Journal of Intelligent Manufacturing (2021) 32:1207–1228
https://doi.org/10.1007/s10845-020-01685-9

Field-synchronized Digital Twin framework for production scheduling with uncertainty

Elisa Negri[1] · Vibhor Pandhare[2] · Laura Cattaneo[1] · Jaskaran Singh[3] · Marco Macchi[1] · Jay Lee[2]

Received: 21 April 2020 / Accepted: 29 September 2020 / Published online: 15 October 2020

Abstract

Research on scheduling problems is an evergreen challenge for industrial engineers. The growth of digital technologies opens the possibility to collect and analyze great amount of field data in real-time, representing a precious opportunity for an improved scheduling activity. Thus, scheduling under uncertain scenarios may benefit from the possibility to grasp the current operating conditions of the industrial equipment in real-time and take them into account when elaborating the best production schedules. To this end, the article proposes a proof-of-concept of a simheuristics framework for robust scheduling applied to a Flow Shop Scheduling Problem. The framework is composed of genetic algorithms for schedule optimization and discrete event simulation and is synchronized with the field through a Digital Twin (DT) that employs an Equipment Prognostics and Health Management (EPHM) module. The contribution of the EPHM module inside the DT-based framework is the real time computation of the failure probability of the equipment, with data-driven statistical models that take sensor data from the field as input. The viability of the framework is demonstrated in a flow shop application in a laboratory environment.

Keywords Digital Twin · Equipment health · Fault detection · Simheuristics · Robust scheduling · PHM · FSSP

Introduction

Modern day industries need to compete for profitability and customer satisfaction in a challenging environment with rising costs, attention to performance and reliability, operations safety, and others. Thus, industries are rapidly integrating emerging digital technologies that have ushered companies to a new era of industrial revolution called Industry 4.0 (Lee et al. 2013, 2015a; Oztemel and Gursev 2020; Shi et al. 2011; Wu et al. 2013; Xu et al. 2014; Yang et al. 2015; Zhang et al. 2014a, b).

Industry 4.0 encompasses multiple evolving technology umbrellas, one of which is Cyber-Physical Systems (CPS). Pervasive sensor technologies, open and standardized communication protocols, and computational convenience have led to its development (Baheti and Gill 2011; Lee et al. 2015b; Leitão et al. 2016). CPS can be defined as a synergetic integration of the physical assets and their Digital Twin (DT), usually with feedback loops where the state of the physical assets affects the computations in the DT and vice versa (Lee 2008). Thus, integrating CPS with production, logistics, maintenance, and other services in the current industrial practice holds the potential to transform today's factories into Industry 4.0-based factories. This can foster significant economic growth and high responsiveness to ever-changing operating conditions, leading to major evolutions in the decision-making processes. The present article focuses on production scheduling. It is an activity which has sustained

✉ Elisa Negri
elisa.negri@polimi.it

Vibhor Pandhare
pandhavr@mail.uc.edu

Laura Cattaneo
laura1.cattaneo@polimi.it

Jaskaran Singh
jaskaran.singh@thapar.edu

Marco Macchi
marco.macchi@polimi.it

Jay Lee
lj2@ucmail.uc.edu

[1] Department of Management, Politecnico di Milano, Economics and Industrial Engineering, Milan, Italy

[2] Mechanical and Materials Engineering, College of Engineering and Applied Science, University of Cincinnati, Cincinnati, USA

[3] Department of Mechanical Engineering, Thapar Institute of Engineering and Technology, Patiala, India

continuous research interest due to its complexities—primarily, the scheduling problem being a NP-hard problem, where complications may explode even for small scale instances (Garey and Jonhson 1979; Liu 2016; Zhang et al. 2014a, 2019b).

CPS systematically integrate into the many levels of the decision-aiding automation pyramid in an industry, from high-level business to operations management (Fig. 1). Therein, the operational decisions at production system level are performed by software modules for planning, supervision and control, working together to achieve the goals set by the ERP (Rossit and Tohmé 2018). In the envisaged future, the simultaneous view of the information contained in the different levels of the automation pyramid defines the way in which production is planned and controlled; besides, the collaboration between several modules, with the exchange of real-time information, becomes essential. To enable this future, CPS are integrated to support different functionalities, serving as the building block for the advanced operations in a smart factory (Napoleone et al. 2020). This fosters a research challenge directed towards architecting frameworks to elaborate dynamic production schedules under uncertain scenarios based on real-time industrial data.

The DT is what empowers CPS and is the focus of a wide research stream. It has been defined as a synchronized simulation of a physical asset with a real-time connection with its operating environment or field. Therefore, it supports a number of functionalities such as system mirroring, parameter optimization, behavior prediction, etc. (Lim et al. 2020; Lopes et al. 2019; Negri et al. 2017, 2019b; Uriarte et al. 2018). This is recalled by Fig. 1, showing examples of DT-based functionalities for different application domains (production, energy, quality, maintenance …) in the industrial practice. Moreover, DT represents the virtual counterpart of the physical asset along all its life cycle, then it aids in the management of physical assets by supporting decision-making throughout the lifecycle (Macchi et al. 2018), also using the integration with an intelligence layer (Negri et al. 2020). Essentially being a simulation, the DT is persistent and exists even when disconnected from the physical counterpart (Borangiu et al. 2020). This includes cases in which connection problems are experienced or there is the need to elaborate data with temporary disconnection from the physical asset, e.g. in the case of evaluating possible future scenarios. Kritzinger proposes a clear categorization of the relationship between the DT and its physical twin: (i) the simulation model describing the systems features, rules and relationships is the Digital Model, this is not connected to the field; (ii) by setting up the communication from the physical equipment to the Digital Model, a Digital Shadow is created; this includes the update of sensors values into the digital world in order to have a synchronized simulation; (iii) the full potentialities of a DT are only reached when the communication is bidirectional, feeding back data from the digital to the physical world (Kritzinger et al. 2018). Literature on DT is rich and, although a common vision on the topic is still not achieved, many reviews provide an interpretation of the concept and its potential applications in manufacturing (Cimino et al. 2019; Enders and Hoßbach 2019; Kritzinger et al. 2018; Marmolejo-Saucedo et al. 2019; Orozco-Romero et al. 2019; Wardhani et al. 2018; Zhang et al. 2019a).

This article is directed at contributing to the research on DT in manufacturing, by proposing a proof-of-concept of a dynamic production scheduling framework based on a field-synchronized DT that deals with uncertainty (represented by failure probability) in real-time. It includes various elements, such as genetic algorithms (GA) for optimization, discrete event simulation (DES), the connection with the physical equipment, the analysis of real time data with a statistical model, etc. The focus of this work is on the joint use of two functionalities supported by the DT, namely production scheduling and health monitoring (as in Fig. 1). The innovation lies also in the integration of knowledge from various research fields into a unique framework, namely the efficient and robust production scheduling, the Digital Twin development, and the Prognostics and Health Management (PHM).

The article is correspondingly organized. The literature review, to set the background on scheduling problems, is presented in "Literature review" section. The research objective and proposed contributions are defined in "Research objective and proposed contributions" section. The proposed framework is then described in "Proposed framework" section, and it is implemented through a case study in "Case study" section. The results of the case study are discussed in "Results and discussion" section. The article makes concluding remarks in "Conclusions and future works" section, also envisioning possible future research directions.

