Production Engineering (2022) 16:65–88
https://doi.org/10.1007/s11740-021-01064-0

MECHANICAL ENGINEERING

Predictive maintenance integrated production scheduling by applying deep generative prognostics models: approach, formulation and solution

Simon Zhai[1] · **Meltem Göksu Kandemir**[1] · **Gunther Reinhart**[1]

Received: 11 April 2021 / Accepted: 8 June 2021 / Published online: 29 July 2021

Abstract
To harness the full potential of predictive maintenance (PdM), PdM information has to be used to optimally plan production and maintenance actions. Hence, operation-specific modelling of degradation, i.e. predictions of the health condition under time-varying operational conditions, has to be realized. By utilizing operation-specific degradation information, maintenance and production can be planned with regard to each other and thus, predictive maintenance integrated production scheduling (PdM-IPS) is enabled. This publication proposes a novel PdM-IPS approach consisting of two interacting modules: an operation-specific Prognostics and Health Management (PHM) module and an integrated production scheduling and maintenance planning (IPSMP) module. Specifically, the mathematical problem of the IPSMP module based on an extended version of the maintenance integrated flexible job shop problem is formulated. A two-stage genetic algorithm to efficiently solve this problem is designed and subsequently applied to simulated condition monitoring, as well as real industrial data. Results indicate that the approach is able to find feasible high quality PdM integrated production schedules.

Keywords Predictive maintenance · Integrated scheduling · Genetic algorithm · Decision support

1 Introduction

Predictive maintenance (PdM) is one of the core Industry 4.0 innovations that substantially increases machine availability by preventing machine failure and enabling maintenance actions *just in time*. Using advanced analytics on condition monitoring (CM) data collected by powerful sensors, the current health condition and remaining useful life (RUL) of machines can be predicted. This enables the introduction of new maintenance strategies. Recent studies indicate that while industrial companies do acknowledge the upside potential of PdM, this technology still faces challenges regarding its value-adding implementation in industry [1]. RUL estimation alone does not lead to value-added, decision support does [2]. One crucial missing link to generate value-added is the inability of most existing PdM models to account for varying operational conditions, i.e. to estimate operation-specific degradation [3]. On the one hand, operation-specific predictions are needed for subsequent scheduling of production and maintenance since different products and their required operations induce different stresses on the machine. On the other hand, existing scheduling algorithms are not designed to jointly optimize production scheduling and maintenance planning using PdM information. Thus, the focus of this publication is the mathematical formulation of the integrated production scheduling and maintenance planning (IPSMP) problem, its interactions with the Prognostics and Health Management (PHM) module developed by Zhai et al. [4] and the subsequent solution of the problem using a novel two-stage genetic algorithm (TSGA) that is specifically designed for interacting with PHM modules.

This paper is structured as follows: Sect. 2 gives an overview of related work and derives the research objective of this publication. Section 3 introduces the concept of the PdM integrated production scheduling (PdM-IPS) approach. Its

✉ Simon Zhai
simon.zhai@iwb.tum.de

Meltem Göksu Kandemir
meltemgoksu.kandemir@tum.de

Gunther Reinhart
emeritus.reinhart@tum.de

[1] Department of Mechanical Engineering, Institute for Machine Tools and Industrial Management, Technical University of Munich, Garching, Germany

mathematical model is presented in Sect. 4, while Sect. 5 proposes a two-stage Genetic Algorithm for its optimization and solution. Computational results applying the approach to benchmark data, as well as real industrial data, are presented in Sect. 7. Finally, Sect. 8 concludes the main findings of this paper.

2 State of the art and research objective

A detailed literature study was carried out to shed light on the current advances in IPSMP, as well as to highlight existing shortcomings to derive research objectives. The entity of IPSMP approaches can be divided into the categories *time-based* and *condition-based* maintenance integrated approaches [5]. Time-based approaches plan production and maintenance according to predetermined maintenance actions with fixed starting times or time windows, i.e. integrating the *preventive maintenance* strategy into production scheduling without considering the real health condition of a machine, see for example [6–8]. Condition-based approaches, such as PdM, carry out maintenance actions based on the assessed or predicted health condition and will be the focus for the following section. The literature will be reviewed and assessed according to the industrial viability of condition-based IPSMP approaches. In detail, these criteria are:

1. consideration of a job shop scheduling problem (JSSP) or flexible job shop scheduling problem (FJSSP) and their respective production metrics (e.g. makespan) in order to be applicable for different shop floor layouts and not being limited to single machine setups,
2. consideration of the machine condition for PdM planning,
3. operation-specific modeling of machine degradation,
4. consideration of the non-linearity, i.e. the operation sequence-dependency, of the machine degradation process, and
5. use of real industrial data for the assessment and prediction of machine condition.

2.1 Literature review

Bougacha et al. [9] proposed a PdM integrated IPSMP framework with two modules; an algorithm used for long-term RUL predictions of the system and an algorithm used for short-term predictions to estimate the system's future state while executing a job or maintenance.

The proposed framework adopts an iterative approach by considering "the effect of a selected decision on the system health and the new possible decision that can be introduced by a change in the estimated RUL" [9]. The authors considered a single machine with several operational profiles. The degradation processes were modeled using exponential functions. However, it was not specified how the parameters of the exponential functions were determined and in what relation they stand to the individual operating profiles of the machine.

Ghaleb et al. [5] followed a similar approach to those of Sloan and Shanthikumar [10] and Bajestani et al. [11] by modelling the degradation process as a continuous-time Markov chain with discrete states. A single machine with different repair policies was studied: full repair, partial repair (one-step or multi-step) and no repair. The duration and cost of a repair depend on the policy and the condition of the machine. Furthermore, deteriorating machine condition was assumed to lengthen the job processing times and increase the energy consumption. The model was solved using a genetic algorithm (GA) with a total cost minimization objective.

Pan et al. [12] utilized sensor and prognostic technologies to continuously monitor a single machine, i.e. to predict its degradation process during the processing of a job sequence. Based on this predictive information, maintenance operations are scheduled simultaneously with production jobs.

Fitouri et al. [13] proposed a decision-making heuristic with decision rules. They considered a job shop with machines subject to operation-dependent degradation. A prognostic module continuously monitors the job shop system and provides RUL predictions for each operation on each machine. Based on the predictive RUL, the cumulative machine degradation Δ imposed by an operation is calculated as follows:

$$\Delta = \frac{p}{RUL} \tag{1}$$

where p denotes the operation's processing duration. The total degradation of a machine imposed by a production schedule equals to the linear accumulation of operation-specific degradations of the assigned operations. Each machine is associated with a minimum and maximum threshold of degradation. The proposed heuristic aims at finding optimal starting times for PdM actions as well as the production jobs.

Ladj et al. [14] pursued a similar approach as Fitouri et al. [13] and studied the operation-specific degradation of a single machine. The cumulative degradation of each operation is calculated using Eq. (1) based on RUL predictions provided by a PHM module. The formulated IPSMP problem was solved using a GA with a total cost minimization criterion. The proposed approach was later extended by Ladj et al. [15] to take uncertainty in production shops into account. They considered a flow shop with uncertain RUL values and degradation of machines, which were both represented using fuzzy sets. The fuzzy sets are constructed

using the probability density function of machine failures based on historical data.

Another study investigating the RUL uncertainty was carried out by Benaggoune et al. [16]. The authors considered a multifunctional single machine and studied the impact of RUL uncertainty on maintenance planning. A PHM module was deployed to continuously monitor the machine and deliver its estimated RUL after the processing of each operation. The operations can be scheduled with respect to a predefined maximum degradation threshold, which cannot be exceeded in any situation. An operation-specific degradation δ was considered, whereas the evolution of the machine degradation f(δ) was considered linear. The aim of the formulated model is the minimization of total maintenance cost formulated as follows:

$$C_{PdM} = C_{PdM,fix} + C_{PdM,adv}\left(\delta_i(t)\right) \tag{2}$$

where $C_{PdM,fix}$ is the fixed cost of maintenance and $C_{PdM,adv}$ is the total cost of advancement, which is a linear function of the cost of advancement per unit time.

Zhai et al. [17] studied a job shop scheduling problem (JSSP) under time-varying operational conditions. The machines are subject to degradation and stochastic failures that follow the Weibull distribution. An operation-specific stress equivalent was introduced to represent the machine degradation imposed by a job operation. A not specified PHM module supplies RUL values, which are transformed into the operation-specific stress equivalents using Eq. (1). The authors assumed a linear accumulation of operation-specific degradation values that in turn influence the failure probability using the Weibull distribution. A GA was employed to solve the formulated problem.

Morariu et al. [18] explored the potentials of Big Data techniques and machine learning algorithms in PdM integrated production planning concerning energy use on the shop floor. They considered a large-scale manufacturing system, in which production scheduling decisions are based on real time learning from Big Data and data-driven decision making. Deep learning was employed to identify possible anomalies of energy consumption in the production processes. The current energy use is compared to learned energy consumption patterns, i.e. the predicted energy data based on previous production data. The proposed approach has two stages: long-term batch planning for optimal scheduling and resource allocation, and short-term resource health monitoring to detect anomalies in energy use and maintain the affected machine.

Paprocka et al. [19] aimed at generating robust schedules for job shops based on historical data of failure frequency using the Maxwell distribution. Time-varying machine failure rates were considered. The optimal schedules are selected assessing makespan, total tardiness, flow time and total idle time, while also taking into account the stability robustness and quality robustness of the schedules. In case of machine breakdown, jobs are rescheduled based on heuristic shifting rules.

Denkena et al. [20] proposed a statistical method to estimate failure durations of machine tools in order to improve production scheduling. By using data from practical experiments, their method showcases high accuracy of these prognosis that in turn can be used for production scheduling purposes. While operation-specific prediction of RUL and subsequent scheduling was not within the scope of their work, the results have potential for holistically improve IPSMP.

2.2 Research Gap and Research Objective

The assessment of the relevant literature regarding the criteria given in the beginning of this chapter is given in Table 1.

It is notable that those publications with focus on solving IPSMP problems apply genetic algorithms as solvers due to its ability to solve multi-objective optimization problems in reasonable time. In addition to the presented literature, further publications were studied and listed in the table to fully capture the scientific landscape, but not described in the section above for the sake of brevity. As the analysis indicates, the state of the art does not provide an efficient approach for the PdM integrated production scheduling of flexible job shops under varying operational conditions. Especially the linear accumulation of degradation as found in publications [13–17] does not hold true in industry and simplifies the problem at hand. Although some publications have dealt with the specific aspects of the problem formulation of this work, no holistic approach was found.

Thus, the present work aims at developing a PdM-IPS approach consisting of an optimization model for the maintenance integrated flexible job shop scheduling problem (MIFJSSP) using non-linear health predictions from a PHM module.

3 Concept

Figure 1 presents the concept of the overall PdM-IPS approach with its modules. Information from the scheduler, production orders and from the shop floor serve as inputs to the approach. The *Planning Module* receives performance measures and the weights for both production and maintenance plans. In addition, it receives the job due dates from production orders. The *PHM Module* receives CM data from sensors installed in machines on the shop floor. Corresponding operating data derived from production orders enable the *PHM Module* to correlate CM and operating data. An *Interface Module* enables the information exchange. Schedule

Table 1 Evaluation of the reviewed condition-based maintenance integrated production scheduling approaches (✓: considered, ✗: not considered)

	JSSP/ FJSSP	PdM planning based on Machine Condition	Operation-Specific Degradation	Non-Linear Accumulation of Operation-Specific Degradation	Use of Real Industrial Data
Benaggoune et al. [16]	✗	✓	✗	✗	✓
Denkena et al. [20]	✗	✗	✗	✓	✓
Morariu et al. [18]	✗	✓	✓	✗	✓
Ghaleb et al. [5]	✗	✗	✗	✗	✗
Bougacha et al. [9]	✗	✓	✗	✗	✓
Khatab et al. [21]	✗	✗	✗	✗	✓
Ladj et al. [15]	✗	✓	✓	✗	✓
Zhai et al. [17]	✓	✓	✓	✗	✗
Zandieh et al. [22]	✓	✗	✗	✗	✗
Fitouri et al. [13]	✓	✓	✓	✗	✓
Aramon Bajestani et al. [11]	✗	✗	✗	✗	✓
Xiang et al. [23]	✗	✗	✗	✗	✗
Pan et al. [12]	✗	✓	✗	✗	✓

