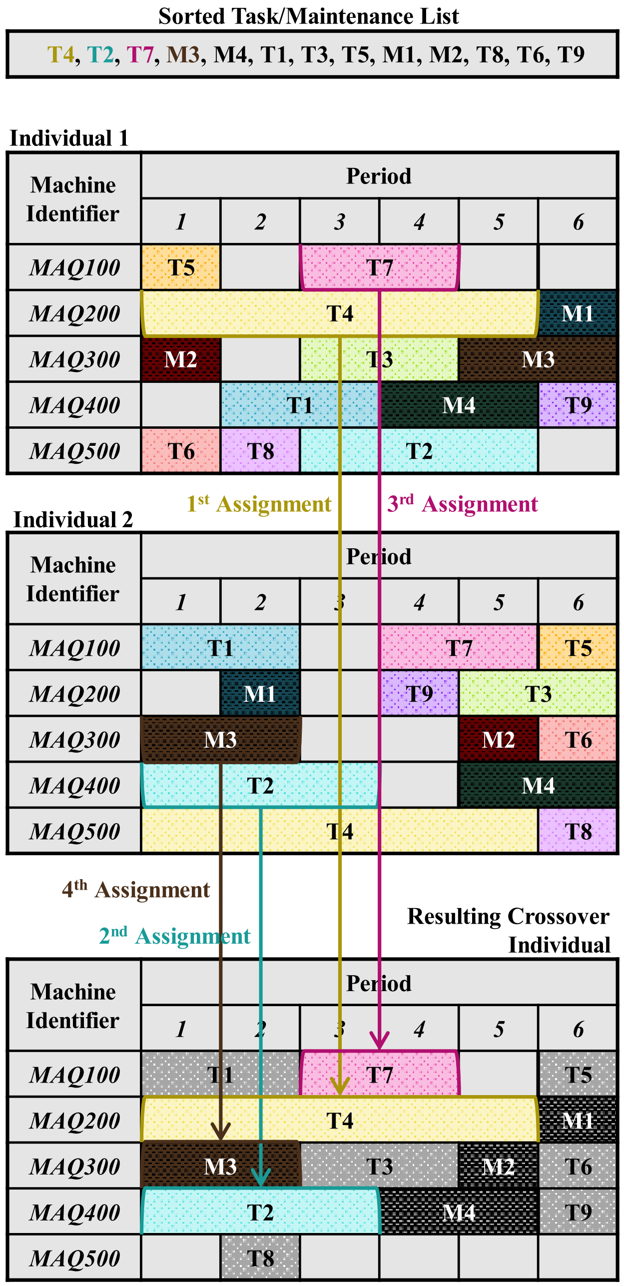

FIGURE 4. Genetic Algorithm crossover example for the first four assignments in the procedure, starting with individual 1 (other tasks/maintenance activities are greyed out to indicate their positions in the resulting individual if the crossover is completed).

to be mutated, a task/maintenance-swapping mutation is applied. It focuses on swapping the positions between a pair of tasks/maintenance activities or a task and maintenance activity. If the mutation creates an invalid individual (e.g., constraints are no longer being respected) or is impossible (e.g., swapped tasks are now in incompatible machines, or insufficient space for a swapped task/maintenance) the mutation is reversed, and another task/maintenance pair is swapped.

D. SELECTION

The final procedure in a genetic generation focuses on selecting the individuals that will inherit to the next genetic generation. Two different populations are combined into a single one to be subjected to the selection procedure: the original population from the current genetic generation iteration (i.e., no crossover and mutation procedures were applied) and the newly obtained population from the mutation procedure (i.e., was also applied with the crossover). In addition, in the combined population, duplicates are removed.

Afterward, each individual in the combined population is evaluated according to their Fitness Score (*FS*). GA individual *FS* is calculated following a multi-objective function that minimizes total costs, which includes energy and maintenance costs, and machine rate occupancy standard deviation, to balance tasks between the different machines to reduce overload in single machines.

After every individual is evaluated, a hybrid selection approach is used to select the individuals that inherit. An elite selection is first employed to select the individuals with the best *FS* (i.e., in the context of the minimization problem, the individuals that have the lowest *FS*). It ensures that the best individual(s) always inherit and no progress of the best schedule found is lost between each genetic generation iteration. The number of elite individuals is a GA optimization parameter defined by the user in the input data. Then, the remaining individuals that were not selected as elites, are subjected to compete in non-elite tournaments. These non-elite tournaments' purpose is to select individuals in a *FS* probability approach, allowing for worse individuals to at least have a chance of inheriting, crucial in reducing the chances of the GA getting stuck in a local optimum. First, individuals are randomly paired, then, each pair competes in a non-elite tournament, where the individual with the lowest *FS* (i.e., best fitness) has the highest probability of inheriting. Therefore the best individual in a non-elite tournament does not always inherit.