Literature review

In the last fifty years there has been a growing interest in scheduling techniques due to the impelling need of industrial companies to optimize an increasingly complex and fragmented production scenario (Vieira et al. 2017). Manufacturing systems have become progressively more complex and the market requires higher levels of adaptability and customization, demanding for a more effective production management. The traditional aim for industrial scheduling applications was the minimization of the total makespan (Allahverdi et al. 2018; Bagheri et al. 2010; Eddaly et al. 2016; Gonzalez-Neira et al. 2017; Hatami et al. 2018; Juan et al. 2014; Pan and Wang 2012); however, more recently, other performances have been taken into the optimization objectives, leading to cost minimization, to total tardiness minimization and others (Behnamian and Zandieh 2011;

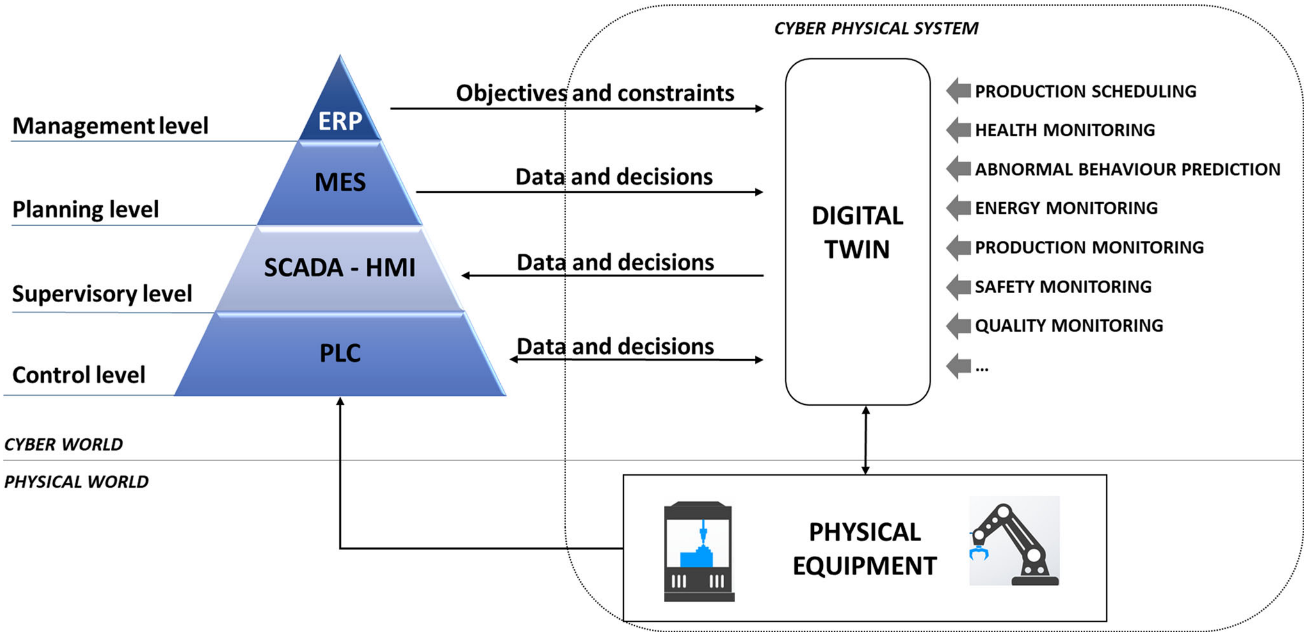

Fig. 1 Role of DT to support the control of manufacturing operations in cyber-physical production systems

Zhang et al. 2017). As the objective is today more comprehensive, the choice of scheduling techniques has become even more crucial (Vieira et al. 2017). Along this evolution, two main topics are of interest for this article: (i) the use of metaheuristics for schedule optimization ("Metaheuristics for schedule optimization" section), and (ii) the robustness of scheduling solutions, in case of uncertain scenarios ("Robustness of scheduling solutions" section).

Metaheuristics for schedule optimization

Until the 1960s research on scheduling was limited to small sized problems (Gupta and Stafford 2006). This was mainly due to the: (i) lack of computing power, (ii) inefficient computer programs and (iii) complexity of most real industrial problems. In particular, many authors classify the majority of scheduling problems as NP-hard problems (Neufeld et al. 2016). This means that the computational effort required for obtaining a solution increases exponentially with the problem size, and even small scale instances may require a long time to converge to an optimal solution (Zhai et al. 2017). This caused a change of direction in the search for methods of resolution. There was in fact an opening towards exploring metaheuristics methods and the development of specific algorithms to solve scheduling problems by searching for a 'good' near-optimal solution, instead of an absolute optimum.

GA have been extremely popular amongst the metaheuristics for production scheduling problems. In addition, more recently simulation has been also proposed: (i) to predict the effects of changes in existing systems without having to reproduce them on the real system; (ii) to predict in advance the performances of systems that have yet to be built. Different kinds of simulation methods can be then coupled with optimization algorithms for production scheduling (Vieira et al. 2017). When coupling simulation and optimization algorithms, simulation results are fed to the optimization algorithm for the computation of the objective function. In literature, the union of a metaheuristic algorithm and a simulation model is widely indicated as simheuristics (Gonzalez-Neira et al. 2017; Hatami et al. 2018; Juan et al. 2014, 2015). A simheuristic model based on GA is chosen to be used in the present work.

Robustness of scheduling solutions

As industrial contexts embed a certain degree of uncertainty, random events may affect the actual adherence to an established production schedule. For example, an increased processing time in one machine can lead to the imposition of delays in all subsequent production steps. Robustness has been defined as the capability to handle small delays, resist to imprecision, tolerate a certain degree of uncertainty, and deal with unexpected disruptions without having to thoroughly modify the production schedule. Thus, a schedule can be defined as robust when it minimizes the impact of disruptions: not necessarily it is the best under a specific performance, but it is the one that performs better in terms of realistic implementation of operations (Vieira et al. 2017; Wu et al. 2018; Zandieh et al. 2010). To cope with uncertainties, two approaches are widely explored (Dias and Ierapetritou 2016; Herroelen and Leus 2005).

- In *Reactive* approach, decisions at scheduling and control level are taken using a model that does not consider uncertainties. When an issue occurs, new information is fed back to the scheduler to find a new solution. This assumes that a continuous recalculation must take place whenever a change in real production is registered and, consequently, it turns out to be computationally expensive. Moreover, feasibility is not guaranteed as decisions taken previously, and already implemented, may compromise the viability of future schedules.
- In *Preventive* approach, decisions at scheduling and control level are taken using a model that considers uncertainties. This approach aims at finding robust solutions that ensure feasibility and high performances even in presence of disruptions and is thus preferred over the reactive approach.

A simheuristics approach enables to pursue a preventive approach for robust scheduling. Numerous works in literature implement it. (Wu et al. 2018) propose a GA coupled with Monte Carlo simulation in order to optimize the schedule for a Job Shop Scheduling Problem (JSSP) with random machine breakdowns, evaluating the risk based on makespan delay. (Aramon Bajestani and Beck 2015) design an iterative two-stage algorithm to solve a Flow Shop Scheduling Problem (FSSP), considering machine maintenance and minimizing costs. (Gonzalez-Neira et al. 2017) solves a stochastic FSSP, where processing and assembly times are random variables. They couple biased randomization and simulation also using metaheuristics and propose different alternatives to find the final solution. Another solution for handling stochastic processing times can be found in (Hatami et al. 2018) which uses an Iterated Local Search (ILS) algorithm to generate a sequence of solutions in an iterative way. It focuses on recommending starting time of operations, to complete all the components of products on a given deadline with a certain input probability. The assessment is made calculating possible deterministic solutions with Monte Carlo simulation and, in this way, comparing and assessing the stochastic ones. (Framinan et al. 2019) consider non-deterministic processing times and aim at rescheduling the remaining jobs in a flowshop when production is already started, considering new available information. In this work, the authors conclude that an event-driven and heuristics-based rescheduling process can improve performances only in case of low variability. (Juan et al. 2014) propose an ILS and Monte Carlo-based simheuristic approach to deal with the stochastic behavior of the operations in a flowshop with stochastic processing times. The authors assume that a correlation exists between deterministic and stochastic solutions, meaning that a good deterministic solution is likely to be also a good solution in a stochastic scenario.

Table 1 reports a summary of the reviewed articles on robust scheduling, evaluating the factors that are considered in the articles in terms of (1) use of simulation; (2) modeling of uncertainty—regarding stochasticity in processing times, times between failures and times for maintenance activities; and (3) chosen metaheuristics algorithm—GA or a different algorithm. The table highlights that the use of simulation and the modelling of uncertainty are common to many articles, whereas most works do not consider all the three factors simultaneously. Other gaps emerge from literature, worth of a discussion in view of the aim of the present work.