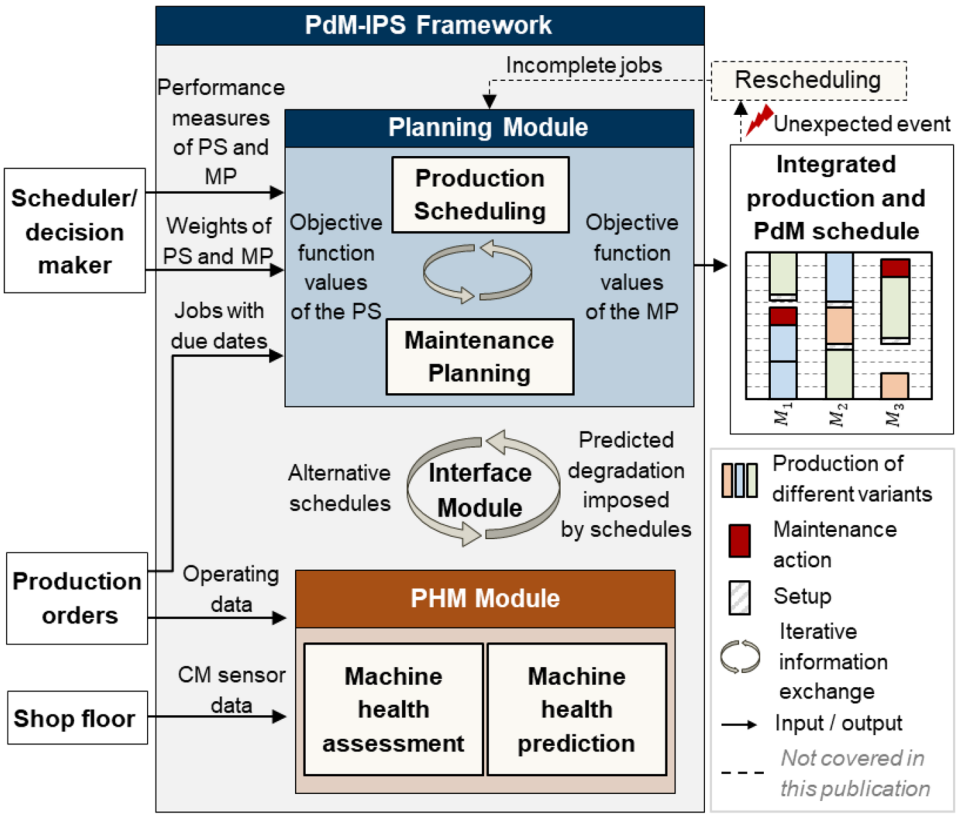

Fig. 1 Overall PdM-IPS framework (*MP* maintenance plan, *PS* production schedule)

candidates are sent to the *PHM Module* and the predicted degradation imposed by this very schedule candidate will be transmitted back to the *Planning Module*. Based on both production and degradation metrics, the *Planning Module* derives the output of the framework: an integrated production schedule with maintenance actions. Mathematically, the IPSMP part is based on a maintenance integrated version of the FJSSP to account for different shop floor layouts (i.e. MIFJSSP). As mentioned, this publication focuses the planning optimization using a two-stage Genetic Algorithm. Thus, the *Interface*, *Planning Module* and the interactions with the *PHM Module* will be explained in detail. Based on

literature analysis [24] and the author's own studies [17], genetic algorithms are well-suited to solve IPSMP problems.

The *PHM Module* that will be applied is based on the work of Zhai et al. [4] and is able to assess and predict the health condition of a machine after manufacturing a specific production schedule. Specifically, two Conditional Variational Autoencoder Neural Networks are applied to derive operation-specific health indicator prediction of machines. The algorithms presented in this publication are able to work with any other PHM module that is capable of returning an operation or product-specific health indicator prediction. We recommend our approach for short-term planning, i.e. for planning multiple shifts or weekly schedules. Depending on the data quality and availability of the *PHM module*, and thus the quality of predictions, longer planning horizons can be also realized.

4 Modelling

Based on the conceptual approach presented in the previous chapter, this chapter establishes the mathematical foundation of the PdM-IPS. Hereby, the interaction between the *Planning* and the *PHM Module* is presented in depth.

4.1 Problem description

The present work studies a flexible job shop scheduling problem with integrated predictive maintenance planning based on operation-specific machine degradation. Mathematically, the integrated problem consists of two simultaneous optimization problems: namely the production scheduling and maintenance planning in a flexible job shop setting, i.e. MIFJSSP. The production scheduling further comprises the machine assignment and operation sequencing subproblems. The MIFJSSP is extended to the PdM-IPS approach by integrating information from a *PHM Module* and resulting PdM actions. The product portfolio contains several product variants, which are produced on the multifunctional machines of the flexible job shop. The machines are subject to varying operating conditions, and thus varying degradation, due to the execution of multiple types of operation using the same machine. A production order initiates the manufacturing process of a specific number of units of a product variant, each unit corresponds to a job. In the following section, a system model for the PdM-IPS is established with the following subsystems: (1) machine, (2) product and operation, and (3) production order and job. All introduced variables and descriptions can be found in Appendix Tables 5, 6 and 7.

4.2 Subsystem 1: machine

The flexible job shop comprises a set of machines, denoted as $\mathbf{M}$, on which the jobs are produced. A single machine is denoted as $M_k \in \mathbf{M}$, whereas the total number of machines in the shop floor equals to m. The following assumptions (A) hold:

A.1 A machine can only execute one activity (job operation or a PdM action) at a time.

A.2 The degree of degradation on each machine is operation-specific. The health state, i.e. the degradation level, of the machine remains constant during setups and idle state.

A.3 PdM actions reset the machine to an "as good as new" state.

A.4 No unexpected machine breakdown or failure occurs during the schedule horizon.

The HI_k of machine takes a value between 0 (degraded) and 1 (new) representing the health of the machine M_k. RUL_k refers to the time period in which the machine is expected to be functional without breakdowns. Appendix Table 5 gives an overview of the introduced notations and variables to describe the machines in a flexible job shop.

4.3 Subsystem 2: Product and Operation

The flexible job shop can produce numerous product variants, denoted as P_v. The set of all product variants is referred to as product portfolio denoted by $\mathbf{P}$. Each product variant comprises several successive operations O_{vj} with index j of variant v. For each product $P_v \in \mathbf{P}$, the following holds:

A.5 The operations O_{vj} of a product variant P_v are subject to precedence constraints.

The total number of operations comprised by a product variant P_v is denoted as o_v, with $\mathbf{O}_v$ being its ordered set of operations. The set of machines an operation O_{vj} can be assigned to is referred to as the eligible machine set of that operation and is denoted by $\mathbf{M}_{vj}$. Each operation O_{vj} has a deterministic processing duration p_{vjk} on machine M_k:

A.6 The processing times of operations on machines are fixed, i.e. deterministic.

Appendix Table 6 gives an overview of the introduced notations and variables. Operations are associated with different operating conditions which are defined by the combination of different operational parameters [25] that lead to distinct CM values. These can be clustered into different load profiles, so called operating regimes. This concept will be introduced in detail in Sect. 4.5.2.

4.4 Subsystem 3: Production Order and Job

Each workpiece requested by a production order represents a job $J_{\mathcal{O}i}$ with the following data:

- product variant $v(\mathcal{O})$ to be produced, and
- due date $d_{\mathcal{O}i}$ equal to the due date $d_{\mathcal{O}}$ of the production order $\mathbf{PO}_{\mathcal{O}}$ it belongs to,

where i is the running index of jobs. The total number of production orders is denoted as Q. Within the scope of this work, following holds:

A.7 Production orders correspond to only one type of product variant P_v each.

Accordingly, each job inherits the specifications of the product variant $v(\mathcal{O})$ it corresponds to. Based on the subsystem "products and operations", following notations are introduced: each job $J_{\mathcal{O}i}$ comprises $o_{v(\mathcal{O})}$ operations $O_{\mathcal{O}ij}$, whereas the j-th operation $O_{\mathcal{O}ij}$ of job $J_{\mathcal{O}i}$ is an instance of the j-th operation $O_{v(\mathcal{O})j}$ of product $P_{v(\mathcal{O})}$ with the same machine program (cf. Fig. 2) and processing duration $p_{\mathcal{O}ijk}$ as follows:

$$p_{\mathcal{O}ijk} = p_{v(\mathcal{O})jk} \tag{3}$$

The starting and completion times of an operation $O_{\mathcal{O}ij}$ are denoted by $S_{\mathcal{O}ij}$ and $C_{\mathcal{O}ij}$. The completion time of job $J_{\mathcal{O}i}$ is denoted by $C_{\mathcal{O}i}$. Once a job operation $O_{\mathcal{O}ij}$ has started on a machine M_k, it cannot be interrupted or preempted by another operation:

A.8 Job preemptions are not allowed.

The set of all jobs of the PdM-IPS instance is denoted as $\mathbf{J}$, where n denotes the total number of jobs. The job operations of these jobs form an unordered set of all job operations, denoted as $\mathbf{O}$, the number of all job operations in the problem instance is denoted as N.

A.9 The jobs in the flexible job shop are independent and can be sequenced in any order that benefits the objective function values.

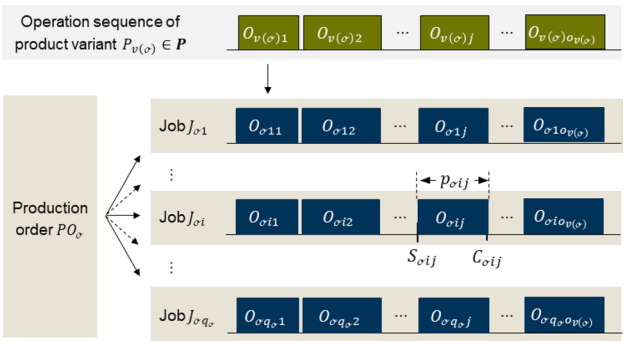

Fig. 2 Subsystem of production orders and jobs

The operations of a job are subject to precedence constraints, i.e., must be executed in the given order. Each operation is associated with a set of eligible machines $\mathbf{M}_{\mathcal{O}ij}$ on which it can be processed:

$$\mathbf{M}_{\mathcal{O}ij} = \mathbf{M}_{v(\mathcal{O})j} \tag{4}$$

Figure 2 illustrates the notation and the principle of inheritance between product variants and jobs.

Appendix Table 7 gives an overview of introduced variables to describe production orders and jobs.

4.5 Mathematical formulation

This section deals with the mathematical formulation of the PdM-IPS approach based on the assumptions made in Sect. 4.1.

4.5.1 The production scheduling problem

The production scheduling problem of the PdM-IPS comprises the *machine assignment* and *operation sequencing* subproblems of the FJSSP.

A.10 All jobs are available at the beginning of the planning horizon.

Following decision variable $Y_{\mathcal{O}ij,\mathcal{O}'i'j',k}$ is introduced to the model:

$$Y_{\mathcal{O}ij,\mathcal{O}'i'j',k} = \begin{cases} 1, & \text{if } O_{\mathcal{O}ij} \text{ immediately precedes } O_{\mathcal{O}'i'j'} \text{ on } M_k \\ 0, & \text{Otherwise} \end{cases} \tag{5}$$

In accordance with A.1, a machine can execute only one operation at a time, meaning that the starting time of an operation on machine M_k must be greater than the completion time of its predecessor:

$$S_{\mathcal{O}'i'j'k} \geq C_{\mathcal{O}ijk} - \left(1 - Y_{\mathcal{O}ij,\mathcal{O}'i'j'k}\right) \cdot B \tag{6}$$

where B is a large positive number. For the completion time $C_{\mathcal{O}ijk}$ of operation $O_{\mathcal{O}ij}$ on machine M_k must hold the following:

$$C_{\mathcal{O}ijk} \geq S_{\mathcal{O}ijk} + p_{\mathcal{O}ijk} - \left(1 - X_{\mathcal{O}ijk}\right) \cdot B \tag{7}$$

with the binary decision variable $X_{\mathcal{O}ijk}$ as follows:

$$X_{\mathcal{O}ijk} = \begin{cases} 1, & \text{if } O_{\mathcal{O}ij} \text{ can be executed on } M_k \\ 0, & \text{otherwise} \end{cases} \tag{8}$$

A machine can only be assigned to an operation, if it belongs to the eligible machine subset $\mathbf{M}_{\mathcal{O}ij} = \mathbf{M}_{v(\mathcal{O})j}$ (see Eq. 4) expressed by the parameter $\gamma_{\mathcal{O}ijk} = \gamma_{v(\mathcal{O})jk}$:

$$\gamma_{v(o)jk} = \begin{cases} 1, & if\ O_{v(o)j}\ can\ be\ executed\ on\ M_k \\ 0, & otherwise \end{cases} \tag{9}$$

Accordingly, the following must hold:

$$X_{oijk} = \gamma_{v(o)jk} \tag{10}$$

indicating that the decision variable X_{oijk} can only be 1, when $\gamma_{v(o)jk}$ equals to 1. Furthermore, the starting and ending times of operation O_{oij} can be formulated as follows:

$$S_{oij} \geq S_{oijk} \quad for\ \forall k \in \mathbf{M} \tag{11}$$

$$C_{oij} \geq C_{oijk} \quad for\ \forall k \in \mathbf{M} \tag{12}$$

In accordance with A.5, the operations O_{oij} of job J_{oi} are subject to precedence constraints, an operation cannot start prior to the completion of its predecessor:

$$S_{oij'} \geq C_{oij} - \left(1 - \mathscr{P}_{o(v)jj'}\right) \cdot B \tag{13}$$

whereas $\mathscr{P}_{o(v)jj'}$ is a binary precedence parameter as follows:

$$\mathcal{P}_{o(v)jj'} = \begin{cases} 1, & if\ O_{o(v)j}\ of\ product\ variant\ P_{o(v)}\ precedes\ O_{o(v)j'} \\ 0, & otherwise \end{cases}. \tag{14}$$

The jobs operations are subject to transportation times $t_{transport}$ and setup times t_{setup}:

A.11 The setup times between two product variants are sequence and product independent.