To obtain the *FS* of an individual, the total cost and machine overload equations are calculated separately and then combined in the final muti-objective equation through optimization weights.

1) TOTAL COST EQUATIONS

The total cost of a joint production and maintenance schedule can be obtained by using four equations: period energy consumption, period energy to pay, period maintenance to pay, and total cost.

The Period Energy Consumption (*PEC*), represented as $PEC_{Demand(p)}$ and calculated using (1), describes the energy consumed in a specific period p.

$$PEC_{Demand(p)} = \sum\nolimits_{m=1}^{M} E_{Demand(p,m)} \tag{1}$$

- p – specific period;
- m – machine index;
- M – total number of machines;

- $E_{Demand(p,m)}$ – energy consumption in period p in the machine of index m.

The Period Energy to Pay (PEP), portrayed as $PEP_{Demand(p)}$ and calculated using (2), represents the energy costs to pay in a specific period p.

$$PEP_{Demand(p)} = \begin{cases} 0, \textit{ if } \ E_{Generation(p)} = PEC_{Demand(p)} \\ \left(E_{Generation(p)} - PEC_{Demand(p)}\right) \times E_{Selling\ Price(p)}, \\ \textit{if } E_{Generation(p)} > PEC_{Demand(p)} \\ \left(PEC_{Demand(p)} - E_{Generation(p)}\right) \times E_{Buying\ Price(p)}, \\ \textit{if } E_{Generation(p)} < PEC_{Demand(p)} \end{cases} \tag{2}$$

- $E_{Generation(p)}$ – available generated energy from RERs in period p;
- $E_{SellingPrice(p)}$ – selling price of energy in period p;
- $E_{BuyingPrice(p)}$ – buying price of energy in period p.

The Period Maintenance to Pay (PMP), depicted as $PMP_{Maintenance(p)}$ and calculated using (3), portrays the maintenance-derived costs to pay in a specific period p. Maintenance activities can be done during a defined period interval, that is, during maintenance hours (e.g., can describe when maintenance workers are available) or outside of that period, and as such outside of maintenance hours (e.g., can represent when maintenance workers need to be paid more for the extra working hours).

$$PMP_{Maintenance(p)} = \begin{cases} 0, \ \textit{if no maintenace} \\ M_{During\ Period\ Price(p)}, \\ \textit{if in maintenance hours} \\ M_{Outside\ Period\ Price(p)}, \\ \textit{if out maintenance hours} \end{cases} \tag{3}$$

- $M_{During\ Period\ Price(p)}$ – maintenance price if done during maintenance hours;
- $M_{Outside\ Period\ Price(p)}$ – maintenance price if done outside of maintenance hours (i.e., can represent a monetary penalty).

The Total Cost (TC) is calculated using (4), which describes the total cost in a schedule (i.e., GA individual). It is noteworthy that, variable $PEP_{Demand(p)}$ already includes the energy costs from all the machines within period p, however, for variable $PMP_{Maintenance(p)}$ this is not true, thus the inclusion of the summation operator.

$$TC = \sum\nolimits_{p=1}^{P} \left(\begin{array}{l} PEP_{Demand(p)} + \\ \left(\sum\nolimits_{m=1}^{M} PMP_{Maintenance(p)}\right) \end{array} \right) \tag{4}$$

- P – total number of periods in the schedule (i.e., time window).

2) MACHINE OVERLOAD EQUATIONS

The machine overload status (i.e., level of task balancing between machines) of a joint production and maintenance schedule can be obtained by employing the machine occupation rate standard deviation. To obtain it, three equations are needed: machine degradation classifier, machine occupation rate, and occupation standard deviation.

The Machine Degradation Classifier (MDC), represented as $MDC_{Factor(p,m)}$ and calculated using (5), classifies if the factor in period p and machine m contributes to the degradation of a machine. For tasks, which are considered to contribute to machine degradation in the problem at hand, their classification is 1 (i.e., true). However, maintenance activities and empty spaces do not contribute to machine degradation, thus they are classified as 0 (i.e., false).

$$MDC_{Factor(p,m)} = \begin{cases} 1, & \textit{if task} \\ 0, & \textit{if maintenance or empty space} \end{cases} \tag{5}$$

The Machine Occupation Rate (MOR), depicted as $MOR_{Factors(m)}$ and calculated using (6), represents the machine occupation rate of factors that contribute to the degradation of a machine of index m.

$$MOR_{Factors(m)} = \frac{\sum_{p=1}^{P} MDC_{Factor(p,m)}}{P} \tag{6}$$

The Occupation Standard Deviation (OSD) is calculated using (7), which describes the population standard deviation (i.e., not sample deviation) of the machine occupation rates in a schedule.