1. The articles that consider simulation for robust scheduling do not rely on a DT-based approach to simulation, but only use non-synchronized data from historical datasets. Considering also those works that model the maintenance times stochasticity, they do not use real-time data from field (which are necessary to build a DT-based approach). The only authors who propose the use of real-time data for robust scheduling are (Framinan et al. 2019). This is a clear gap emerged from literature (Gap #1).
2. From the reviewed literature, it appears that right techniques to properly model uncertainty for robust scheduling—both considering production and maintenance management requirements—are already available and consolidated. This is a relevant background to switch the focus on the outcomes: the interest now lies in the robustness of the scheduling solution itself, and the impact of uncertainty on the evolution of the optimization algorithm searching for the solution. This is also stated by (Lee 2018), who recognizes the need to investigate the behavior of the GA fitness function in relation to the stability and robustness of systems to uncertainties (Gap #2).
3. From the reviewed literature, it also emerges that several works consider uncertain events in production environments, even if they solve theoretical scenarios without presenting implementations in real industrial cases or laboratory demonstrations. Therefore, none of them proposes a way to synchronize with field data in real time for data-driven analytics, be it AI technologies and statistical modelling (Gap #3).

Overall, from a practical point of view it can be noted that there is an excess of attention in the rigor of mathematical formulation of the proposed solutions at the cost of neglecting the realism of the industrial problems addressed. This is also witnessed by two of the analyzed articles that wish for problems inspired by real life situations, rather than encountered in mathematical abstractions (Gupta and Stafford 2006; Juan et al. 2015).

Table 1 Literature review on robust scheduling

Article	Sim	Uncertainty			Algorithm	
		Stoch PT	Fail	Maint	GA	Other
Abedinnia et al. (2017)	X	X				
Allahverdi et al. (2018)						X
Aramon Bajestani and Beck (2015)			X	X		
Bagheri et al. (2010)						X
Della Croce et al. (1995)					X	
Dias and Ierapetritou (2016)	X	X	X	X		X
Eddaly et al. (2016)						X
Falkenauer et al. (1991)					X	
Framinan et al. (2019)		X	X			X
Framinan et al. (2014)						
Gonzalez-Neira et al. (2017)	X	X				X
Gupta and Stafford (2006)		X			X	X
Hatami et al. (2018)	X	X				X
Johnson (1954)						
Juan et al. (2014)	X	X				X
Juan et al. (2015)	X	X	X		X	X
Krug et al. (2002)	X				X	X
Lee (2018)	X	X	X	X	X	
Lolli et al. (2017)		X				
Neufeld et al. (2016)		X			X	X
Nguyen and Bagajewicz (2008)	X		X	X	X	
Pan and Wang (2012)						X
Pessoa and Andrade (2018)						X
Teschemacher and Reinhart (2016)			X	X		X
Uslu et al. (2019)					X	X
Vieira et al. (2017)	X	X	X	X	X	
Wu et al. (2018)		X	X	X	X	
Yu and Seif (2016)			X	X	X	

Sim = Simulation; Stoch PT = Stochastic Processing Times; Fail = Times between failures; Maint = Maintenance activities times; GA = Genetic Algorithm; Other = Other algorithms

Research objective and proposed contributions

Despite increasingly sophisticated algorithms have been proposed with the capability to solve more and more complex scheduling problems assisted by an always greater computing power, the availability of data and information from the industrial equipment nowadays is not exploited to the extent it could. The literature analysis in fact demonstrates that the intersection between optimization algorithms and available data and information from field is not thoroughly explored. This work then aims at contributing to the DT research in manufacturing by proposing a proof-of-concept of a DT-based simheuristic framework for the flowshop scheduling that considers the uncertainty detected in real-time and related to the failure probability of equipment. The framework is intended to support the decision makers in manufacturing companies to evaluate and rank the most robust schedules for flowshops.

The proposed framework copes with the found gaps in literature, as follows:

- it is based on a field-synchronized DT that uses real-time data coming from the industrial equipment to simulate the alternative job sequences in the flowshop; it addresses Gap #1, stating that most scheduling applications that consider uncertainty in literature use historical data for the calculation of optimal solutions, and do not consider the use of real-time data;
- it copes with the uncertainties of a real flowshop through a data-driven analytical capability integrated in the DT to address (1) health of the industrial equipment through an EPHM module; (2) associated maintenance times upon prediction of failure; (3) stochastic job processing times;

besides Gap #1, it addresses Gap #3, stating that none of analyzed works proposed a way to synchronize with field data in real-time to use them in the data-driven analytics supporting the scheduling activity;

- it optimizes the performance of the production system through a GA, in which the optimization function also ensures the robustness of the optimal solutions with uncertain conditions due to failures; it addresses Gap #2, stating that the research interest lies in searching the robustness of the scheduling solution and the impact of uncertainty on the evolution of the optimization algorithm.

Proposed framework

The proposed framework is composed of multiple modules that interact to find robust scheduling solutions, both using historical and real-time data coming from the connected industrial equipment. These modules are represented in Fig. 2:

- Physical flowshop system, with the possibility to synchronize the digital information with the physical equipment;
- Input data module, illustrated in "Input data" section;
- Optimization module based on GA, illustrated in "Optimization module" section;
- Digital Twin module, composed of:
  - Equipment PHM—or EPHM—module, illustrated in "Equipment PHM module" section;
  - Simulation model, illustrated in "Simulation model" section;
- A human–machine interface, an example of which is briefly discussed in the results section, "Human machine interface" section.

The framework is in line with previous work presented in (Fumagalli et al. 2018, 2017; Negri et al. 2019a). The present article extends the framework description, the theoretical background, and the experimental results.

Input data

The input data for the scheduling of a production system include: (i) the list of jobs to be produced; (ii) the list of operations needed for each job; (iii) the list of available resources as physical assets capable to perform the operations; (iv) any additional technological or managerial constraint to be considered when scheduling. The production objectives of the company for scheduling should be also declared here, i.e. what performance indicators the company wants to optimize. Finally, to be able to properly model the behavior of the industrial equipment, historical equipment data needs to be input to the framework at this point for statistical processing.

Optimization module

The optimization module is based on GA, to provide different functionalities (according to well-known background (Castelli et al. 2019; Falkenauer et al. 1991; Uslu et al. 2019)):

- at the beginning of the scheduling activity, the GA generates an initial population in a random way;
- then, the GA oversees the assessment of each individual of the population according to a fitness function: with the individual assessment, each individual is evaluated separately through a fitness function; the data needed for this function come from the interaction with the simulation model within the DT module;
- if the optimal solution has been reached, a selection of the best individuals is communicated to the human–machine interface; otherwise, genetic operators are applied by the GA to generate a new population, thus creating a new iteration of the algorithm.

More details on the proposed GA, as implemented in the case study, are described in "Proposed genetic algorithm" section.

Digital Twin module

The DT module is composed by a DES model of the production system to be scheduled, and an Equipment PHM (EPHM) module. The EPHM module receives real-time data from the physical equipment and elaborates them to feed predictive information to the simulation model of the production system. Therefore, the DT synchronization with the field is done through the EPHM module. Details on these two components of the DT module are provided in this section.

Equipment PHM module

The EPHM module accounts for the uncertainty of equipment failure. Although PHM primarily encompasses three aspects—fault detection, fault diagnosis, and failure prediction, the current work focuses only on integrating the fault detection and failure probability quantification for production scheduling. In a future work the fault diagnosis and the failure prediction capabilities will be included. Two aspects must be considered and identified to develop the EPHM module: (i) the operating states of the industrial equipment as physical twin; (ii) the critical components and failure modes of the equipment.