A.12 The time to transport a job between machines is taken into account and de-terministic. It has the same duration regardless of the origin and destination machine.

Transportation time applies when two consequent operations of a job are not processed on the same machine, expressed by the following variable:

$$\theta_{tr,oij} = \begin{cases} 1, & if\ X_{oijk} = 1\ and\ X_{oi(j-1)k} \neq 1 \\ 0, & otherwise \end{cases} \tag{15}$$

The variable of setup time needed for operation O_{oij} is expressed as:

$$\theta_{stp,o'i'j'} = \begin{cases} 1, & if\ Y_{oij,o'i'j',k} = 1\ and\ v(o) \neq v(o') \\ 0, & otherwise \end{cases} \tag{16}$$

meaning that setup is required if O_{oij} immediately precedes $O_{o'i'j'}$ on machine M_k and the product orders o and o' do not correspond to the same product variant. Thus, considering setup times, Eq. (6) can be extended as follows:

$$S_{oi'j'k} \geq C_{oijk} + t_{setup}\Delta\theta_{stp,oi'j'} - \left(1 - Y_{oij,oi'j',k}\right) \cdot B \tag{17}$$

meaning that operation $O_{o'i'j'}$ cannot start on machine M_k prior to the completion of its predecessor and the completion of setup if needed. The transportation times can be considered in Eq. (15), resulting in the following constraint:

$$S_{oij'} \geq C_{oij} + t_{transport} \cdot \theta_{tr,oij'} - \left(1 - \mathscr{P}_{o(v)jj'}\right) \cdot B \tag{18}$$

For production scheduling, the following objective functions are selected: minimization of makespan C_{max}, total tardiness T_{total} and the total penalty cost of the production schedule (PS), denoted as ζ_{PS}.

$$C_{max} = \max C_{oi}\ \ for\ \ \forall o, i \tag{19}$$

whereas C_{oi} denotes the completion time of job J_{oi}:

$$C_{oi} \geq C_{oij}\ for\ \forall j \in \boldsymbol{O}_{v(o)} \tag{20}$$

Makespan is the most commonly used objective function within production scheduling and is directly related with the cost of a PS [26]. The minimization of T_{total} is selected with respect to the industrial applicability of the model. Any order that exceeds its due date is considered as *lost opportunity* associated with costs [9]. Thus, the model aims at minimizing the total tardiness

$$T_{total} = \sum_{o=1}^{Q} \sum_{i=1}^{q_o} T_{oi} \tag{21}$$

where T_{oi} denotes the tardiness of job J_{oi}:

$$T_{oi} = max\left(0, C_{oi} - d_o\right) \tag{22}$$

Lastly, the model minimizes ζ_{PS}. ζ_{PS} is introduced to the model to minimize the transportation of jobs in the job shop as well as frequent setup actions. Each job transportation and setup action is associated with costs, denoted as $cost_{transport}$ and $cost_{setup}$, respectively:

$$\begin{aligned} \zeta_{PS} = & \sum_{o=1}^{Q} \sum_{i=1}^{q_o} \sum_{j=1}^{o_{v(o)}} cost_{transport} \cdot \theta_{transport}\left(O_{oij}\right) \\ & + cost_{setup} \cdot \theta_{transport}\left(O_{oij}\right) \end{aligned} \tag{23}$$

where $\theta_{transport}(J_{oi})$ and $\theta_{transport}(J_{oi})$ are model variables:

$$\theta_{transport}\left(O_{oij}\right) = \begin{cases} 1, & if\ t_{tr,oij} \neq 0 \\ 0, & otherwise \end{cases} \tag{24}$$

$$\theta_{setup}\left(O_{oij}\right) = \begin{cases} 1, & if\ t_{stp,oi'j'} \neq 0 \\ 0, & otherwise \end{cases} \tag{25}$$

4.5.2 The maintenance planning problem

Maintenance planning problem is concerned with the planning of PdM actions based on health state of the machines in the flexible job shop. It is assumed that the flexible job shop

in consideration is equipped with a *PHM Module*, which continuously collects sensory data and can deliver RUL_k predictions. In industrial practice, machines rarely deplete all RUL_k.

The present work adopts the concept of *Remaining Maintenance Life*, denoted as RML_k, proposed by Pan et al. [12]. Each machine M_k is associated with two *HI* values $HI_{k,safe}$ and $HI_{k,fail}$, reached at $t_{k,safe}$ and $t_{k,fail}$, respectively, see Fig. 3. Regarding maintenance planning, $t_{k,fail}$ represents the last timestep before a PdM action has to be planned on machine M_k. The time span between $t_{k,safe}$ and $t_{k,fail}$ represents the ideal time to plan maintenance. An earlier starting time $t_{k,PdM}$ of PdM than $t_{k,safe}$ means unnecessary maintenance costs. In order to integrate RML_k and RUL_k, we model maintenance costs C_{PdM} (see Fig. 3):

$$cost_{PdM,k}\left(t_{k,PdM}\right) = \begin{cases} cost_{PdM,fix} + \alpha_{PdM,adv} \cdot \left(t_{k,safe} - t_{k,PdM}\right), & if\ t_{k,PdM} < t_{k,safe} \\ cost_{PdM,fix}, & if\ t_{k,safe} \le t_{PdM} \le t_{k,fail} \\ B, & if\ t_{k,PdM} > t_{k,fail} \end{cases} \tag{26}$$

whereas $cost_{PdM,fix}$ is the fixed cost of maintenance and $\alpha_{PdM,adv}$ is the advancement cost per timestep.

The next assumptions integrate maintenance and production scheduling:

A.13 A planned PdM action will be performed immediately after the processing of a job operation is complete.

Thus, the binary decision variable $Z_{\mathcal{O}ijk}$ is introduced to the model:

$$Z_{\mathcal{O}ijk} = \begin{cases} 1, & if\ a\ PdM\ action\ is\ scheduled\ after\ O_{\mathcal{O}ijk} \\ 0, & otherwise \end{cases} \tag{27}$$

A. 14 At most one PdM action is scheduled per machine M_k

This can be mathematically formulated as follows:

$$\sum_{\mathcal{O}=1}^{Q} \sum_{i=1}^{q_{\mathcal{O}}} \sum_{j=1}^{v(\mathcal{O})} Z_{\mathcal{O}ijk} \le 1, \quad \forall M_k \in \mathbf{M} \tag{28}$$

Thus, total cost of maintenance denoted as ζ_{MP} is formulated as ζ_{MP}:

$$\zeta_{MP} = cost_{PdM}\left(t_{PdM}\right) \cdot \sum_{\mathcal{O}=1}^{Q} \sum_{i=1}^{q_{\mathcal{O}}} \sum_{j=1}^{v(\mathcal{O})} \sum_{k=1}^{n} Z_{\mathcal{O}ijk} \tag{29}$$

The PdM-IPS approach uses the *PHM Module* of Zhai et al. [4] to consider the machine degradation during IPSMP to supply RML_k and RUL_k predictions. Here, the model assumes operation-specific machine degradation, referred to as $\delta_{vjk} \in [0,1]$. It corresponds to the amount of degradation imposed on M_k during the processing of operation O_{vjk} for one timestep, if the machine would only produce this operation. The total amount of degradation imposed by the O_{vjk} is denoted as Δ_{vjk}. However, as opposed to the publications [13, 15, 17], the present work does not assume linear accumulation of the operation-specific machine degradation. In order to model operation-specific degradation, we consider each timestep of an operation individually according to its degradation. This allows us to accurately predict the machine health in consideration of the active operation sequence. The model quantifies the effects of different processing steps during an operation O_{vjk}, referred to as operating regime *OpReg*. Each *OpReg* represents a different load profile of the machine. $OpReg_t$ corresponds to the active *OpReg* in timestep t and is dependent on the active job operation O_{vjk}. *OpRegs* are identified by clustering available CM data, during the time an operation O_{vjk} is active, into K clusters. Thus, an operation O_{vjk} corresponds to an ordered set, i.e. sequence, of active operating regimes:

$$OpRegSeq\left(O_{vjk}\right) = \left[OpReg_{vjk,1},\ OpReg_{vjk,2},\ \ldots, OpReg_{vjk,p_{vjk}}\right] \tag{30}$$

where it holds $OpReg_{vjk,t} \in [OpReg_1, \ldots, OpReg_K]$ and $t \in p_{vjk}$. The total degradation Δ_{vjk} imposed by an operation O_{vj} on machine M_k is dependent on the sequence of operating regimes the operation comprises:

$$\delta_{vjk} = f\left(OpRegSeq\left(O_{vj}\right)\right) \tag{31}$$

where $OpRegSeq_{vj}$ denotes the *OpReg* sequence of operation O_{vj}. Accordingly, the degradation of machine M_k imposed by a schedule, denoted as $\mathcal{S}$, also depends on the sequence of *OpRegs* of that schedule:

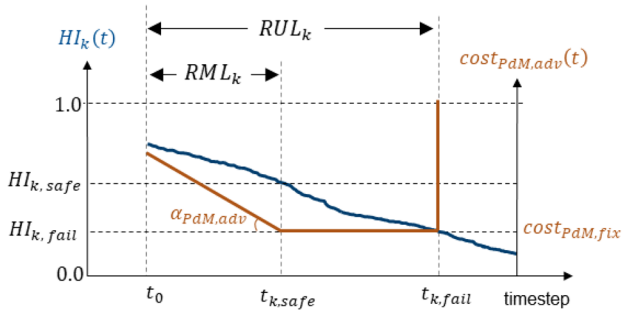

Fig. 3 RML concept [12] and adapted PdM advancement cost model [27]

$$\Delta_k(\mathcal{S}) = f(OpRegSeq(\mathcal{S})) \tag{32}$$

In accordance with assumption A.2, setup and idle times do not change the degradation of the machine, as no *OpReg* is active.

The total degradation of machine M_k due to schedule $\mathcal{S}$ is formulated as follows:

$$\Delta_k(\mathcal{S}) = HI_{k,pred}(\mathcal{S}) - HI_k(t=0) \tag{33}$$

where $HI_{k,pred}(\mathcal{S})$ represents the predicted level of $HI_{k,pred} \in [0,1]$ after the execution of schedule $\mathcal{S}$.

The total degradation Δ_{total} of the machines should be minimal, so the frequency of maintenance actions and associated cost can be minimized.

$$\Delta_{total} = \sum_{k=1}^{n} \Delta_k, \tag{34}$$

Furthermore, the critical machine degradation $\Delta_{critical}$, i.e. the maximum degradation of a single machine, is chosen as minimization criteria:

$$min\ \Delta_{critical} = \max \Delta_k\ \forall k \in \boldsymbol{M} \tag{35}$$

This facilitates the balanced use of resources. By using the machines to a similar extent, the PdM actions can be synchronized, which would result in time and cost savings in industrial practice [17].