$$OSD = \sqrt{\frac{\sum_{m=1}^{M} \left(MOR_{Factors(m)} - \left(\frac{\sum_{m=1}^{M} MOR_{Factors(m)}}{M} \right)^2 \right)}{M}} \tag{7}$$

3) FITNESS EQUATION

Before the FS is calculated for each individual, TC, from (4), and OSD, from (7), are applied with a Min-Max normalization method using the obtained results from the individuals in the GA population. Therefore, all individuals need to have their TC and OSD evaluated in order to be able to apply the Min-Max method. After obtaining the normalized values, each individual has their FS calculated according to (8), which describes the fitness score of a GA individual.

$$FS = TC_{Norm} \times W_{TC} + OSD_{Norm} \times W_{OSD} \tag{8}$$

- TC_{Norm} – Min-Max normalized TC value;
- OSD_{Norm} – Min-Max normalized OSD value;
- W_{TC} – optimization weight of TC;
- W_{OSD} – optimization weight of OSD.

Variables W_{TC} and W_{OSD} assume a value between 0 (i.e., 0%) and 1 (i.e., 100%), inclusive. In addition, the sum of W_{TC} and W_{OSD} is always 1.

E. OBTAIN OPTIMAL SCHEDULE FOUND

After a genetic generation ends, with the selection procedure, the algorithm validates if at least one stopping criteria has been fulfilled, as represented in Fig.2. There are four stopping criteria available, for the user to define, in the proposed GA: algorithm execution time in seconds, number of completed genetic generations iterations, number of consecutive genetic generations with fitness stagnation, and minimum total overall cost reached. In case at least one of the four stopping criteria is fulfilled, the algorithm stops iterating genetic generations, and the schedule with the best *FS* found by the GA is obtained from the last genetic generation iteration, and provided to the user as the optimal schedule found by the GA. However, if no stopping criteria is fulfilled, then a new genetic generation is initiated with the individuals that inherited from the selection procedure, beginning another procedure sequence crossover, mutation, and selection. It is worth mentioning that, the minimum total cost stopping criteria is fulfilled when the *TC* of an individual, obtained through (4), is equal to or lower than the value defined by the user as the minimum total cost.

IV. CASE STUDY

A baseline scenario from the literature available in [32], from the work in [31], is used the validate the present paper's methodology. The baseline scenario uses real production data from a textile company that manufactures hang tags, ideal for evaluating the present methodology in the real world. It focuses on the minimization of energy costs, achieving a total energy cost of 36.42 EUR. However, it lacks in expanding the total cost optimization to also include surplus energy selling, for further energy cost minimization, and the consideration of maintenance activities that need to be properly scheduled, in order to minimize maintenance costs. Moreover, in the baseline scenario, no machine overload optimization, to improve machine longevity, is done. Accordingly, the purpose of the present paper is to expand on the concepts of the baseline scenario and address the abovementioned issues. In addition, the baseline scenario also includes the same constraints available in the present paper, allowing the evaluation of the current paper's methodology performance when handling constraints.

From the baseline scenario in [32], the following data are considered:

- **A working week** – from Monday to Saturday, each day from 7:00 a.m. to 11:00 p.m.;
- **275 tasks** – to be scheduled, from 132 manufacturing orders of 14 different products;
- **Three available machines** – with different task compatibilities and in a FJSP layout are considered for manufacturing: "MAQ118", "MAQ119", and "MAQ120";
- **5 minute-periods** – are considered for all the production, energy, machine, maintenance, and price data. As such, there is a schedule time window of 1152 periods (i.e., 192 periods per day) resulting in a GA individual being a matrix of 3 $\times$ 1152 (i.e., 3 machines $\times$ 1152 periods);
- **Wh/period energy units**– are considered for all the energy data;
- **Four constraints**– are considered:
  - *Task order* – tasks "Harden [2]" precede tasks "Harden [1.5]";
  - *Task collision* – tasks "Harden [2]" and "Sublimation" cannot be executing at the same time;
  - *Order deadline* – one manufacturing order of "Elastic w/ inscr" has a completion time limit of 11:00 p.m. Friday;
  - *Time transitions* – between each day, the transition of the current day at 11 p.m. to the next day at 7 a.m.
- **MIBEL (Iberian Electricity Market) buying prices**– from [33] are considered from the 7th to the 12th of January of 2019;
- **Local generated photovoltaic energy**– is considered from a 3 kW peak solar panel in Portugal, from the 6th to the 11th of June 2020;

V. RESULTS AND DISCUSSION

A new scenario based on the baseline scenario [32], from Section IV, is proposed to evaluate the proposed system capabilities in also including RER surplus selling, maintenance scheduling, and single-machine overload reduction. The proposed new scenario is a more complex adaptation of a scenario already available in [34] from the work [8], which includes more maintenance activities to be scheduled, different GA optimization parameters, and cheaper surplus energy buyers. Moreover, maintenance data was partially obtained from the work in [35].