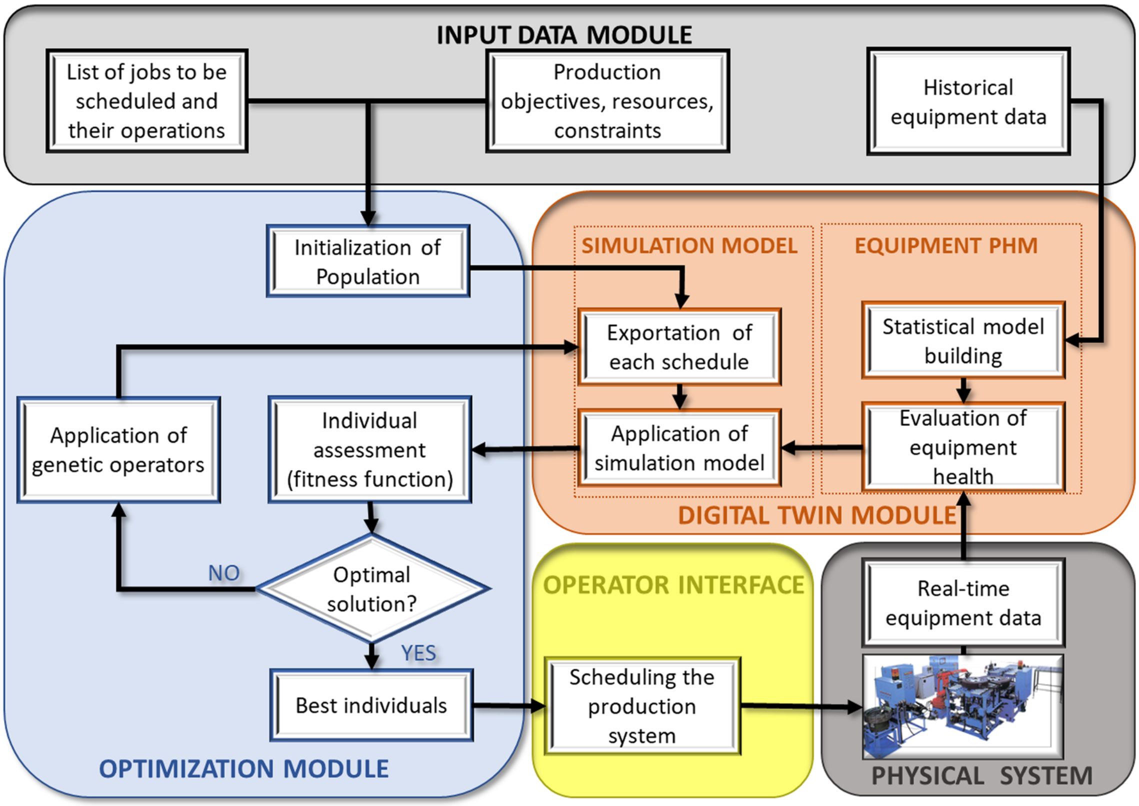
INPUT DATA MODULE
List of jobs to be scheduled and their operations
Production objectives, resources, constraints
Historical equipment data
Initialization of Population
SIMULATION MODEL
EQUIPMENT PHM
Exportation of each schedule
Statistical model building
Application of genetic operators
Individual assessment (fitness function)
Application of simulation model
Evaluation of equipment health
DIGITAL TWIN MODULE
NO
Optimal solution?
YES
OPERATOR INTERFACE
Real-time equipment data
Best individuals
Scheduling the production system
OPTIMIZATION MODULE
PHYSICAL SYSTEM

Fig. 2 Digital Twin-based simheuristics framework for robust scheduling

(i) An industrial equipment can exist either in a healthy state, a degraded state, or a failed state. Mechanical systems, especially rotating machinery and systems involving friction, are more likely to show a gradual trend towards failure, whereas other types of equipment, especially electrical components, have a predominantly immediate failure without an indicative trend. Thus, it is important to note that not all equipment necessarily exists in a degraded state. A system is said to be existing in a healthy state if it is available to perform its intended process and operates according to the designed normal behavior. A system is said to be degraded when it still has the capacity to perform the operation, however its operation/outputs may lie beyond the expected normal limits but are acceptable. A failed state of a system is one where either the system has lost its capacity to perform the function or the outputs of the system are not acceptable.

(ii) Monitoring an industrial equipment using data in real-time can be costly in terms of sensors, data management, data acquisition frequency, and personnel time. Industry best practices call for the identification of critical components within a system which not only justifies the cost of monitoring, but also ensures the detectability of failures. Additionally, appropriate signals either from sensors or controllers and data acquisition parameters need to be chosen for effective collection of useful data. Moreover, a critical equipment can fail with multiple failure modes, it is then necessary to identify the failure modes of interest and their suitability for monitoring. Maintenance logs, Failure Mode, Effects, and Criticality Analysis (FMEA/FMECA) and Expert knowledge play a key role in selecting the failure modes that qualify to be monitored (Utne et al. 2012).

The overall architecture of the EPHM module for the selected failure modes is given in Fig. 3.

In this work, supervised learning is used to develop a model for fault detection and failure probability quantification. This assumes that the signature of the failure in terms of the signals collected is known and is a generalized PHM approach of monitoring a physical asset in manufacturing. The dataset comprises useful signals from healthy and failed state of a critical component along with its labels as ‘healthy’ and ‘faulty’. The primary step before starting to build any analytical model is data pre-processing. This includes data cleaning, data segmentation, and any other operation as needed. Further, predefined features are extracted from the pre-processed dataset. Features can include time-domain features such as mean, standard deviation, kurtosis, etc.

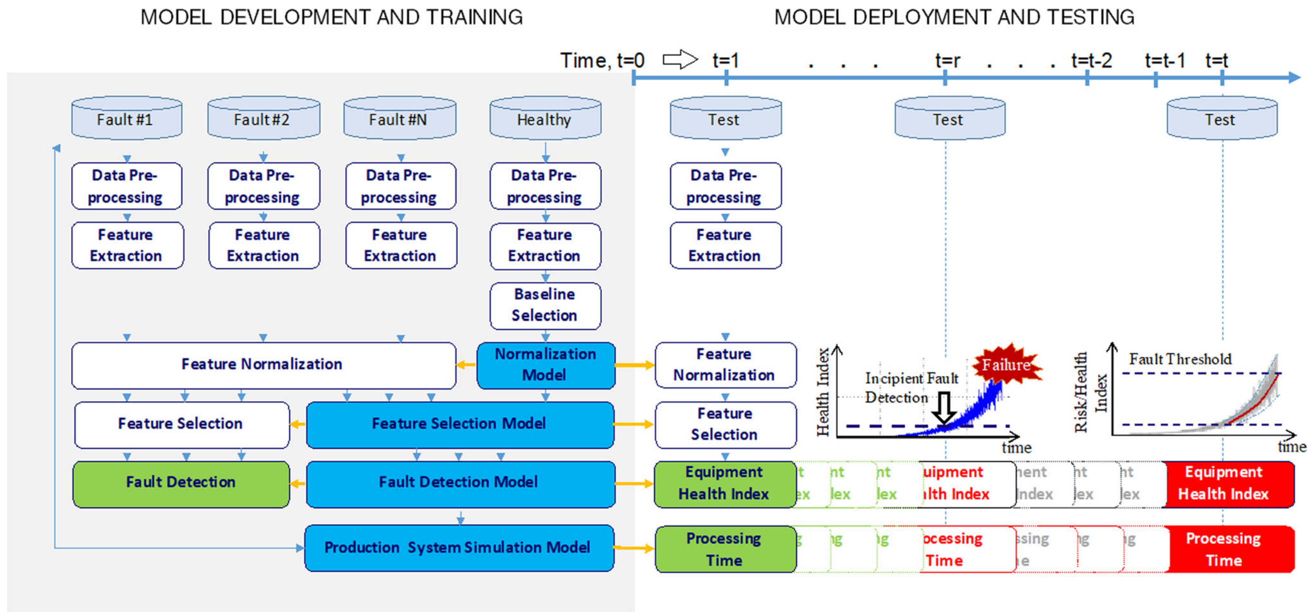

Fig. 3 Schema of the EPHM module

and frequency domain features such as amplitude of frequency spectrum at rotation frequency, at its first harmonic, etc. for equipment of rotating nature. Advanced features in time–frequency domain, using Short-time Fourier Transform or Wavelets or specially designed features, can also be used. A portion of samples from the healthy state feature-set is selected as the baseline for forming the fault detection model. The healthy feature samples are normalized to mean = 0, and standard deviation = 1, and the obtained parameters are saved as the normalization model. The remaining portion of the healthy state feature-set and all the faulty state feature-sets form the validation set for fault detection. The validation feature-sets are normalized using the normalization model.