4.5.3 The integrated problem

The IPSMP part of the PdM-IPS approach is formulated by combining the constraints of the partial problems outlined in previous sections. In consideration of the binary decision variable Z_{oijk} for maintenance planning, Eq. (17) is reformulated:

$$S_{oi'j'k'} \geq C_{oijk} + t_{setup} \cdot \theta_{strp,oi'j'} + t_{PdM} \cdot Z_{oijk} - \left(1 - Y_{oij,oi'j',k}\right) \cdot B \tag{36}$$

where t_{PdM} denotes the duration of one PdM action. The calculation of C_{max} of the integrated schedule requires a reformulation from Eq. (19):

$$C_{\max} = \max\left(C_{oi} + t_{PdM} \cdot Z_{oio_ik}\right) \forall o, i, j, k, \tag{37}$$

to consider the case where a PdM action is scheduled after the last operation O_{oio_i} of a job J_{oi}. All other constraints presented in previous sections remain unchanged for the integrated problem. Thus, the MIFJSSP comprises following decision variables:

- $Y_{oij,oi'j',k}$ for operation sequencing (Eq. 9),
- X_{oijk} for machine assignment (Eq. 8), and
- Z_{oijk} for PdM scheduling (Eq. 28).

and a total of six objective functions presented in the previous sections. The objective functions regarding the PS are as follows:

$$\min C_{\max} \tag{38}$$

$$\min T_{total} \tag{39}$$

$$\min \zeta_{PS} \tag{40}$$

The maintenance plan (MP) has the following objective functions:

$$\min \Delta_{total} \tag{41}$$

$$\min \Delta_{critical} \tag{42}$$

$$\min \zeta_{MP} \tag{43}$$

5 Two-stage genetic algorithm (TSGA)

We propose a TSGA for the PdM integrated production scheduling of flexible job shops. The general procedure of the designed algorithm is shown in Fig. 4. The optimization process refers to the simultaneous scheduling of job operations and PdM actions. The integrated scheduling is carried out by the TSGA based on the feedback provided by the *PHM Module*. As mentioned, the genetic algorithm is chosen due to its viability to solve IPSMP problems as the scientific literature points out. Specifically, its ability to solve the present multi-objective optimization problem in reasonable time is crucial for industrial applicability.

We adopt a two-stage procedure comprising a single-objective stage *S*1 and a multi-objective stage *S*2. The single-objective *S*1 represents a preliminary stage for generating a high-quality initial population for the *S*2, aiming at enhancing the convergence speed of the multi-objective stage [28], which represents the main phase where both production scheduling and maintenance planning are jointly optimized.

5.1 Single-objective Stage S1

*S*1 starts by randomly initializing a population and cloning it into two populations, referred to as *Population*S1(1) and *Population*S1(2). Each chromosome corresponds to a candidate solution and represents a PS. These populations are evolved by two single-objective genetic algorithms with fitness functions $f_{S1(1)}$ and $f_{S1(2)}$, *GA-S*1(1) using the makespan criterion min C_{max} and *GA-S*1(2) minimizing the total tardiness, i.e. min T_{total}. The latest generations of these populations are then merged into a high-quality initial population

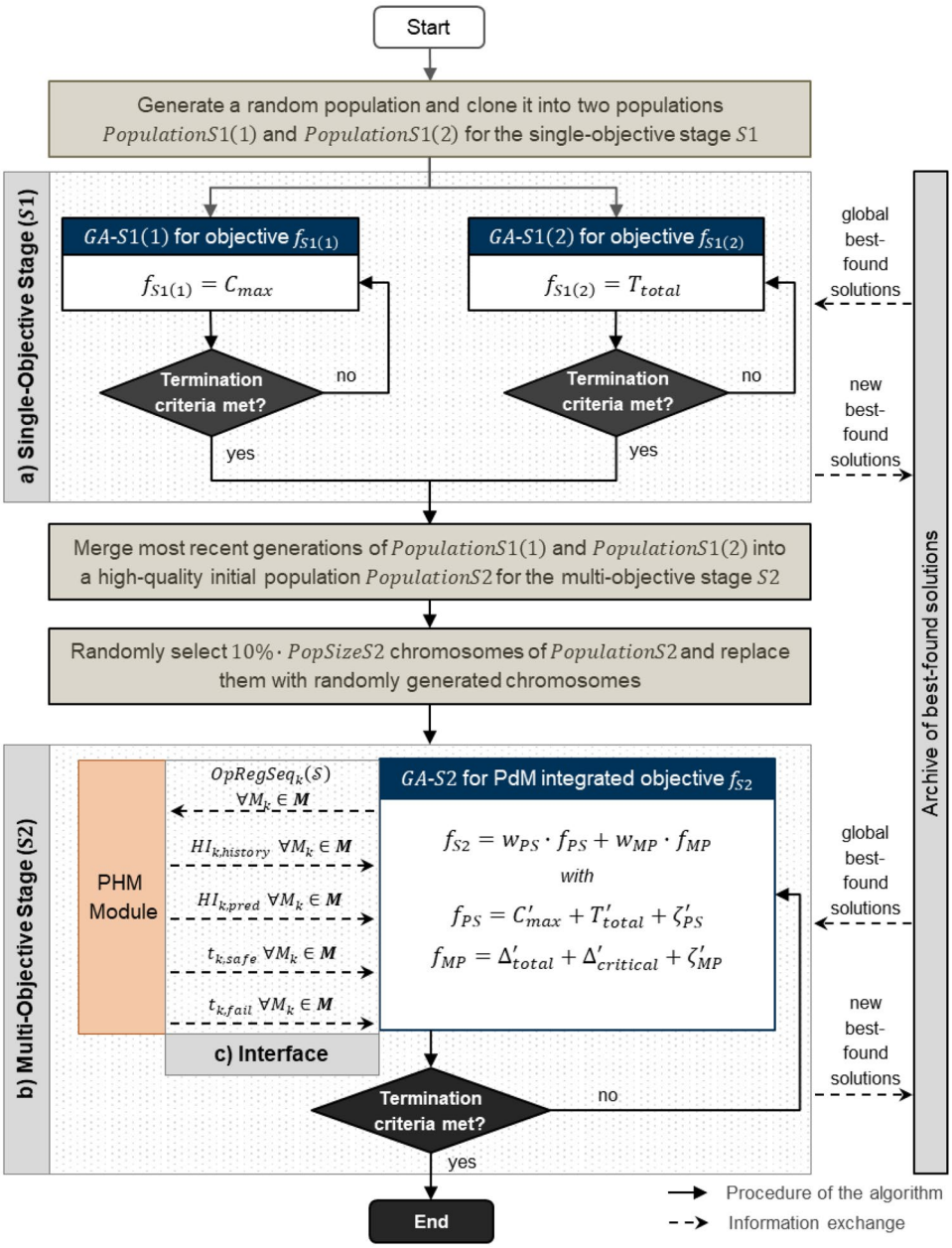

Fig. 4 Flowchart of the proposed TSGA with **a** single-objective stage S1, **b** multi-objective stage S2 and **c** interface between the *PHM module* and GA-S2

for $S2$, referred to as *PopulationS2*, comprising *PopSizeS2* chromosomes. Thus, it is ensured that the search process of $S2$ will start in promising regions of the genetic search space. In order to enhance population diversity and prevent premature convergence, 10% of *PopSizeS2* are arbitrarily selected and replaced with randomly generated chromosomes.

5.2 Multi-objective stage S2

The second stage aims at the multi-objective optimization of both production and maintenance using following fitness function:

$$f_{S2} = w_{PS} \cdot f_{PS} + w_{MP} \cdot f_{MP} \tag{44}$$

where f_{PS} and f_{MP} refer to the fitness functions of the PS and MP, their corresponding weights w_{PS} and w_{MP} (with $w_{PS} + w_{MP} = 1$) and represent the input data of the TSGA.

The fitness function of the PS comprises the minimization of makespan C_{max}, total tardiness T_{total} and total cost of PS ζ_{PS}, which were formulated in Sect. 4.5.1. As these objectives have different units, their simultaneous optimization using the weighted sum method is not possible. Thus, the scaling method proposed by [29] is adopted in order to constitute a linear fitness function for the PS and MP:

$$f_{PS} = C'_{max} + T'_{total} + \zeta'_{PS} \tag{45}$$

$$f_{MP} = \Delta'_{total} + \Delta'_{critical} + \zeta'_{MP} \tag{46}$$

where $C'_{\max}$, T'_{total} and ζ'_{PS} are scaled fitness values. Here, each objective function value *fitness* of a chromosome, denoted as $\mathcal{C}$, will be converted to a scaled value *fitness*$'$ as follows:

$$fitness'(\mathcal{C}) = \begin{cases} \dfrac{fitness(\mathcal{C}) - MIN\left(Gen_z\right)}{MAX\left(Gen_z\right) - MIN\left(Gen_z\right)}, & if\ MAX\left(Gen_z\right) \neq MIN\left(Gen_z\right) \\ 0, & otherwise \end{cases} \tag{47}$$

where $MIN(\mathcal{G})$ and $MAX(\mathcal{G})$ correspond to the fitness values of the fittest and least fit chromosomes within the chromosome's generation Gen_z, respectively.

Hereby, each schedule comprises m job operation sequences, i.e. one for each machine M_k of the flexible job shop. The adopted *PHM Module* simulates each sequence regarding the following:

- current health state of the machine, expressed by HI_k, and
- the production history, i.e. operation sequence previously executed on the machine.

Here, the latter is expressed in form of HI_k history of the machine, denoted as $\mathbf{HI}_{k,history}$. Hence, assuming a TSGA population with *PopSize* chromosomes, the algorithm requires $m \times PopSize$ predictive simulations per generation.

5.3 Interface to the *PHM module*

The application of the proposed approach enables the decision maker (scheduler) to consider the machines' health states during production planning, and thus to simultaneously optimize the PP and MP. As visualized in Fig. 4c, the proposed TSGA and the *PHM Module* have an iterative exchange of information during the entire optimization process.

Functionality and Architecture of the PHM Module The *PHM Module* simulates candidate production schedules, i.e. chromosomes, generated by the TSGA and forecasts the future HI_k trajectory $\{HI_k(t), HI_k(t+1), \ldots\}$ based on candidate schedules. It comprises two generative deep learning models, namely conditional variational autoencoders (CVAE) as modelled by Zhai et al. [4]:

- a *health assessor model* (HA-CVAE), used for deriving a health indicator (HI) of a machine, and
- a *data simulator model* (DS-CVAE), which generates realistic synthetic multivariate CM data that resemble those of the training data under the same *OpReg* and health state.

The operation-specific machine degradation evaluation of alternative schedules requires the joint deployment of these models. In order to consider the operation-specificity of the machine degradation process, the adopted *PHM Module* clusters CM data into operation specific clusters, i.e. *OpRegs*. A sequence of *OpReg* is denoted as *OpRegSeq* and mirror the operations necessary to manufacture a product—in other words, candidate PSs can be translated to *OpRegSeq* which are then evaluated by the *PHM Module* to obtain changes in HI_k. The main steps of the predictive simulation are summarized in the following, which occur recursively for each timestep t:

- The DS-CVAE uses the information of the active *OpReg* of the candidate production schedule (PS) by the TSGA and the health condition $HI_k(t)$ to generate synthetic sensor observations.
- The sensor observations generated by the DS-CVAE are passed to the HA-CVAE for health assessment. Its output is the predicted $HI_{k,pred}(t+1)$.

The described procedure is illustrated in Appendix Fig. 6.

Interface Between the TSGA and *PHM Module* The feedback regarding the health states of machines is considered by the TSGA within the fitness function f_{MP} of the multi-objective stage $S2$. As expressed in Eq. (46), this function comprises the minimization of total degradation of machines Δ_{total}, critical machine degradation $\Delta_{critical}$ and the total cost of MP ζ_{MP}. Considering the Eqs. (29), (34) and (35) of these objectives, the following can be identified as the output data required by the *PHM Module*:

- predicted health indicator $HI_{k,pred}$,
- predicted timestep $t_{k,safe}$ in which the health state will reach $HI_{k,safe}$, and
- predicted timestep $t_{k,fail}$ in which the health state will reach $HI_{k,fail}$.

In correspondence, the *PHM Module* requires the following input data:

- current health state expressed by HI_k,
- job operation sequence of the machine in form of $OpRegSeq_k$, and

- the health indicator history $HI_{k,history}$ up to the time of prediction, i.e. the current timestep.

The TSGA and *PHM Module* interact through a defined interface realized by the function *simulate_schedules*, of which the pseudo-code is shown in Appendix Algorithm 3. Each chromosome corresponds to m simulations, i.e. the number of machines in the flexible job shop. When all predictive simulations for every machine are completed, TSGA calculates the total degradation of machines Δ_{total}, critical machine degradation $\Delta_{critical}$ and the total cost of MP ζ_{MP} of the chromosome.

5.4 Genetic Operators

Chromosome Encoding Chromosomes correspond to candidate solutions to the MIFJSSP, i.e. production schedules with PdM actions. The proposed TSGA adopts a chromosome representation with three segments as visualized in Appendix Figs. 7, 8 and 9. Each segment is a vector consisting of integers and corresponds to a subproblem of the MIFJSSP (see Sect. 4.1) as follows:

- operation sequencing segment *OS*,
- machine assignment segment *MA*,
- maintenance planning segment *MP*.