In the new proposed scenario, in addition to what is currently in the baseline scenario, the following data is considered:

- **Energy selling** – corresponding to 30% of the energy price for buying
- **Eight maintenance activities** – to be scheduled along with production, having a labor cost of 3,22 EUR/hour during maintenance hours, and a monetary penalty of 6.44 EUR/hour if done outside of maintenance hours. The considered maintenance activities are:
  - *One 4-hour maintenance in "MAQ118"* – to be scheduled at any time;
  - *One 2-hour maintenance in "MAQ118"* – to be scheduled on Thursday;
  - *Two 1-hour maintenances in "MAQ119"* – one to be scheduled from Monday to Wednesday, and another from Thursday to Saturday;
  - *One 1-hour maintenance in "MAQ120"* – to be scheduled from Monday to Tuesday;
  - *One 30-minute weekly maintenance inspection for each machine* – to be scheduled at any time;

TABLE 2. Baseline and proposed scenarios' total cost with/without maintenance costs, and machine occupancy rate standard deviation (each execution of the proposed scenario is represented by "proposed" plus a numerical).

Scenario	*TC* [a] in EUR (change from baseline in %)	TCWM [b] in EUR (change from baseline in %)	*OSD* [c] in % (change from baseline in %)
Baseline	36.42 (0%)	36.42 (0%)	0.0997 (0%)
Proposed "1"	65.93 (+81.0%)	32.06 (-12.0%)	0.0918 (-7.9%)
Proposed "2"	67.74 (+86.0%)	33.87 (-7.0%)	0.0472 (-52.7%)
Proposed "3"	67.30 (+84.8%)	32.89 (-9.7%)	0.0479 (-52.0%)
Proposed "4"	64.69 (+77.6%)	30.82 (-15.4%)	0.0750 (-24.8%)
Proposed "5"	65.68 (+80.3%)	31.81 (-12.7%)	0.0756 (-24.2%)
Proposed "Average"	66.27 (+81.9%)	32.29 (-11.3%)	0.0675 (-32.3%)

[a]TC = Total Cost (EUR).
[b]TCWM = Total Cost Without Maintenance associated costs (EUR).
[c]OSD = machine Occupancy rate Standard Deviation (%).

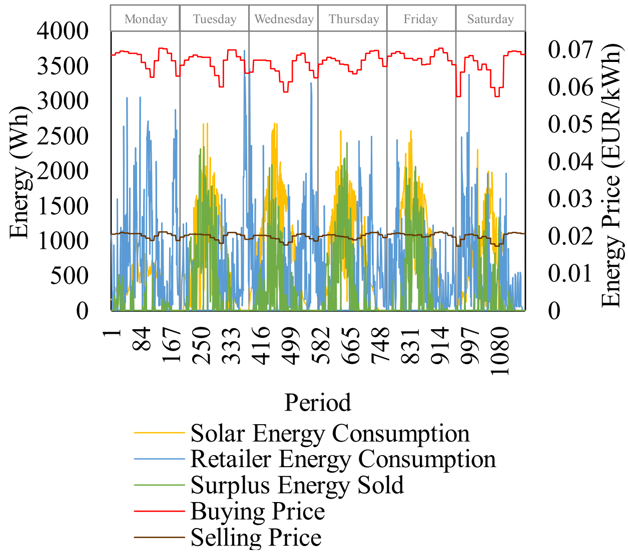

FIGURE 5. Energy consumption, energy bought and sold, as well as buying and selling prices for the fourth execution of the proposed scenario (Proposed "4").

- **Single-machine overload reduction**– corresponding to a 30% weight in the GA multi-objective function (70% weight for total cost).
- **Genetic algorithm optimization parameters**– 20 individuals in the GA population, 3 elite individuals, and a mutation rate of 5% (both crossover and mutation procedures are used in the GA).

It is worth noting that, while there are eight maintenance activities, this is not a common working week, only the mandatory three short-maintenance activities for inspection are common, the remainder maintenance activities are to better demonstrate the system's capabilities. Also, the chosen GA multi-objective weights allow to better illustrate the system's ability in reducing single-machine overload while still maintaining good levels of total cost minimization.

The proposed scenario was executed for about 2 hours, the same as the baseline scenario, on an AMD®Ryzen 7 3700X processor 4.05 GHz, 32 GB of RAM, and Windows 11 Home version 22H2.