Feature selection/compression techniques are used to reduce the dimensionality of the feature vectors for each sample preserving the amount of information present in the vectors. Fisher criteria or Principal Component Analysis (PCA) is typically used for this purpose. Using the final feature-set, a fault detection model can be built. Regression models are a good choice for the current application as they can allocate the output as ‘0’ for healthy samples, and ‘1’ for faulty samples. These models allow for a range of values between 0 and 1, generally called as the failure conformance value (CV) or equipment health index. A Logistic regression model can be directly trained on the features to obtain the equipment health index. In other approaches, an unsupervised learning method can also be used to obtain deviation metrics between baseline and faulty samples, such as PCA with Hotelling T^2 metric or Self-Organizing Map with Minimum Quantization Error. These deviation metrics can then be mapped between 0 and 1 using regression methods to obtain the equipment health index.

Summarizing, every incoming test sample undergoes preprocessing, validating the quality of data and performing data segmentation. Predefined features are extracted from the sample, followed by normalization, feature selection, and application of the fault detection model to obtain the equipment health index. This value serves as the quantified failure probability and becomes an input to the simulation model.

Simulation model

The connection between the EPHM module and the simulation model is based on the uncertainty coming from the failure probability which is utilized based on a Monte Carlo approach. Therefore, the simulation of the same production schedule is repeated several times, where failures statistically occur according to the quantified failure probability. In this way, the higher the failure probability, the higher will be the number of repetitions where failures occur.

The simulation model receives this parameter from the EPHM in real-time when the scheduling is triggered, allowing the synchronization of the digital and physical worlds. This parameter throughout the scheduling optimization is not updated during the scheduling activity since the process of alternative schedules comparison should be done on the same initial conditions. In those repetitions where the failure occurs, the processing times in the simulation model are increased by the failure repair time. The simulation model has therefore two input information sources:

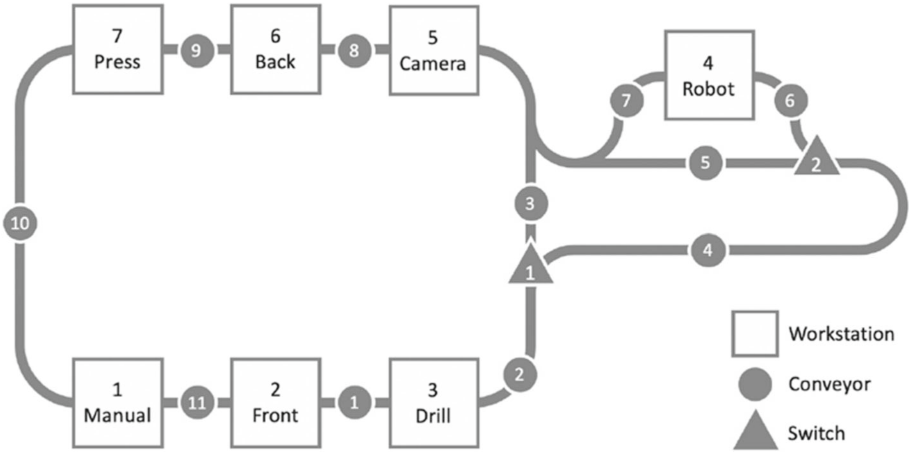

Fig. 4 Active elements of the line at I4.0 Lab

- from the GA comes the single individual to be simulated, i.e. the sequence of jobs. The simulator is fed with one job sequence at a time for the simulation repetitions.
- from the EPHM module comes the failure probability of each equipment, calculated in real-time just before starting the optimization process, as explained in "Equipment PHM module" section.

Overall, once the input data are received, the simulation model runs a DES of the job sequence that reproduces the behavior of the real production system under consideration. After that, the simulation model outputs a set of performance indicators designed in line with the production objectives and the fitness function of the GA.

Case study

The application case used to validate the proposed framework is the assembly line at the Industry 4.0 Laboratory (I4.0 Lab) at the School of Management of Politecnico di Milano. This is represented in Fig. 4 and can be considered as a flowshop (Fumagalli et al. 2016). It is composed of seven workstations: (i) Manual station, where the operator loads the empty carriers at the beginning of the assembly process and unloads the finished pieces at the end; (ii) Front Cover station, where the front cover is positioned; (iii) Drilling station, where holes are drilled; (iv) Robot station, where Printed Circuit Boards (PCB) and fuses are assembled on the cover; (v) Camera Inspection station, that checks their correct positioning; (vi) Back Cover station, which places the back cover; (vii) Press station, where the back and front covers are pressed together and the assembly is completed. The I4.0 Lab is equipped with the Open Platform Communication Unified Architecture (OPC UA), a machine-to-machine communication protocol that is considered one of the connectivity enablers of the Industry 4.0 (https://opcfoundation.org/about/what-is-opc/). It was used in this case to connect the physical assets with the DT.

Preliminary data gathering

The simulation model requires the processing and handling times of each active element of the I4.0 Lab assembly line. The considered active elements are represented in Fig. 4:

- 7 Workstations, which process the workpieces;
- 11 Conveyors, which transport the carriers through the line;
- 2 Switches, which direct each carrier towards the right direction.

A data gathering campaign has been performed on the assembly line by recording the time of each process with a centesimal stopwatch. 20 repetitions of each measure have been performed.

Critical station data gathering

After performing a FMECA on the assembly line, the drilling station was identified as the most critical station of the line. In particular, the critical failure mode was identified as the tool bearing failure, that leads to an imbalance of the drilling axis. For this reason, it became the scope of the failure probability quantification in the EPHM module of the proposed framework. The drilling station has been equipped with additional sensors, such as a set of accelerometers to perceive vibrations along the three axes in order to have visibility on the present state of rotatory elements (Heng et al. 2009). Figure 5 is a picture of the drilling station. During the drilling, the accelerometer data are read by the single-board computer, that sends them to a server available in I4.0 Lab in which a Mongo Database is installed. Using this infrastructure, it is

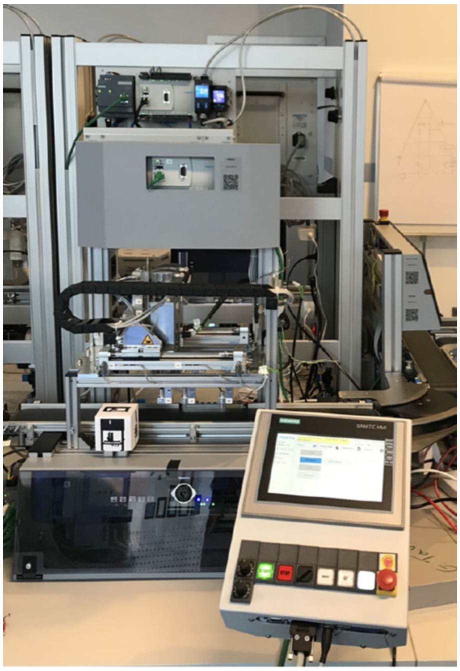

Fig. 5 Drilling station

possible to access previously recorded data as well as real-time data.

For the academic purposes for which the laboratory has been set up, abnormal vibrations were created through a shaker attached to the drilling axis to recreate bad operating conditions of the drilling station. The employed shaker consists of a motor mounted on a PCB which rotates an eccentric mass to produce vibrations along the working axis of the drill and can be adjusted to reduce or amplify its effect.

Proposed genetic algorithm

The proposed GA follows a direct approach, i.e. the encoding of each individual directly describes a sequence of jobs to be performed. The encoding represents each workpiece as a job with the corresponding processes to be performed on the workstations in the right sequence. A single individual is then represented by a sequence of different jobs expressed in the form of a matrix where each row represents a job in the sequence and the columns define the Job code, the sequencing number and all the workstations that each job will encounter.