OS and *MA* are adopted from [30, 31] and [32]. To include the maintenance planning subproblem, this representation is extended by *MP*. Each segment uses an operation-based encoding and comprises N genes, where N is the total number of job operations. By choosing and applying suitable genetic operators for each segment, the creation of infeasible chromosomes during iterations are avoided and no repair mechanism is needed.

Crossover operators The *precedence operation crossover (POX), job-based crossover (JBX)*, and *2-point crossover (2PX)* adopted from Li & Gao [30] are applied in the TSGA. For the *OS* segment, the TSGA employs either the POX or JBX operator, where one is selected randomly. The *MA* and *MP* segments are executed by the 2PX operator. In *S*2, the TSGA chooses randomly whether to crossover the *OS* and *MA* segments or the *MP*. In other words, at each iteration of the TSGA, either the production plan (represented by the *OS* and *MA* segments) or the maintenance plan (represented by the *MP* segment) is altered in order the generate offsprings for the next generation. The described crossover procedure in the multi-objective stage *S*2 is summarized as pseudo-code in Appendix Algorithm 1.

Mutation operators Mutation operators are applied to the new generation Gen_z generated by the crossover procedure. In each new TSGA generation, a chromosome is selected randomly. The proposed TSGA uses *swapping, neighborhood* and *flip mutation* operators with *mutation rate* p_m [30], as well as *binary flip mutation* operator designed by the present work for the *MP* segment. The *OS* segment is executed by the *swapping* and *neighborhood mutation* operators. The general procedure adopted for the mutation of generations is analog to the crossover process. Thus, either the OS and MA segments are mutated with a probability of p_m.

Fitness evaluation and selection operators Selection operators are responsible for choosing the parent chromosomes from a generation, denoted as Gen_z, that will produce the offspring chromosomes for the next generation Gen_{z+1}. Thus, the result of the selection process is a set of chromosomes, referred to as the *parental population*, for the crossover process. The proposed TSGA adopts the selection procedure of Li and Gao [30] using the *elitist* and *tournament* selection operator.

The described selection procedure is summarized as pseudo-code in Appendix Algorithm 2. The combined application of the elitist and tournament selection operators allows the TSGA to make a trade-off between exploration and exploitation of the genetic search space. In order to prevent the loss of the best-found solutions between generations, an external archive is employed based on publications [33, 34]. For a population with *PopSize* chromosomes, the archive contains *archiveRate* * *PopSize* of the best chromosomes found by the entire TSGA generations. In each TSGA iteration, the chromosomes stored in the archive are added to the population of the most recent generation Gen_z.

Chromosome decoding By decoding, the chromosome is translated to the PdM integrated schedule. We adopt the priority-based decoding and reordering by [31]. The decoding process is carried out by scanning the *OS* segment of the chromosome from left to right, with the machine-assignment information for each operation provided by *MA*. Finally, the *MP* segment specifies whether the operation is followed by a PdM action. The main steps of the decoding process are summarized in Appendix Algorithm 4.

Termination The termination of both stages *S*1 and *S*2 use the following criteria:

- the convergence of the optimization process, i.e. the allowed maximum of generations with stagnation of the fitness value, and
- the allowed maximum number of generations, denoted as *maxGenS*1 and *maxGenS*2.

Note that due to the stochastic nature of the health predictions in f_{S2}, the fitness of S2 will not converge in a monotonic decreasing manner. Thus, the termination criteria will be based on the average total cost of production schedule and maintenance plan, makespan and tardiness

$\zeta_{PS} + \zeta_{MP} + T_{total} + C_{max}$ for the 10 fittest chromosomes in each generation.

6 Computational results

In this section, the results of computational experiments of the TSGA are presented. The objectives of the computational experiments are as follows:

- check the plausibility of the implemented code, i.e. determine whether the implemented algorithms are capable of generating feasible solutions for FJSSP instances,
- investigate the performance of the different components of the proposed algorithm,
- calibrate the hyperparameters of the implemented algorithm, and
- investigate the convergence of the search process.

One of the common challenges in the field of PdM integrated scheduling is that there are no benchmark data available for the testing of proposed algorithms [15]. Thus, for the following experiments in this section, a commonly used FJSSP benchmark dataset by Brandimarte [35] was adopted. The used test instance ("Mk01") belongs to a 10×15 FJSSP with 10 machines and 15 jobs. The test instance comprises a total of 55 operations and is aimed at makespan minimization. Thus, for the following experiments in this section, a simple single-stage structure of the proposed GA is considered, which corresponds to the *GA-S*1(1) with $f_{S1(1)} = \min C_{max}$ of the TSGA.

For the hyperparameter optimization, a full-factorial experimental design with four levels was used, see Appendix Table 8. Based on these experiments, the setting in Table 2 is chosen for the application of the proposed TSGA in the following sections:

Note that the reported computational study of hyperparameters was conducted using the *GA-S*1(1) in the single-objective stage *S*1 of the proposed TSGA. However, all GAs of the TSGA are implemented using the same chromosome representation and genetic operators. Furthermore, the chosen test instance of Brandimarte [35] exhibits a problem size which is representative of the case studies in the following chapters. Hence, the results of these computational experiments can be transferred to the TSGA. In order to improve the usability of the approach, a graphical user interface (GUI) was developed. Job shop data, as well as parameters for cost and time and TSGA hyperparameters can be easily set. In Fig. 5, the GUI shows an exemplary PdM integrated production schedule with corresponding costs.

Next, a simulated dataset and a real industrial dataset are applied to investigate the general applicability of the algorithm. For the sake of brevity, we will focus on reporting respective optimization metrics and the associated total cost of the schedule and not visualize every schedule, since the visualizations do not offer any more information than already displayed in Fig. 5.

Table 2 TSGA hyperparameter setting for case studies

Maximum generation *MaxGen*	Maximum stagnation	Population size *PopSize*	Crossover rate p_c	Mutation rate p_m	Reproduction rate (elitism) p_r
S1: 300, S2: 100	$0.1 * MaxGen$	300	0.90	0.20	0.05

6.1 Case 1: Partial MIFJSSP with simulated dataset

In order to assess the applicability of the developed algorithm, an exemplary PdM-IPS instance is studied in this section. Here, a *partial flexible job shop* is considered using the NASA's simulated C-MAPSS dataset [36]. Note that this dataset comprises run to failure CM data of jet engines and not manufacturing data. Therefore, virtual products and operations, as well as corresponding machines that mirror the jet engine's degradation behavior are generated.

Dataset description Case 1 adopts the C-MAPSS FD002 sub-dataset. The dataset was clustered into a total of 6 *OpRegs*: $\{OpReg_1, \ldots, OpReg_6\}$. Accordingly, the *OpRegSeq* of an operation may comprise several operating regimes, i.e. consists of several sub-operations with different degrading effects on the machines. At each time step, the sensor observation was mapped to the currently active *OpReg*. The HA-CVAE and DS-CVAE models were trained and supplied by [4]. The results presented in this section are obtained by integrating the trained models into the developed TSGA.

Case description The *OpRegs* identified for the C-MAPSS are used by the present work to generate virtual products and operations. An exemplary middle-sized PdM-IPS instance was created considering a partial flexible job shop:

Machine Data. The partial flexible job shop comprises 5 multifunctional machines. It is assumed that the machines are identical, and thus have the same $HI_{k,safe}$ and $HI_{k,fail}$ values. However, they show different levels of machine health $HI_k(t = 0)$ at the time of planning, i.e. are degraded to varying degrees, see Appendix Table 9. The *PHM Module* requires the historical records of operational conditions up to the time of prediction. Thus, five time series with varying machine degradation rates were selected from the training dataset used for the CVAE models. The selected time series correspond to the $HI_{k,history}$ of machines.

Product Data The product portfolio **P** of the problem instance comprises 5 product variants P_v, corresponding

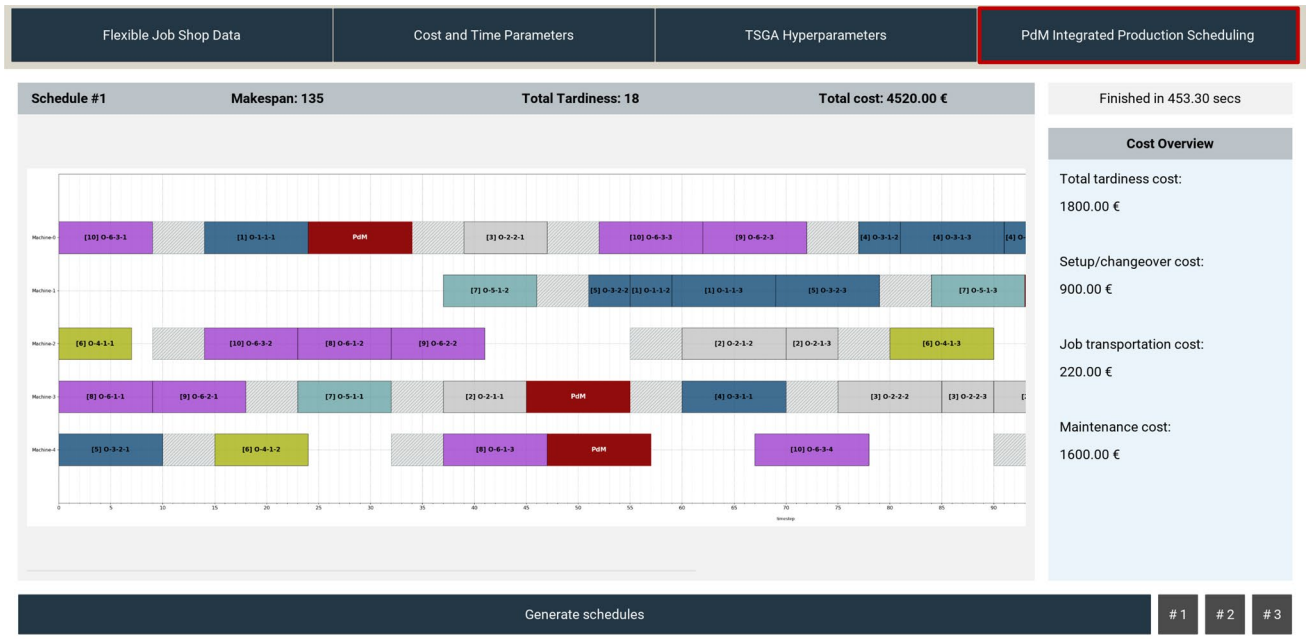

Fig. 5 Exemplary visualization of a PdM integrated production schedule

to a total of 16 operations O_{vj}. A detailed overview of the processing times p_{vj} and eligible machine sets $\mathbf{M}_{vj}$ of these operations is given in Appendix Table 10.

Product Order Data The problem comprises 8 production orders $PO_{\mathcal{O}}$, corresponding to 20 jobs $J_{\mathcal{O}i}$ with a total of 61 job operations $J_{\mathcal{O}i}$. The product variants P_v, due dates $d_{\mathcal{O}}$ and quantities $q_{\mathcal{O}}$ of these production orders are given in Appendix Table 11.

Cost and Time Parameters The cost and time parameters of PdM actions, setup actions between product variants and transportation of jobs are given in Appendix Table 12.

Application of the TSGA The hyperparameters of $S1$ are given in Table 2. For $S2$, the number of iterations was set to $MaxGenS2 = 100$. The production plan and maintenance plans are weighted with same importance, i.e. $w_{PP} = w_{MP} = 0.5$. The algorithm was run on a macOS system with 2.4 GHz Intel Core i5 processor and 8 GB RAM.

As described in Sect. 5.1, $S1$ evolves two populations in parallel, namely $PopulationS1(1)$ and $PopulationS1(2)$. For the MIFJSSP instance studied in this case, the evolving process of both populations were terminated due to the convergence criterion after 110.48 seconds. In particular, the $GA\text{-}S1(1)$ terminated after 191 generations with a makespan $C_{max} = 191$, whereas $GA\text{-}S1(2)$ terminated after 94 generation and was able to minimize the total tardiness to $T_{total} = 0$.

The most recent generations of $PopulationS1(1)$ and $PopulationS1(2)$ were merged by the TSGA to a high-quality initial population $PopulationS2$, which evolved for $MaxGenS2 = 100$ iterations in 567.0 s. The fittest chromosome represents the PdM integrated schedule selected by the TSGA with C_{max}=230, T_{total}=22, ξ_{PP}=1480 and ξ_{MP}=1600.