The new proposed scenario was executed five times in the proposed joint production and maintenance scheduling system in order to evaluate the GA's reliability and efficiency. Accordingly, in Table 2, the obtained results from the five executions (i.e., portrayed by "Proposed" plus a numerical) are compared to the baseline scenario regarding the total costs, total costs without maintenance costs (i.e., for a more fair comparison with the baseline scenario), and the machine occupancy rate standard deviations. Table 2 demonstrates that the proposed GA can reliably provide well-optimized production and maintenance schedules, with the difference from the highest and lowest *TC* being 4.5% (i.e., 67.74 and 64.69 EUR, respectively) and 48.6% for the *OSD* (i.e., 0.0918 and 0.0472 %). It is worth mentioning that, while the *OSD* is much more volatile, with its 48.6% difference, this can be a result of its optimization weight being so low when compared to the *TC* (i.e., 30% for *OSD* and 70% for *TC*), hence being disproportionally impacted negatively but still an improvement from the baseline scenario. Furthermore, from Table 2, it is shown that the system was capable of minimizing costs up to 15.4% from the baseline scenario, when not considering maintenance costs, and up to 52.7% in single-machine overload minimization. Overall results are very promising, with an average improvement of 11.3% in total costs without the consideration of maintenance costs and 32.3% in machine occupancy rates. Also, while maintenance costs are ignored for comparison with the baseline scenario, these activities still occupy a large amount of space that could be used by tasks to further reduce energy costs and balance occupancy rates, thus it demonstrates the proposed system's capabilities in reducing costs even with maintenance activities.

From Table 2, the authors consider that the best production and maintenance schedule found of the proposed scenario was provided from the fourth execution (i.e., Proposed "4") since it offers the highest cost reductions and the third-best improvement in machine occupancy rates. As such, the energy consumption, energy bought and surplus sold, as well as buying and selling prices for the Proposed "4" are described in Fig. 5.

It shows that the proposed system uses RERs (i.e., solar energy) as much as possible both to cover the energy consumption and to profit from the surplus, particularly during midday when solar energy is vastly available. Also worth mentioning, is the low energy consumption during high energy buying price periods and high consumption in cheaper periods (e.g., the retailer energy consumption peak on Tuesday at the end of the day, where prices are cheaper),

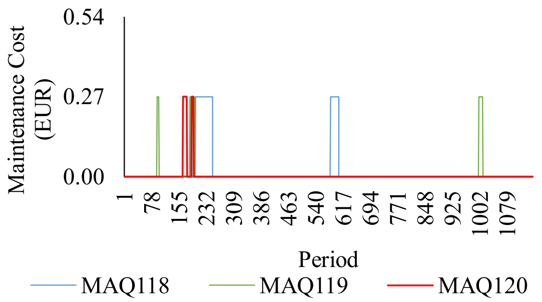

FIGURE 6. Maintenance cost per machine for the fourth execution of the proposed scenario (Proposed "4").

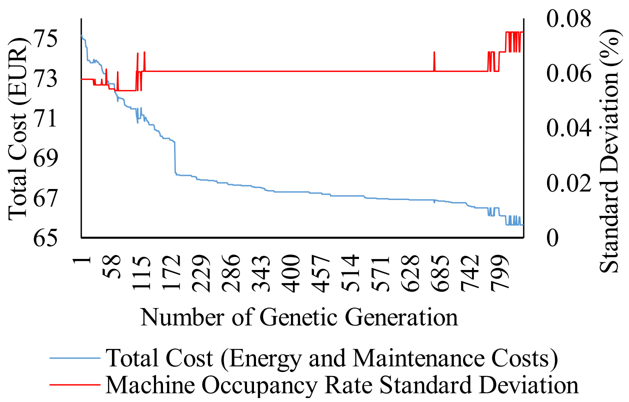

FIGURE 7. Total cost and machine occupancy rate standard deviation per genetic generation of the Genetic Algorithm for the fourth execution of the proposed scenario (Proposed "4").

demonstrating that the proposed system is able to effectively shift tasks to adapt to changes in energy prices.

The maintenance costs of each machine for the Proposed "4" scenario are represented in Fig. 6.

It indicates that all maintenance activities that had stipulated maintenance hours, were done in these periods, because all maintenance activities had a cost of about 0.27 EUR/period (i.e., 3,22 EUR/hour). Therefore, there was no monetary penalty of about 0.54 EUR/period (i.e., 6.44 EUR/hour) applied to any maintenance activity, illustrating the proposed system's capabilities in efficiently scheduling maintenance activities along with tasks.

The total cost (i.e., *TC*) and machine occupancy rate standard deviation (i.e., *OSD*) evolution throughout each genetic generation in the GA is portrayed in Fig. 7. The balance between *TC* and *OSD* is demonstrated in Fig. 7 by the values spiking in opposite directions, for instance, when the *OSD* greatly increases the cost tends to decrease. Moreover, while the evolution of *OSD* tends to be negative, which is a result of having less optimization weight in the GA, the *TC* shows good evolution performance, and with more time, costs could be further reduced. Also worth noting, is the massive decrease in *TC* around the 172 genetic generation, which could have resulted from a maintenance activity being shifted to maintenance hours, decreasing the *TC* greatly.