In this work, a mutation rate equal to 0.02 is adopted and the mutation procedure is performed using a two-point approach, i.e. swapping randomly two rows of the individual to create a new sequence. This is in line with literature, where values used for mutation rate span from 0.01 to 0.3, but the most suggested rates lie between 0.01 and 0.03 (Nguyen and Bagajewicz 2008; Snaselova and Zboril 2015; Uslu et al. 2019; Wu et al. 2018).

Simulation model

According to (Fumagalli et al. 2019), the simulation model has been developed using MATLAB/Simulink and its toolbox for DES, “Simevents” (https://it.mathworks.com/products/simevents.html).

Individual upload

The list of jobs that is input to the simulation model by the GA in form of matrix is divided into individual jobs that are separately considered. Each job is a row of the input matrix from the GA and becomes an entity that flows within the simulated production system. Therefore, the physical workpieces processed by the real workstations correspond to the entities processed by the blocks of the virtual assembly line in the DES model. Information about the operations to be performed in each workstation on the job are uploaded into the simulation model as attributes of the flowing entities that represent the individual jobs.

Production simulation

The behavior of the real production system is virtually reproduced for each job sequence. The model reproduces both the operations performed on the workpiece and the constraints and rules of the I4.0 Lab production system. Indeed, the model developed in DES replicates the behavior of the active elements of the line:

- the 7 Workstations, capable to process one job (single workpiece) at a time; each workstation will then operate on each workpiece for a time that is established by the corresponding attribute of the flowing entity representing the job;
- the 11 Conveyors, that can accommodate a number of carriers on the basis of their physical lengths;
- the 2 Switches, that simulate the carrier’s waiting phase during which the RFID is read to identify the right direction, and the action of addressing the carrier toward the right direction.

In addition to the reproduction of the active elements of the line, different logical constraints are modelled such as the maximum number of simultaneous jobs in line; the maximum number of carriers in the robot queue; the routing decision before the Robot cell.

Performance extraction

The makespan of the whole sequence of jobs is the performance on which the fitness function of the GA will be elaborating. The simulation model is then used to extract

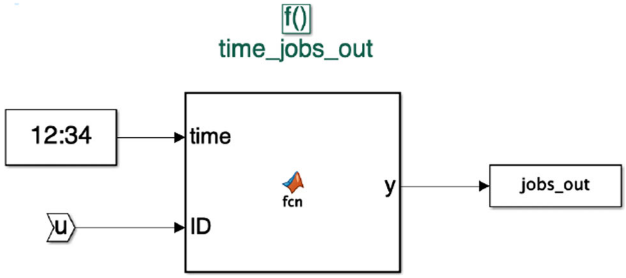

Fig. 6 Function for performance extraction

this performance for each simulated individual. To this end, a Function in the simulation model is responsible for the time tracking and performance extraction, as shown in Fig. 6. In particular:

- the Function records all exit times of all jobs in the sequence;
- the inputs of the Function are: (i) the ID of each job; (ii) its exit time;
- the exit time of the last job from the system is the makespan of the sequence;
- this value is the one fed into the fitness function of the GA.

Modeling equipment failure probability

As described in "Critical station data gathering" section, tool bearing failure in drilling station is selected to be monitored for this study. Vibration signals are collected using the three-axis accelerometer. Additionally, four signals—current, air flow rate, power, and air pressure—are also collected from the PLC using OPC-UA. 100 samples are collected in healthy state of the drilling station and 20 samples are collected with the shaker to simulate a faulty state of the tool. Figure 7 shows the raw signal plots for all seven signals for both healthy and faulty states.

Since the data are collected between the points where the pallet enters and leaves the drilling station, a data segmentation regimen is performed to extract the section where the drilling tool is engaged. On these segmented data, a total of 66 features are extracted. Frequency domain features are used for accelerometer signals and features such as Shannon entropy, kurtosis, skewness etc. are used for time domain signals. Figure 8 shows a sample of extracted features where the first 20 samples represent healthy state and the last 20 samples represent faulty state. Although it is possible to make meaningful inference out of each feature, these features would differ with each failure mode and equipment. Thus, more importantly, it can be observed from the figure that not all features show a discrepancy between the healthy and faulty samples. Thus, selecting useful features is necessary to create an effective model.

Features are normalized with respect to the health samples to mean = 0 and standard deviation = 1. Fisher criteria, given by Eq. (1), is used to calculate the Fisher score for each feature, as shown in Fig. 9. In Eq. (1), μ_A and σ_A define the