The total cost of the selected schedule corresponds to 3080€, composed of setup costs (1150€), transportation costs (330€) and cost of maintenance (1600€). (see Appendix Table 12 for cost positions). As shown in Table 3, each of these PdM actions were scheduled with a starting time $t_{k,PdM}$ between $t_{k,safe}$ and $t_{k,fail}$, thus constituting no advancement costs. Table 3 also shows the predicted health conditions of machines at time $t_{horizon}$, i.e. after the execution of the schedule. The critical machine degradation Δ_{max} imposed by the integrated schedule is printed in bold and belongs to machine M_2.

The predictions indicate that the machines M_1 and M_4 do not reach their failure states. M_1 reaches its $HI_{1,safe}$ value and was accordingly scheduled by the TSGA to be maintained after the completion of its job operations. Although M_2 exhibits the lowest HI_k (i.e. worst health condition), the *PHM Module* predicts that M_5 will reach its $HI_{5,safe}$ before M_2 reaches $HI_{2,safe}$. However, the time period between $t_{5,safe}$ and $t_{5,fail}$ is longer than it is for M_2. Furthermore, M_4 is expected to degrade the least. It can be concluded that the *PHM Module* is able to assess and predict operation-specific machine degradation based on the assigned operations. Using this predictive information, the TSGA schedules the necessary PdM actions so that (1) the machines do not reach their failure states at time $t_{k,fail}$ and (2) the multi-objective f_{S2} of the schedule is minimized. The analysis shows that the

scheduling results of the implemented TSGA are effective and reasonable.

6.2 Case 2: Total MIFJSSP with industrial CM data

Dataset Description For the following case, a real industrial dataset from an automotive manufacturing company is applied. Note that due to confidentiality, raw data and metrics associated with efficiency and costs cannot be shown. The dataset belongs to a machine park with five multifunctional machines and comprises CM data collected from 140 sensors covering a time period of 9 months. The significant differences in the real industrial dataset compared to C-MAPSS are as follows:

- During the preprocessing step, a total of $K = 32$ *OpRegs* were defined. 15 of these belong to the manufacturing of a product, i.e. operating conditions during which the machine experiences degradation.
- Compared to C-MAPSS, the *OpRegs* are defined high-level (workpiece ID). Thus, *OpRegSeq* of each product comprises a single operating regime.

Case Description

Machine Data The flexible job shop comprises three identical multifunctional machines with different health conditions as listed in Appendix Table 13.

Product Data The product portfolio **P** of the studied problem instance comprises 10 product variants P_v. As mentioned above, each *OpReg* corresponds to a product variant. Thus, each product has one operation O_{v1} with a *OpRegSeq* consisting of a single *OpReg*. The processing times p_{vj} are listed in Appendix Table 14. The eligible machine sets $\mathbf{M}_{vj}$ are omitted as a total flexible job shop is studied, i.e. each operation can be produced on any machine.

Product Order Data The problem comprises a total of 50 jobs J_{oi} that belong to 12 production orders PO_o. The product variants P_v, due dates d_o and quantities q_o of these production orders are given in Appendix Table 15.

Cost and Time Parameters. The problem uses the same cost and time parameters of Case 1 given in Appendix Table 12.

Application of the TSGA

The hyperparameters of the TSGA are configured as in the previous Case 1. The algorithms were run on a Windows system with 3.3 GHz Intel Xeon processor and 64 GB RAM.

The single-objective stage S1 was terminated after 416.11 seconds. The *GA-S1*(1) terminated after 145 generations due to the convergence with a makespan value of $C_{max} = 526$. The *GA-S1*(2) terminated after 99 generations with a total tardiness of $T_{total} = 0$. To showcase the effectiveness of the archive, a simulation run without archive was conducted. Without archive, GA-S1(1) converged after 236 with $C_{max} = 575$, while GA-S1(2) terminated after 142 generations with $T_{total} = 0$. This clearly shows that the archive approach ensures better results while requiring less computation.

Table 4 shows the results of the PdM integrated schedule selected by the TSGA after *S2* for the PdM-IPS instance under study. The algorithm terminated after 1998.1 s. The table gives an overview of the machines' health conditions as predicted by the *PHM Module* and the starting times $t_{k,PdM}$ of the PdM actions as scheduled by the TSGA. The critical machine degradation Δ_{max} imposed by the integrated schedule is printed in bold.

With a predicted machine degradation $\Delta_1 = 0.1577$, machine M_1 is expected to experience the highest degradation imposed by the selected schedule. Another finding is that the predicted machine degradations are significantly lower than in Case 1, although the problem size is bigger (i.e. the prediction horizon is longer). This may be explicated by the fact that the *OpRegs* in the industrial dataset have been defined on product level, leading to a generalization of degradation processes and reduced accuracy of the operation-specific machine predictions.

7 Conclusion and outlook

Conclusion

This publication presented the PdM-IPS approach in order to realize integrated production and maintenance planning using operation-specific predictions from a *PHM Module*. After review of related literature, five main requirements for value-adding IPSMP in industrial application were identified:

Table 3 Predicted health conditions of machines in the selected schedule in Case 1

M_k	Δ_k	$HI_k(t=0)$	$HI_{k,pred}(t_{horizon})$	$t_{k,safe}$	$t_{k,fail}$	$t_{k,PdM}$
M_1	0.3481	0.7912	0.4431	163	–	170
M_2	0.5036	0.6579	0.1543	58	86	66
M_3	0.3551	0.7144	0.3593	80	118	90
M_4	0.1095	0.8777	0.7682	–	–	–
M_5	0.3926	0.7174	0.3248	37	118	85

Table 4 Predicted health conditions of machines in the selected schedule in Case 2

M_k	Δ_k	$HI_k(t=0)$	$HI_{k,pred}(t_{horizon})$	$t_{k,safe}$	$t_{k,fail}$	$t_{k,PdM}$
M_1	0.1577	0.8963	0.7386	242	321	255
M_2	0.0704	0.8503	0.7799	218	–	260
M_3	0.1495	0.8406	0.6911	101	205	122

R1: Consideration of a JSSP or FJSSP in order to be applicable for different shop floor layouts and not being limited to single machine setups,

R2: consideration of the machine condition for maintenance planning,

R3: operation-specific modeling of machine degradation,

R4: consideration of the non-linearity, i.e. the operation sequence-dependency, of the machine degradation process, and

R5: use of real industrial data for the assessment and prediction of machine condition.

To fulfill these requirements, the presented PdM-IPS approach models the production scheduling part after the FJSSP, while the maintenance planning is based on non-linear accumulation of degradation. The mathematical model was formulated and solved using a two-stage genetic algorithm. Two case studies using both artificial and real industrial data showcased that the approach is applicable to different scenarios. Furthermore, results indicate that feasible production and maintenance integrated schedules can be retrieved. Given its nature, the genetic algorithm cannot guarantee a global optimum. However, results indicate that the fitness function significantly improves over generations and does not converge prematurely and thus, (local) optima concerning production and maintenance objectives are found. Following statements concerning the requirements can be made:

R1: The exemplary applications of the TSGA in Sect. 6 confirmed that the proposed algorithm is able to generate production plans for both partial and total flexible job shops.

R2: By adopting a three-segment chromosome, each corresponding to a subproblem of the IPSMP, the TSGA is able to simultaneously optimize both production and maintenance. Thus, the starting times of job operations and PdM actions are jointly scheduled with regard to the overall performance.

R3 + R4: These requirements have been met by the proposed PdM-IPS approach through the integration of the unsupervised deep learning *PHM Module* developed by [4]. As described, this model was adopted by the present work in order to obtain operation-specific predictions regarding machine deterioration. Each job operation to be scheduled by the developed TSGA corresponds to a sequence of *OpRegs*, that in turn is evaluated by the *PHM Module* according to its operation-specific degradation. The deep learning model learns the representation of data and accumulates degradation non-linearly.

R5: The modelling of the IPSMP based on a MIFJSSP ensures high applicability to different production layouts and is thus industrial viable. The applied *PHM Module* is able to handle industrial-scale CM data. The application of the TSGA enables the timely generation of schedules while at the same time requiring reasonable computation time. Finally, the two use cases presented in Sect. 6 showcase that the approach indeed can be applied to an industrial setting. Results indicate that the TSGA converges and critical machine conditions and potential failures can be avoided through scheduling PdM actions.

To conclude, by considering the future machine condition before the execution of a production schedule, the implemented approach enables the decision maker to avoid failures due to machine degradation. Job operations and PdM actions are jointly optimized regarding overall cost of the manufacturing system, resulting in improved planning and cost efficiency. It is noteworthy, however, that the benefits of PdM-IPS depend on the shop floor layout of the manufacturing company. In a flexible job shop layout, i.e. where jobs can be freely routed, the benefits are the biggest due to the inherent flexibility of the system to schedule jobs on less degraded machines. In traditional flow shops where the routing of jobs is limited, PdM-IPS can still optimally schedule maintenance and production, but has less room to control future degradation on machines. Together with recent developments of "Industrie 4.0" that enables the so called matrix production with flexible production cells [37], which represent a flexible job shop, the relevance of PdM-IPS will further increase.

Outlook

Further research can and should be conducted on the basis of this publication. Specifically, three major fields should be investigated upon.

Handling of uncertainties. In this work, we consider deterministic times (e.g., deterministic processing and maintenance times, new job arrivals, etc.). By considering uncertainties, i.e. non-deterministic times, robustness measures of the generated predictive schedules can be improved. Robust and stable schedules may help to decrease the costs due to unexpected events. Moreover, this would represent the first step toward dynamic flexible job shop scheduling.

Determining HI thresholds for maintenance. This publication assumed the thresholds for $HI_{k,safe}$ and $HI_{k,fail}$ based

on observation of degradation trends and expert knowledge. Future works should focus on an automated approach to derive these thresholds based on data and study its effect and sensitivity on subsequent scheduling approaches.

Development of a methodological approach for the evaluation and validation of PdM frameworks. One of the major challenges in the field of data-driven PdM is the lack of methodologies that enable the systematic evaluation of the operational and monetary performance of a proposed PdM approach. Future research needs to investigate which KPIs are suitable for this purpose, allowing the benefits promised by a PdM approach to be quantified. This is especially important for post-prognostic decision support approaches like the presented PdM-IPS.

8 Appendix

The appendix consists of Tables 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, Figures 6, 7, 8, 9 and Algorithms 1, 2, 3, 4.

8.1 Notations and variables

Table 5 Notations and variables of the subsystem "Machine"

Symbol	Designation
HI_k	Health index of machine M_k, $HI_k \in [0, 1.0]$
k	Index of machines
$m \in \mathbb{Z}^+$	Number of machines M_k in the problem instance,$\|\mathbf{M}\| = m$
$\mathbf{M} = \{M_k : k \in \{1, \dots, m\}\}$	Set of $m \in \mathbb{Z}^+$ machines M_k
M_k	Machine
RUL_k	Remaining useful life of machine M_k in [timesteps]

Table 6 Notations and variables of the subsystem "Products and Operations"

Symbol	Designation
j	Running index of operations comprised by a product variant
$\mathbf{M}_{vj} \subseteq \mathbf{M}$	Eligible machine set (unordered) of operation O_{vj}
$\mathbf{O} = \cup_{v \in \{1,\dots,P\}} \mathbf{O}_v$	Set of all operations of the problem instance
$o_v \in \mathbb{Z}^+$	Number of operations of product $v, \|\mathbf{O}_v\| = o_v$
$\mathbf{O}_v = \{O_{vj} : j \in \{1, , o_v\}\}$	Ordered set of operations of product variant P_v
O_{vj}	j-th operation of product v
p_{vjk}	Unit processing time of operation O_{vj} on machine M_k in [timesteps]
$\mathbf{P}$	Product portfolio
P_v	Product
$v \in \mathbf{P}$	Index of product variants

Table 7 Notations and variables of the subsystem "Production Orders and Jobs"