VI. CONCLUSION

Recent crises such as the COVID-19 pandemic and the Russo-Ukrainian conflict have significantly increased energy prices in the European continent, leaving many European manufacturing companies less competitive in the global market. The use of generated RERs such as solar energy by companies can be a great alternative not only to reduce dependability on external retailers, mitigating the impact of high energy prices, but also to get additional profits by participating in the electricity market to sell surplus RERs. Similarly, maintenance costs are also an important issue to address since they can cost up to 70% of the cost of the product. As such, maintenance activities need to be carefully planned with production to optimize energy and maintenance costs, and at the same time, machine degradation needs to be minimized to reduce the need for additional maintenance activities.

The purpose of the present paper is to address these issues by implementing and exploring a scheduling system for joint production and maintenance optimization in FJSP manufacturing layouts, by using a GA. It considers retailer energy price volatility, generated RER availability, RER surplus selling, maintenance stipulated hours, and constraints applied in the production schedule to optimize production and maintenance in manufacturing environments. Furthermore, it also takes into account task balancing between different machines to reduce overload in single machines, improving longevity in machines in the long run. A multi-objective function is considered in the GA to minimize both the energy and maintenance costs (i.e., total costs) as well as the machine occupancy rate standard deviation (i.e., machine overload). Moreover, in the GA, an uncommon crossover approach is considered and implemented due to the problem at hand consisting of FJSP layouts, GA individuals matrix, and imposed constraints in the production schedule.

To validate the proposed system a scenario from the literature that incorporates real-production data is used as a baseline. It considers the scheduling of 275 tasks among three machines throughout a working week, subject to constraints. In addition, to evaluate the reliability of the proposed GA, the scenario was executed five different times. Results highlight the effectiveness of the proposed system in scheduling both tasks and maintenance activities, as energy costs can be reduced by up to 15.4% and on average by 11.3%, and single-machine overloads by up to 52.7% with an average of 32.3%. Furthermore, maintenance costs were minimized by complying with stipulated maintenance hours, negating any type of monetary penalty. Finally, results also demonstrate the system's capabilities in intelligently balancing total costs with machine occupancy rate standard deviations.

In future work, degradation values will be considered for each machine. Accordingly, product quality will be implemented and explored into the GA multi-objective function, and associated with the machine degradation values.

REFERENCES

[1] S. Halbrügge, P. Schott, M. Weibelzahl, H. U. Buhl, G. Fridgen, and M. Schöpf, "How did the German and other European electricity systems react to the COVID-19 pandemic?" *Appl. Energy*, vol. 285, Mar. 2021, Art. no. 116370, doi: 10.1016/j.apenergy.2020.116370.

[2] L. Cui, S. Yue, X.-H. Nghiem, and M. Duan, "Exploring the risk and economic vulnerability of global energy supply chain interruption in the context of Russo-Ukrainian war," *Resour. Policy*, vol. 81, Mar. 2023, Art. no. 103373, doi: 10.1016/j.resourpol.2023.103373.

[3] J. S. González and C. Á. Alonso, "Industrial electricity prices in Spain: A discussion in the context of the European internal energy market," *Energy Policy*, vol. 148, Jan. 2021, Art. no. 111930, doi: 10.1016/j.enpol.2020.111930.

[4] B. Wormuth, S. Wang, P. Dehghanian, M. Barati, A. Estebsari, T. P. Filomena, M. H. Kapourchali, and M. A. Lejeune, "Electric power grids under high-absenteeism pandemics: History, context, response, and opportunities," *IEEE Access*, vol. 8, pp. 215727–215747, 2020, doi: 10.1109/ACCESS.2020.3041247.

[5] D. Carmon, A. Navon, R. Machlev, J. Belikov, and Y. Levron, "Readiness of small energy markets and electric power grids to global health crises: Lessons from the COVID-19 pandemic," *IEEE Access*, vol. 8, pp. 127234–127243, 2020, doi: 10.1109/ACCESS.2020.3008929.

[6] B. Canizes, B. Mota, P. Ribeiro, and Z. Vale, "Demand response driven by distribution network voltage limit violation: A genetic algorithm approach for load shifting," *IEEE Access*, vol. 10, pp. 62183–62193, 2022, doi: 10.1109/ACCESS.2022.3182580.

[7] H. T. Dinh, J. Yun, D. M. Kim, K. Lee, and D. Kim, "A home energy management system with renewable energy and energy storage utilizing main grid and electricity selling," *IEEE Access*, vol. 8, pp. 49436–49450, 2020, doi: 10.1109/ACCESS.2020.2979189.

[8] B. Mota, P. Faria, B. Canizes, and C. Ramos, "Production and maintenance scheduling for total cost and machine longevity optimization," in *Proc. Int. Conf. Future Energy Solutions (FES)*, Jun. 2023, pp. 1–6, doi: 10.1109/FES57669.2023.10183219.