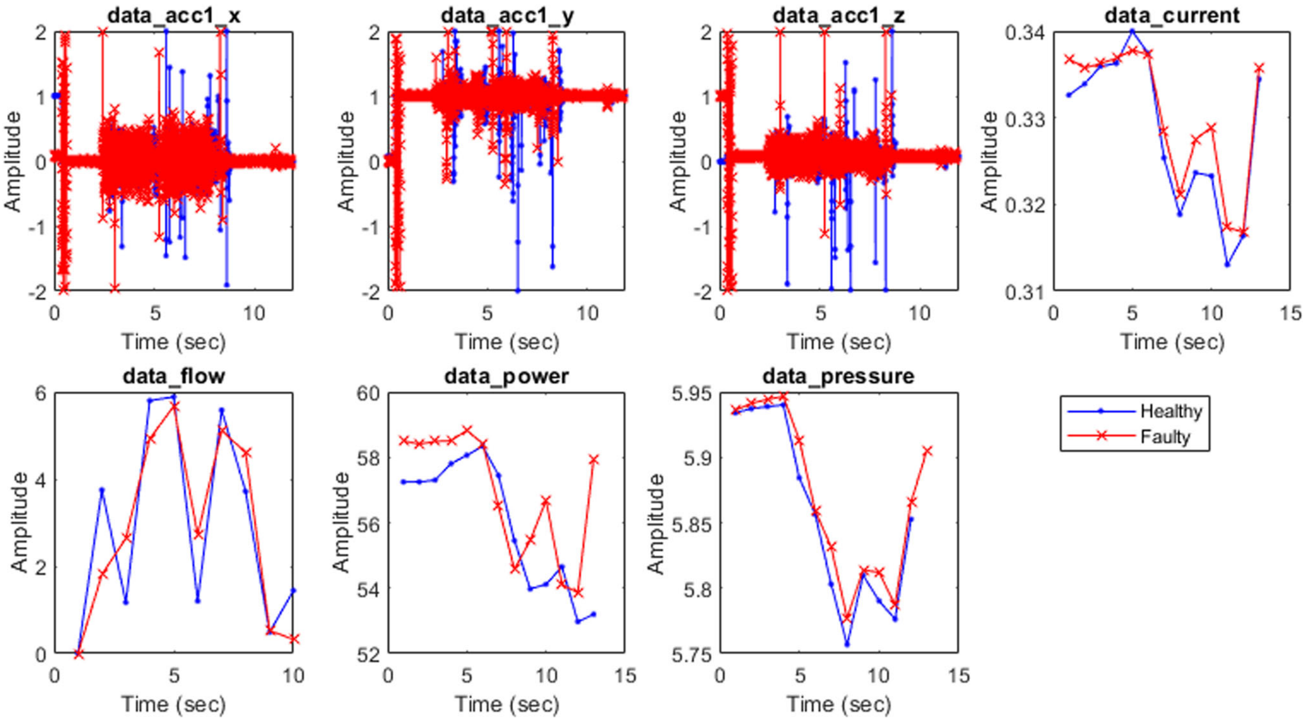

Fig. 7 Visualizing raw data collected from drilling station for modeling EPHM

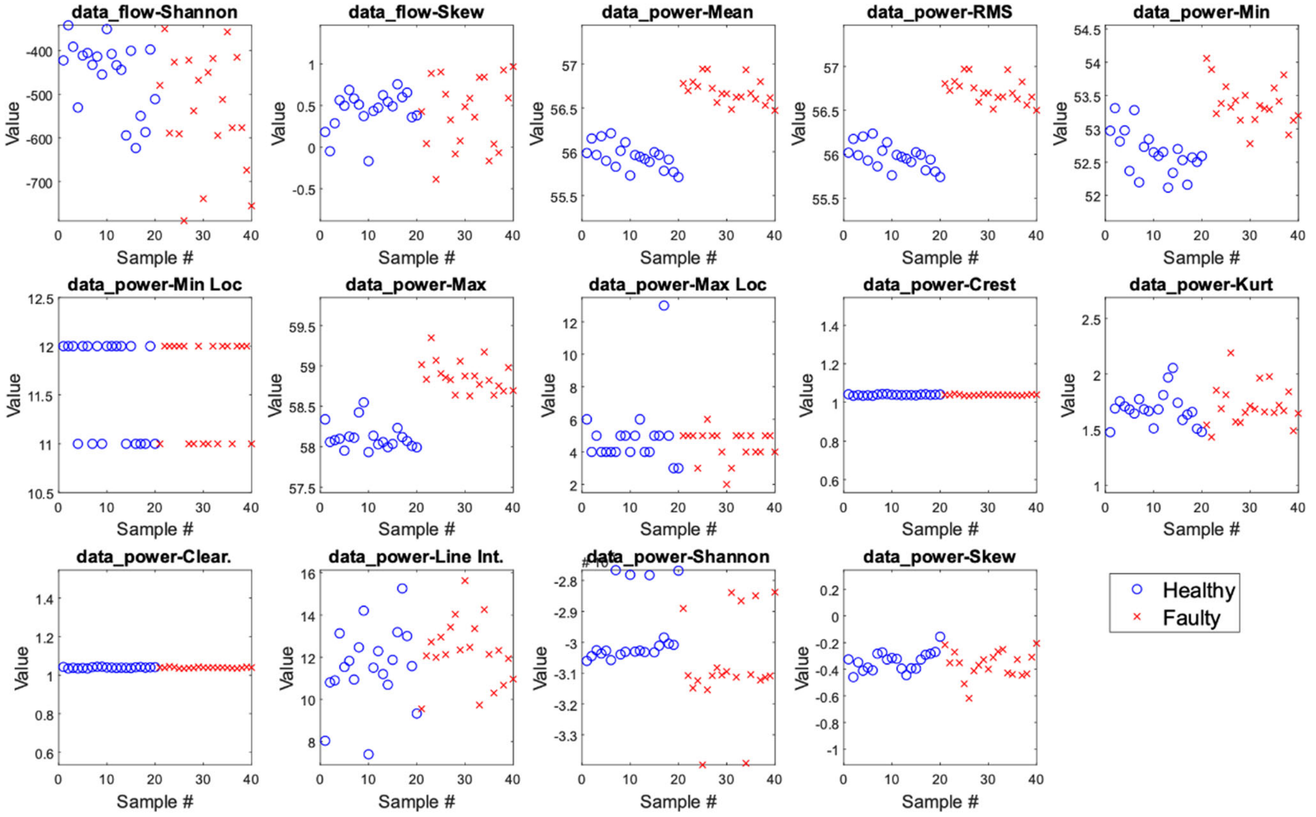

Fig. 8 Visualizing a sample of extracted features for the collected data

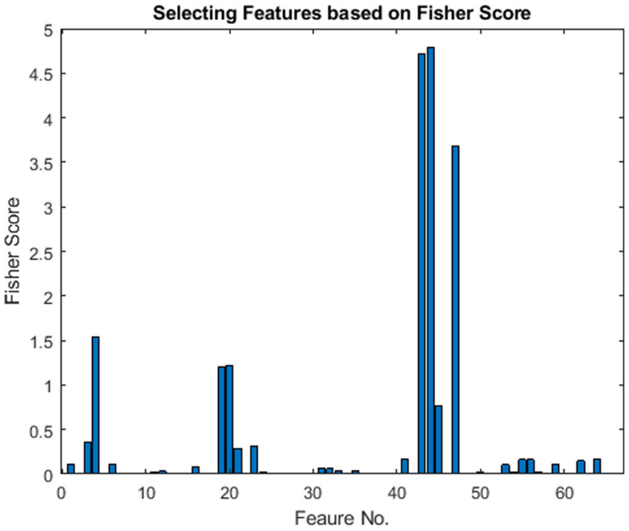

Fig. 9 Selecting features based on Fisher score

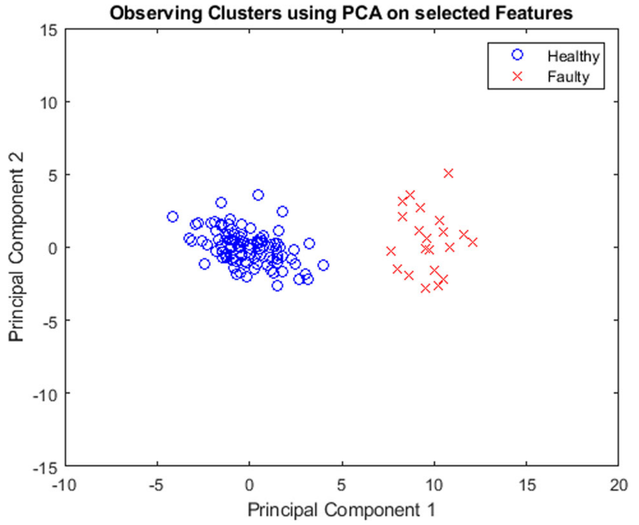

Fig. 10 Principal component 1 versus principal component 2

mean and standard deviation of feature values for healthy samples, and μ_B and σ_B define the mean and standard deviation of feature values for faulty samples. Six features with the highest Fisher scores were selected based on Fig. 9. They are: (1) maximum amplitude in the frequency domain of the vibration signal in the X direction, the (2) mean value and the (3) root mean square (RMS) value of the current signal in the time domain, and (4) mean value, (5) RMS value, and (6) maximum value of the power signal in the time domain. PCA is implemented on the selected six features to observe the presence of potential clusters for healthy and faulty samples. Figure 10 plots the obtained principal component 1 versus

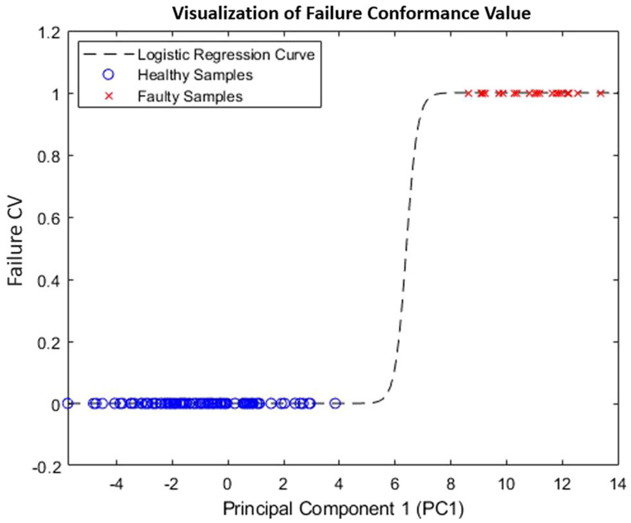

Fig. 11 Failure conformance value modeling

principal component 2 to reveal the clusters, which validates the effectiveness of features extracted and selected.

$$F = \frac{|\mu_A - \mu_B|^2}{\sigma_A^2 - \sigma_B^2} \tag{1}$$

Multivariate logistic regression is further used to model the selected features into a failure conformance value (CV) between 0 and 1, where 0 indicates the absence of a fault and 1 indicates the presence of a fault. Essentially, the failure CV is integrated as the failure probability for the production simulation model. The multivariate logistic regression is given by Eq. (2), where Y_i is the output of the ith sample, $\beta_j, j=0$ to m are coefficients of the model, m is the number of input variables, and $X_{j,i}$ is the jth input variable for the ith sample. In this problem, the value of j is equal to 6, as there are six selected features as input variables for the model. Figure 11 shows the output values of the collected samples based on the trained logistic regression model whose coefficients are obtained using Maximum Likelihood estimation method. First principal component derived from the PCA is used to visualize the approximate fitted curve of the model. The curve represents the behavior of the failure probability as the equipment moves from healthy state to a faulty state. Data could be collected for different severities of fault and the logistic regression model can be trained using labels corresponding to desired health index to further adjust the curve.

$$Y_i = \frac{1}{1 + \exp\left[-\left(\beta_0 + \sum_{j=1}^{m} \beta_j X_{j,i}\right)\right]} + \varepsilon_i \tag{2}$$

Integration of EPHM results into the simulation model

The output of the EPHM module is integrated into the simulation model, to embed it into the scheduling activity. It is going to impact on the processing time of the drilling station. In particular:

- the EPHM module provides a failure probability;
- this probability determines when the drilling station breaks down and, in that case, the standard processing times of the drilling station increase by an amount of time equal to the Mean Time To Repair (MTTR), i.e. the required manual repair intervention time.