Symbol	Designation
$C_{\sigma i}$	Completion time of job $J_{\sigma i}$ in [timestep]
$C_{\sigma ij}$	Completion time of job operation $O_{\sigma ij}$ in [timestep]
d_{σ}	Due date of $\mathbf{PO}_{\sigma}$
$d_{\sigma i}$	Due date of jobs $J_{\sigma i} \in \mathbf{PO}_{\sigma}$
i	Running index of the jobs comprised by a production order
$\mathbf{J} = \cup_{\sigma \in \{1,\dots,Q\}} \mathbf{PO}_{\sigma}$	Unordered set of all jobs of the problem instance
$J_{\sigma i}$	job Belonging to $\mathbf{PO}_{\sigma}$
$n \in \mathbb{Z}^+$	Total number of jobs $J_{\sigma i}$ in the problem instance, $\lvert\mathbf{J}\rvert = n$
$N \in \mathbb{Z}^+$	Total number of job operations in the problem instance
$\mathbf{O}$	Set of all job operations of the problem instance
$o_{v(\sigma)} \in \mathbb{Z}^+$	Number of job operations comprised by job $J_{\sigma i}$, $J_{\sigma i},\ \lvert\boldsymbol{O}_{\sigma i}\rvert = \left\lvert\boldsymbol{O}_{v(\sigma)}\right\rvert = o_{v(\sigma)}$
$\boldsymbol{O}_{\sigma i} = \left\{O_{\sigma i} : j \in \left\{1, \dots, o_{v(\sigma)}\right\}\right\}$	Ordered set of operations of job $J_{\sigma i}$
O_{ij}	j-th operation of job $J_{\sigma i}$
$p_{\sigma ijk}$	unit Processing time of operation $O_{\sigma ij}$ on machine M_k
$\mathbf{PO}_{\sigma}$	Production order, i.e. set of q_{σ} jobs
Q	Total number of $\mathbf{PO}_{\sigma}$ in the problem instance
q_{σ}	Production quantity requested by $\mathbf{PO}_{\sigma}$
$S_{\sigma ij}$	Starting time of operation $O_{\sigma ij}$ in [timestep]
$v(\sigma) \in \mathbf{P}$	Product variant requested by PO_{σ}

8.2 Algorithms

Algorithm 1: Crossover in Multi-Objective Stage S2

input: *parent population*, *parameters*

choose $parent1$ and $parent2$ out of the parent population
split the segments of $parent1$ and $parent2 \rightarrow (OS1, MA1, MP1)$ and $(OS2, MA2, MP2)$
get the crossover rate p_c from *parameters*
while *new population* $< PopSize$:
    choose a *random float number* between 0 and 1
    if *random number* $< p_c$:
        choose randomly between $True$ or $False$
        if $True$**:**
            do crossover on OS segments of $parent1$ and $parent2$
            do crossover on MA segments of $parent1$ and $parent2$
        else:
            do crossover on MP segments of $parent1$ and $parent2$
        add $offspring1$ and $offspring2$ to *new population*
    else:
        add $parent1$ and $parent2$ to *new population*
end while

output: *new population*

Note that $S1$ adopts the same procedure, except the random choice on line 7 is always *True*, i.e. lines 11 and 12 are never visited. For implementation, [38] was adopted.

Algorithm 2: Algorithm for selection of parent chromosomes

input: $population$, $fitness\ function$, $parameters$

do elitist selection:
  get the fitness of the individuals using the $fitness\ function$
  sort the individuals of $population$ according to their fitness
  get the reproduction rate p_r from $parameters$
  add the best $p_r * PopSize$ chromosomes to $parent\ population$
while $parental\ population < PopSize$:
  do tournament selection:
    randomly select b chromosomes out of $population$
    add the fittest chromosome to $parent\ population$
end while

output: $parental\ population$

Algorithm 3: Predictive simulation of schedules

input: $chromosome$, $parameters$, HA_CVAE, DS_CVAE

decode $chromosome$ using Algorithm 7-1 $\rightarrow decoded$
translate $decoded$ into one-hot-encoded (OHE) operating regime sequences $\rightarrow decoded\_OHE\_OpRegs$
for each $machine$:
  get the OHE operating regime sequence of the machine from $parameters$ $\rightarrow machine\_OHE\_opReg$
  get the $HI_{k,history}$ of the machine $\rightarrow machine\_HI\_history$
  get the $HI_{k,safe}$ of the machine $\rightarrow machine\_HI\_safe$
  get the $HI_{k,fail}$ of the machine $\rightarrow machine\_HI\_fail$
  get $machine\_predictions$ by $call\_PHM\_module$
  append $machine\_predictions$ to $predictions$

output: $predictions$

Algorithm 4: Decoding algorithm of the chromosomes

input: $chromosome$, $parameters$

split the segments of $chromosome \rightarrow (OS, MA, MP)$
for operation **in** OS**:**
    check which job operation the allele corresponds
    get the processing time and the product type of the operation from $parameters$
    check MA to get the assigned machine
    if the last job on the assigned machine is not of the same product type:
        plan machine setup before the operation
    if the previous operation of the job is not on the same machine:
        consider transportation time of the job between machines
    check MP if a PdM action is scheduled after the operation
    get the earliest starting time of the operation place on the assigned machine
    update $machine\_operations[machine]$
    if setup planned:
        get the starting time of the setup
        update $machine\_setups[machine]$
    if PdM action scheduled:
        get the starting time of the PdM action
        update $maintenance\_actions[machine]$

output: $machine\_operations$, $machine\_setups$, $maintenance\_actions$

8.3 Predicting Future Operation-specific Health Indicator

Training Two CVAEs are trained to predict the future health indicator given a production schedule. The Data Simulator (DS) is trained on all historical CM data of a run to failure, i.e. from the maintenance action until the failure. The Health Assessor (HA) is trained on "healthy" data only, i.e. the first 20% of data of the run to failure.

Application Both DS and HA are jointly applied. The decoder part of the DS-CVAE is used as a generative model to create simulated sensor signals (i.e. CM data) conditioned on a given future production sequence. This simulated data is in turn assessed by the HA-CVAE to derive an operation-specific HI prediction given a future production schedule (see Fig. 6).

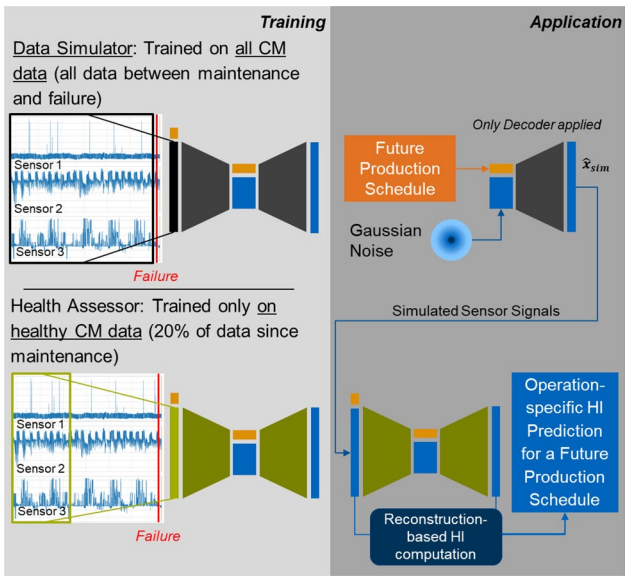

Fig. 6 Health assessment and prediction procedure as outlined by [4]

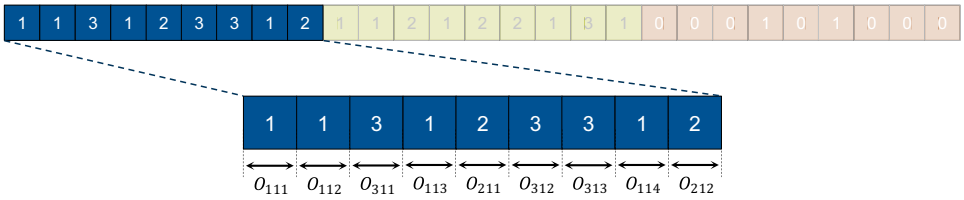

Fig. 7 Encoding of the OS segment of chromosomes

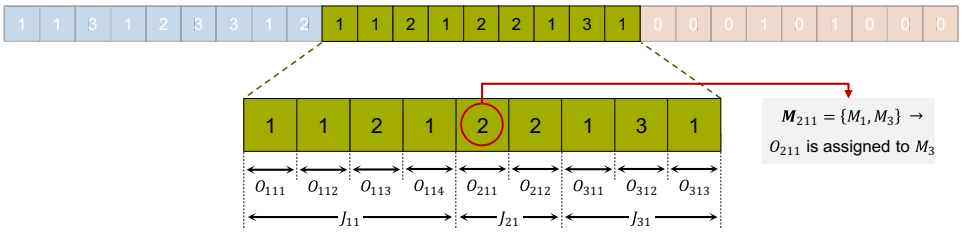

Fig. 8 Encoding of the MA segment of chromosomes

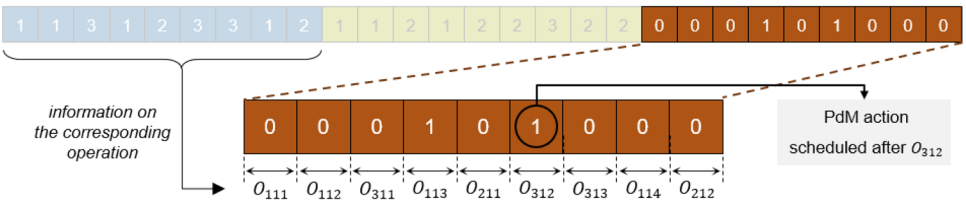

Fig. 9 Encoding of the *MP* segment of chromosomes

8.4 GA Encoding

Encoding of the OS segment The *OS* segment of the chromosome adopts a permutation-based representation as visualized in Fig. 7.

The operation sequencing decision variables $Y_{oij,o'i'j',k}$ are implicitly expressed in this segment and are to be decided by the decoding algorithm. By scanning through the *OS* segment from left to right, the j-th appearance of an index refers to the j-th operation O_{oij} of the job J_{oi}, to which the index belongs. The main advantage of this representation is that each chromosome represents a feasible schedule [31]. Furthermore, the precedence constraints between operations are not explicitly expressed in the chromosomes. The decoding algorithm tracks the number of appearances of an index in the *OS* segment and determines the corresponding job operation for each gene.

Encoding of the MA segment The *MA* segment contains information concerning the machine assignments of job operations. It adopts a partitioned permutation of job indices as visualized in Fig. 8.

Encoding of the MP segment The *MP* segment contains information regarding the PdM actions and adopts the binary encoding proposed by [7] for maintenance planning (see Fig. 9). An allele of *1* means that a PdM action scheduled after the completion of the job operation, directly corresponding to the to the decision variable Z_{oijk}.

8.5 Selected Hyperparameters

The selected values of these hyperparameters are those often encountered in the reviewed literature and are listed in the following Table 8.

11 possible combinations of the shown levels of each hyperparameter were tested. Each setting was run five times by the algorithm. The computations have been performed on a macOS High Sierra [Version 10.13.6] operating system with 2.4 GHz Intel Core i5 processor and 8 GB RAM. The behavior of the algorithm in each run was recorded with regard to the following performance measures:

Table 8 GA hyperparameter levels

Hyperparameter	Level 1	Level 2	Level 3	Level 4
MaxGen	100	200	300	400
PopSize	100	200	300	400
p_c	0.75	0.80	0.85	0.90
p_m	0.05	0.10	0.15	0.20
p_r	0.005	0.01	0.05	0.10

Table 9 Machine data for Case 1

M_k	$HI_k(t=0)$	$HI_{k,fail}$	$HI_{k,safe}$
M_1	0.7912	0.45	0.55
M_2	0.6579	0.45	0.55
M_3	0.7144	0.45	0.55
M_4	0.8777	0.45	0.55
M_5	0.7174	0.45	0.55

- obtained objective function value, i.e. the makespan in timesteps and
- the computational run time in seconds.

8.6 Machine and Product Data for Case 1 & 2

The cost and time parameters used in case 1 are listed in the following. Note that the same parameters were also used for case 2 (see Table 12).