[9] Y. Fernando and W. L. Hor, "Impacts of energy management practices on energy efficiency and carbon emissions reduction: A survey of Malaysian manufacturing firms," *Resour., Conservation Recycling*, vol. 126, pp. 62–73, Nov. 2017, doi: 10.1016/j.resconrec.2017.07.023.

[10] P. Saxena, P. Stavropoulos, J. Kechagias, and K. Salonitis, "Sustainability assessment for manufacturing operations," *Energies*, vol. 13, no. 11, p. 2730, May 2020, doi: 10.3390/en13112730.

[11] B. Mota, P. Faria, and Z. Vale, "Residential load shifting in demand response events for bill reduction using a genetic algorithm," *Energy*, vol. 260, Dec. 2022, Art. no. 124978, doi: 10.1016/J.ENERGY.2022.124978.

[12] L. Xiao, S. Song, X. Chen, and D. W. Coit, "Joint optimization of production scheduling and machine group preventive maintenance," *Rel. Eng. Syst. Saf.*, vol. 146, pp. 68–78, Feb. 2016, doi: 10.1016/j.ress.2015.10.013.

[13] M. S. Alvarez-Alvarado, D. L. Donaldson, A. A. Recalde, H. H. Noriega, Z. A. Khan, W. Velasquez, and C. D. Rodríguez-Gallegos, "Power system reliability and maintenance evolution: A critical review and future perspectives," *IEEE Access*, vol. 10, pp. 51922–51950, 2022, doi: 10.1109/ACCESS.2022.3172697.

[14] D. S. Thomas, "The costs and benefits of advanced maintenance in manufacturing," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. NIST AMS 100-18, 2018, pp. 1–45, doi: 10.6028/nist.ams.100-18.

[15] Q. Liu, M. Dong, and F. F. Chen, "Single-machine-based joint optimization of predictive maintenance planning and production scheduling," *Robot. Comput.-Integr. Manuf.*, vol. 51, pp. 238–247, Jun. 2018, doi: 10.1016/j.rcim.2018.01.002.

[16] D. Belyi, E. Popova, D. P. Morton, and P. Damien, "Bayesian failure-rate modeling and preventive maintenance optimization," *Eur. J. Oper. Res.*, vol. 262, no. 3, pp. 1085–1093, Nov. 2017, doi: 10.1016/j.ejor.2017.04.019.

[17] T. Liu, Y. Chen, and J. Chou, "Solving distributed and flexible job-shop scheduling problems for a real-world fastener manufacturer," *IEEE Access*, vol. 2, pp. 1598–1606, 2014, doi: 10.1109/ACCESS.2015.2388486.

[18] J. Xie, L. Gao, K. Peng, X. Li, and H. Li, "Review on flexible job shop scheduling," *IET Collaborative Intell. Manuf.*, vol. 1, no. 3, pp. 67–77, Sep. 2019, doi: 10.1049/IET-CIM.2018.0009.

[19] H. Zhong, J. Liu, Q. Chen, N. Mao, and X. Yang, "Performance assessment of dynamic flexible assembly job shop control methods," *IEEE Access*, vol. 8, pp. 226042–226058, 2020, doi: 10.1109/ACCESS.2020.3043880.

[20] A. K. Mishra, D. Shrivastava, D. Tarasia, and A. Rahim, "Joint optimization of production scheduling and group preventive maintenance planning in multi-machine systems," *Ann. Oper. Res.*, vol. 316, pp. 1–44, Dec. 2021, doi: 10.1007/s10479-021-04362-z.

[21] Z. Zhang and Q. Tang, "Integrating flexible preventive maintenance activities into two-stage assembly flow shop scheduling with multiple assembly machines," *Comput. Ind. Eng.*, vol. 159, Sep. 2021, Art. no. 107493, doi: 10.1016/j.cie.2021.107493.

[22] H. Feng, L. Xi, L. Xiao, T. Xia, and E. Pan, "Imperfect preventive maintenance optimization for flexible flowshop manufacturing cells considering sequence-dependent group scheduling," *Rel. Eng. Syst. Saf.*, vol. 176, pp. 218–229, Aug. 2018, doi: 10.1016/j.ress.2018.04.004.

[23] H. Yang, W. Li, and B. Wang, "Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning," *Rel. Eng. Syst. Saf.*, vol. 214, Oct. 2021, Art. no. 107713, doi: 10.1016/j.ress.2021.107713.

[24] Y. Liu, X. Wang, Y. Zhang, and L. Liu, "An integrated flow shop scheduling problem of preventive maintenance and degradation with an improved NSGA-II algorithm," *IEEE Access*, vol. 11, pp. 3525–3544, 2023, doi: 10.1109/ACCESS.2023.3234428.

[25] L. Wang, Z. Lu, and X. Han, "Joint optimisation of production, maintenance and quality for batch production system subject to varying operational conditions," *Int. J. Prod. Res.*, vol. 57, no. 24, pp. 7552–7566, Dec. 2019, doi: 10.1080/00207543.2019.1581956.

[26] G. Cheng and L. Li, "Joint optimization of production, quality control and maintenance for serial-parallel multistage production systems," *Rel. Eng. Syst. Saf.*, vol. 204, Dec. 2020, Art. no. 107146, doi: 10.1016/j.ress.2020.107146.

[27] L. Hidri, K. Alqahtani, A. Gazdar, and A. Badwelan, "Integrated scheduling of tasks and preventive maintenance periods in a parallel machine environment with single robot server," *IEEE Access*, vol. 9, pp. 74454–74470, 2021, doi: 10.1109/ACCESS.2021.3081495.

[28] M. Ghaleb, S. Taghipour, and H. Zolfagharinia, "Real-time integrated production-scheduling and maintenance-planning in a flexible job shop with machine deterioration and condition-based maintenance," *J. Manuf. Syst.*, vol. 61, pp. 423–449, Oct. 2021, doi: 10.1016/j.jmsy.2021.09.018.

[29] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: Past, present, and future," *Multimedia Tools Appl.*, vol. 80, no. 5, pp. 8091–8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[30] A. Lambora, K. Gupta, and K. Chopra, "Genetic algorithm—A literature review," in *Proc. Int. Conf. Mach. Learn., Big Data, Cloud Parallel Comput. (COMITCon)*. Piscataway, NJ, USA: Institute of Electrical and Electronics Engineers, Feb. 2019, pp. 380–384, doi: 10.1109/COMITCon.2019.8862255.

[31] B. Mota, L. Gomes, P. Faria, C. Ramos, Z. Vale, and R. Correia, "Production line optimization to minimize energy cost and participate in demand response events," *Energies*, vol. 14, no. 2, p. 462, Jan. 2021, doi: 10.3390/en14020462.

[32] B. Mota, L. Gomes, P. Faria, C. Ramos, and Z. Vale, "Production line dataset for task scheduling and energy optimization—Schedule optimization," Zenodo, Nov. 2022, doi: 10.5281/zenodo.4106746.

[33] *Mibel—Mercado Ibérico de Electricidade*. Accessed: Jan. 23, 2022. [Online]. Available: https://www.mibel.com/

[34] B. Mota, "Joint optimization of production and maintenance for cost-effective manufacturing and demand response participation dataset—Joint optimization of production and maintenance," Zenodo, 2022, doi: 10.5281/zenodo.7060462.

[35] I. H. Afefy, "Reliability-centered maintenance methodology and application: A case study," *Engineering*, vol. 2, no. 11, pp. 863–873, 2010, doi: 10.4236/eng.2010.211109.

BRUNO MOTA received the master's degree in informatics from the School of Engineering, Polytechnic Institute of Porto, in 2022. He has participated in innovation projects at the European level with a focus on the technology transfer from academia to SMEs. His research interests include genetic algorithms applied to industrial facility planning and operation with special concern on energy aspects, namely, in interaction with the smart grid.

PEDRO FARIA (Member, IEEE) received the B.Sc. and M.Sc. degrees in electrical engineering from the Polytechnic Institute of Porto, Portugal, in 2008 and 2011, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Trás-os-Montes and Alto Douro, Vila Real, Portugal, in 2016. He is currently a Researcher with GECAD, Polytechnic Institute of Porto. He is a member of LASI, the Portuguese AI Associated Laboratory. His research interests include demand response, smart grids, electricity markets, and real-time simulation.

CARLOS RAMOS (Senior Member, IEEE) received the B.Sc. and Ph.D. degrees in electrical engineering and computers and the Habilitation (Agregação) degree from the University of Porto, Porto, Portugal, in 1986, 1993, and 2001, respectively. From 2010 to 2018, he was the Vice President of the Polytechnic Institute of Porto, where he has been the Pro-President since April 2022. He was the Founder and a former Director of the Research Group on Intelligent Engineering and Computing for Advanced Innovation and Development (GECAD), a Research and Development Unit ranked as Excellent by the Portuguese Science and Technology Foundation. He is currently a Full Professor with the Department of Informatics, Institute of Engineering, Polytechnic Institute of Porto. He is also a member of LASI, the Portuguese AI Associated Laboratory. He is also the Director of the Master of Artificial Intelligence Engineering, Polytechnic Institute of Porto. His research and development is centered on artificial intelligence. He has around 500 scientific publications, participated in 58 international and national research and development projects, and coordinated 25 and supervised 13 Ph.D. works. His research interests include AI planning, knowledge-based systems, decision support systems, machine learning, ambient intelligence, industry 4.0, power and energy systems, smart homes, smart cities, and tourism.