Table 10 Product data in Case 1

Product variant P_v	Operation O_{vj}	Processing time p_{vj}	Eligible Machine Set $\mathbf{M}_{vj}$
P_1	O_{11}	$p_{11}=10$	$\mathbf{M}_{11}=\{M_1,M_4,M_5\}$
	O_{12}	$p_{12}=8$	$\mathbf{M}_{12}=\{M_1,M_2,M_3\}$
	O_{13}	$p_{13}=10$	$\mathbf{M}_{13}=\{M_1,M_2,M_3\}$
	O_{14}	$p_{14}=4$	$\mathbf{M}_{14}=\{M_1,M_3\}$
P_2	O_{21}	$p_{21}=8$	$\mathbf{M}_{21}=\{M_1,M_2,M_3,M_4,M_5\}$
	O_{22}	$p_{22}=10$	$\mathbf{M}_{22}=\{M_1,M_2,M_3,M_4,M_5\}$
	O_{23}	$p_{23}=6$	$\mathbf{M}_{23}=\{M_1,M_3,M_4\}$
P_3	O_{31}	$p_{31}=7$	$\mathbf{M}_{31}=\{M_1,M_2,M_3\}$
	O_{32}	$p_{32}=9$	$\mathbf{M}_{32}=\{M_1,M_5\}$
	O_{33}	$p_{33}=10$	$\mathbf{M}_{33}=\{M_1,M_2,M_3\}$
P_4	O_{41}	$p_{41}=9$	$\mathbf{M}_{41}=\{M_1,,M_4,M_5\}$
	O_{42}	$p_{42}=10$	$\mathbf{M}_{42}=\{M_1,M_2,M_3\}$
	O_{43}	$p_{43}=9$	$\mathbf{M}_{43}=\{M_1,M_2,M_3\}$
P_5	O_{51}	$p_{51}=11$	$\mathbf{M}_{51}=\{M_1,M_2,M_3,M_4,M_5\}$
	O_{52}	$p_{52}=9$	$\mathbf{M}_{52}=\{M_1,M_2,M_3\}$
	O_{53}	$p_{53}=10$	$\mathbf{M}_{53}=\{M_1,M_5\}$

Table 11 Product order data in Case 1

PO_o	$v(o)$	q_o	d_o
PO_1	P_1	2	100
PO_2	P_2	4	150
PO_3	P_1	2	200
PO_4	P_3	2	240
PO_5	P_4	1	240
PO_6	P_5	2	280
PO_7	P_3	3	320
PO_8	P_2	4	400

Table 12 Cost and time parameters in Case 1 and 2

Cost component	Time parameter [timesteps/action]	Cost parameter [€/action]
PdM action	$t_{PdM}=60$	$cost_{PdM,fix}=400$
		$cost_{PdM,adv}=50$
Setup	$t_{setup}=15$	$cost_{setup}=50$
Transportation	$t_{transport}=10$	$cost_{transport}=10$

Case 2: Total FJSSP with a real industrial dataset

Detailed data of the PdM-FJSSP instance studied in Case 2 (see Sect. 6.2) are given below. Note that the indices operations and processing durations are separated by a comma, as some comprise two digits in this case (see Table 13, 14 and 15).

Table 13 Machine data for Case 2

M_k	$HI_k(t=0)$	$HI_{k,safe}$	$HI_{k,fail}$
M_1	0.8963	0.80	0.75
M_2	0.8503	0.80	0.75
M_3	0.8406	0.80	0.75

Table 14 Product data of Case 2

Index	P_v	O_{vj}	p_{vj}
[1]	P_1	$O_{1,1}$	$t_{1,1} = 15$
[2]	P_2	$O_{2,1}$	$t_{2,1} = 12$
[3]	P_3	$O_{3,1}$	$t_{3,1} = 18$
[4]	P_4	$O_{4,1}$	$t_{4,1} = 12$
[5]	P_5	$O_{5,1}$	$t_{5,1} = 22$
[6]	P_6	$O_{6,1}$	$t_{6,1} = 10$
[7]	P_7	$O_{7,1}$	$t_{7,1} = 14$
[8]	P_8	$O_{8,1}$	$t_{8,1} = 10$
[9]	P_9	$O_{9,1}$	$t_{9,1} = 15$
[10]	P_{10}	$O_{10,1}$	$t_{10,1} = 22$

Table 15 Product order data of Case 2

PO_o	$v(o)$	q_o	d_o	PO_o	$v(o)$	q_o	d_o
PO_1	P_1	8	140	PO_7	P_8	3	460
PO_2	P_2	4	200	PO_8	P_6	2	460
PO_3	P_3	2	280	PO_9	P_2	3	480
PO_4	P_4	4	350	PO_{10}	P_3	5	500
PO_5	P_7	6	380	PO_{11}	P_9	3	500
PO_6	P_5	2	400	PO_{12}	P_{10}	8	530

Funding Open Access funding enabled and organized by Projekt DEAL.

References

1. Zhai S, Achatz S, Groher M et al. (2020) An empirical expert study on the status quo and potential of predictive maintenance in industry. In: International Conference on Sensing, Diagnostics, Prognostics, and Control (SDPC). https://doi.org/10.1109/SDPC49476.2020.9353177
2. Iyer N, Goebel K, Bonissone P (2006) Framework for Post-Prognostic Decision Support. In: 2006 IEEE Aerospace Conference proceedings: March 4–11, 2006, Big Sky, MT. [IEEE], [Piscataway, N.J.], pp 1–10
3. Zhai S, Reinhart G (2018) Predictive maintenance als Wegbereiter für die instandhaltungsgerechte Produktionssteuerung. ZWF 113:298–301. https://doi.org/10.3139/104.111912
4. Zhai S, Gehring B, Reinhart G (2021) Enabling predictive maintenance integrated production scheduling by operation-specific health prognostics with generative deep learning. J Manuf Syst 104:284. https://doi.org/10.1016/j.jmsy.2021.02.006
5. Ghaleb M, Taghipour S, Sharifi M et al (2020) Integrated production and maintenance scheduling for a single degrading machine with deterioration-based failures. Comput Ind Eng. https://doi.org/10.1016/j.cie.2020.106432
6. Cassady CR, Kutanoglu E (2005) Integrating preventive maintenance planning and production scheduling for a single machine. IEEE Trans Rel 54:304–309. https://doi.org/10.1109/TR.2005.845967
7. Ye J, Ma H (2015) Multiobjective joint optimization of production scheduling and maintenance planning in the flexible job-shop problem. Math Probl Eng 2015:1–9. https://doi.org/10.1155/2015/725460
8. Chen X, Xiao L, Zhang X (2014) A production scheduling problem considering random failure and imperfect preventive maintenance. Proc Instit Mech Eng Part O: J Risk Reliab 229(1):26–35. https://doi.org/10.1177/1748006X14545834
9. Bougacha O, Varnier C, Zerhouni N et al. (2019) Integrated Production and Predictive Maintenance Planning based on Prognostic Information. In: 2019 International Conference on advanced systems and emergent technologies (IC_ASET). IEEE, pp 363–368
10. Sloan TW, Shanthikumar JG (2010) Combined production and maintenance scheduling for a multiple-product, single-machine production system. Prod Oper Manag 9:379–399. https://doi.org/10.1111/j.1937-5956.2000.tb00465.x
11. Aramon Bajestani M, Banjevic D, Beck JC (2014) Integrated maintenance planning and production scheduling with Markovian deteriorating machine conditions. Int J Prod Res 52:7377–7400. https://doi.org/10.1080/00207543.2014.931609
12. Pan E, Liao W, Xi L (2012) A joint model of production scheduling and predictive maintenance for minimizing job tardiness. Int J Adv Manuf Technol 60:1049–1061. https://doi.org/10.1007/s00170-011-3652-4
13. Fitouri C, Fnaiech N, Varnier C et al (2016) A Decison-making approach for job shop scheduling with job depending degradation

and predictive maintenance. IFAC-PapersOnLine 49:1490–1495. https://doi.org/10.1016/j.ifacol.2016.07.782
14. Ladj A, Benbouzid-Si Tayeb F, Varnier C et al (2017) A hybrid of variable neighbor search and fuzzy logic for the permutation flowshop scheduling problem with predictive maintenance. Proc Comput Sci 112:663–672. https://doi.org/10.1016/j.procs.2017.08.120
15. Ladj A, Tayeb FB-S, Varnier C et al. (2019) Improved genetic algorithm for the fuzzy flowshop scheduling problem with predictive maintenance planning. In: 2019 IEEE 28th International Symposium on industrial electronics (ISIE). IEEE, pp 1300–1305
16. Benaggoune K, Meraghni S, Ma J et al. (2020) Post prognostic decision for predictive maintenance planning with remaining useful life uncertainty. In: 2020 Prognostics and Health Management Conference (PHM-Besançon). IEEE, pp 194–199
17. Zhai S, Riess A, Reinhart G (2019) Formulation and solution for the predictive maintenance integrated job shop scheduling problem, pp 1–8. https://doi.org/10.1109/ICPHM.2019.8819397
18. Morariu C, Morariu O, Răileanu S et al (2020) Machine learning for predictive scheduling and resource allocation in large scale manufacturing systems. Comput Ind 120:103244. https://doi.org/10.1016/j.compind.2020.103244
19. Paprocka I, Kempa WM, Skołud B (2020) Predictive maintenance scheduling with reliability characteristics depending on the phase of the machine life cycle. Eng Optim 31:1–19. https://doi.org/10.1080/0305215X.2020.1714041
20. Denkena B, Dittrich M-A, Keunecke L et al (2020) Continuous modelling of machine tool failure durations for improved production scheduling. Prod Eng Res Devel 14:207–215. https://doi.org/10.1007/s11740-020-00955-y
21. Khatab A, Diallo C, Aghezzaf E-H et al (2019) Integrated production quality and condition-based maintenance optimisation for a stochastically deteriorating manufacturing system. Int J Prod Res 57:2480–2497. https://doi.org/10.1080/00207543.2018.1521021
22. Zandieh M, Khatami AR, Rahmati SHA (2017) Flexible job shop scheduling under condition-based maintenance: Improved version of imperialist competitive algorithm. Appl Soft Comput 58:449–464. https://doi.org/10.1016/j.asoc.2017.04.060
23. Xiang Y, Cassady CR, Jin T et al (2014) Joint production and maintenance planning with machine deterioration and random yield. Int J Prod Res 52:1644–1657. https://doi.org/10.1080/00207543.2013.843037
24. Ladj A, Varnier C, Tayeb FB-S (2016) IPro-GA: an integrated prognostic based GA for scheduling jobs and predictive maintenance in a single multifunctional machine. IFAC-PapersOnLine 49:1821–1826. https://doi.org/10.1016/j.ifacol.2016.07.847
25. Wang T (2010) Trajectory similarity based prediction for remaining useful life estimation. Dissertation, University of Cincinnati
26. Chaudhry IA, Khan AA (2016) A research survey: review of flexible job shop scheduling techniques. Intl Trans in Op Res 23:551–591. https://doi.org/10.1111/itor.12199
27. Benbouzid-Sitayeb F, Guebli SA, Bessadi Y et al (2011) Joint scheduling of jobs and Preventive Maintenance operations in the flowshop sequencing problem: a resolution with sequential and integrated strategies. IJMR 6:30–48. https://doi.org/10.1504/IJMR.2011.037912
28. Rooyani D, Defersha FM (2019) An efficient two-stage genetic algorithm for flexible job-shop scheduling. IFAC-PapersOnLine 52:2519–2524. https://doi.org/10.1016/j.ifacol.2019.11.585
29. Gao J, Gen M, Sun L (2006) Scheduling jobs and maintenances in flexible job shop with a hybrid genetic algorithm. J Intell Manuf 17:493–507. https://doi.org/10.1007/s10845-005-0021-x
30. Li X, Gao L (2016) An effective hybrid genetic algorithm and tabu search for flexible job shop scheduling problem. Int J Prod Econ 174:93–110. https://doi.org/10.1016/j.ijpe.2016.01.016
31. Gao J, Sun L, Gen M (2008) A hybrid genetic and variable neighborhood descent algorithm for flexible job shop scheduling problems. Comput Oper Res 35:2892–2907. https://doi.org/10.1016/j.cor.2007.01.001
32. Gen M, Tsujimura Y, Kubota E (1994) Solving job-shop scheduling problems by genetic algorithm. In: Proceedings of IEEE International Conference on systems, man and cybernetics. IEEE, pp 1577–1582
33. Zhang G, Zhang L, Song X et al (2019) A variable neighborhood search based genetic algorithm for flexible job shop scheduling problem. Cluster Comput 22:11561–11572. https://doi.org/10.1007/s10586-017-1420-4
34. Huang X, Guan Z, Yang L (2018) An effective hybrid algorithm for multi-objective flexible job-shop scheduling problem. Adv Mech Eng 10:168781401880144. https://doi.org/10.1177/1687814018801442
35. Brandimarte P (1993) Routing and scheduling in a flexible job shop by tabu search. Ann Oper Res 41:157–183. https://doi.org/10.1007/BF02023073
36. Saxena A, Goebel K (2008) turbofan engine degradation simulation data set: NASA Ames Prognostics Data Repository. http://ti.arc.nasa.gov/project/prognostic-data-repository. Accessed 15 Jan 2021
37. KUKA Systems GmbH (2019) Industrie 4.0—Matrix production. https://www.kuka.com/-/media/kuka-downloads/imported/2d0d7c7d6573491a8b43eb5f66aec596/kuka_matrixproduktion_screenpdfs_en_160419.pdf. Accessed 10 Mar 2021
38. Bour G (2018) Python implementation of a genetic algorithm for FJSP [source code]. https://github.com/guillaumebour/flexible-job-shop. Accessed 10 Mar 2021