Figure 12 shows the function determining the drilling processing times, based on Monte Carlo approach (i.e. using a random_num) to generate the failure occurrence.

Fitness function

The work was developed in two phases, each denoted by a different fitness function.

(1) *Fitness function₁* only minimizes the makespan, given by Eq. (3): it compares the average makespan of the worst individual belonging to the first generation (i.e. first generated population), makespan$_{\text{ref}}$, and the average makespan of the other individuals that are considered one by one by the simulation runs. These values are averaged from the different repetitions of simulations of a single individual, to have statistical robustness of results. The first population is randomly generated, therefore, by taking the worst individual of this population as the reference point, it is reasonable that all subsequent individuals generated will have a lower average makespan, thus maintaining the fitness value always positive. The objective of the following generations is to increase this difference.

$$fitness\ function_1 = makespan_{ref} - makespan \tag{3}$$

(2) *Fitness function₂* leads to a balance between the average makespan and its standard deviation; thus, it is composed of two fitness functions with weighing factors λ and (1 − λ), as in Eq. (4). The first fitness function in Eq. (4) is relative to the makespan (same structure as Eq. (3)); the second is relative to the standard deviation of the makespan, introduced to measure the robustness of the solutions (the negative sign denotes that the fitness function rewards individuals that have a lower standard deviation). The best solutions will be those that have a good balance between the average makespan and

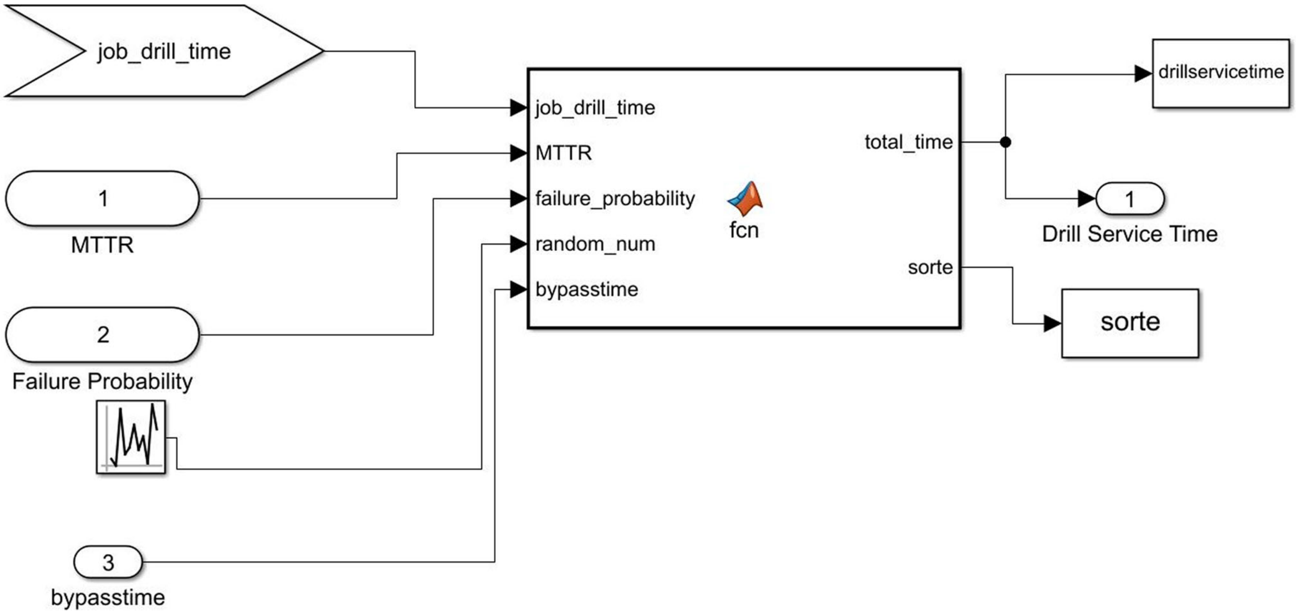

Fig. 12 Drilling station processing time function

its standard deviation. By changing λ, it is possible to modify the behavior of the algorithm to favor solutions with lower average makespan or better robustness of the scheduling solutions.

$$fitness\ function_2 = \lambda * \left(makespan_{ref} - makespan\right) + (1 - \lambda) * (-stddev) \tag{4}$$

Results and discussion

Due to the uncertainty about failures in the drilling station, its processing times become stochastic. According to the Monte Carlo approach, simulation of each individual is repeated several times to evaluate the robustness of the results. The number of repetitions was set to 30. In all repetitions, the input sequence of jobs was the same to allow comparability of results. Increasing the number of repetitions protracts the time required to simulate them, even though the computational time is highly dependent on the performances of the single computer used. Since the proposed application is for real-time scheduling, it is of utmost importance to have a quick response of the scheduling framework. The proposed application used parallel computing to launch parallel simulations, to reduce the time of the simulation repetitions within minutes. This is possible because the sequence simulations are independent.

This section reports and discusses the results of a series of experiments that show the viability of the proposed framework integrating scheduling, DT and PHM. Table 2 shows the parameters of the GA and of the simulation for the experiments done.

The results of experiments are correspondingly presented in the next sections, that is: experiments with fitness function$_1$ in "Results with fitness function1" section; experiments with fitness function$_2$ in "Results with fitness function2" section; the three additional tests to analyze the effect of failures, using fitness function$_2$, in "Additional tests to analyze the effect of failures" section; the experiments with 100 iterations for the GA, using fitness function$_2$, in "Results with one hundred iterations for the genetic algorithm" section. Finally, "Human machine interface" section introduces the Human Machine Interface considered in the framework to ease a decision-maker when choosing the best sequence, while "Discussion" section discusses the general features of the obtained results.

Results with fitness function$_1$

Figure 13 shows the results of the experiments with the fitness function$_1$. In particular, it visualizes (i) the evolution of the average makespan of the individuals of a generation (i.e. green colored circle shown in the figure, indicating the average value computed across the individuals evaluated at each generation) and (ii) the evolution of the average makespan and standard deviation of the worst and best individuals (i.e., blue and red colored at each generation, indicating the average makespan—blue and red colored circle—inserted in the

Table 2 Parameters of the experiments

Parameter	Experiments with fitness function1	Experiments with fitness function2	Additional Test 1	Additional Test 2	Additional Test 3	Experiments with 100 iterations
Number of jobs to be scheduled	50	50	50	50	50	50
Population dimension	10	10	10	10	10	10
Mutation rate	0.02	0.02	0.02	0.02	0.02	0.02
Number of children for elitism operator	1	1	1	1	1	1
Maximum number of iterations	100	100	100	100	100	100
Minimum number of iterations	10	10	10	10	10	100
Number of iterations for stall criterion	20	10	10	10	10	10
Number of repetitions of each simulation	30	30	30	30	30	30
Probability of failure from EPHM	0.149	0.149	0.564	0.149	0.564	0.149
MTTR [s]	60	60	120	120	60	300
Weighing factors λ	–	0.5	0.5	0.5	0.5	0.5

symmetric interval due to the standard deviation of the individual repetitions run in the simulation; the interval is as wide as twice the standard deviation). In this way it is possible to have an idea of both average makespan and robustness of the generated schedules.

It must be remembered that the use of a GA randomly impacts the convergence of results reported in Figs. 13, 14, and 15. As a result, according to fitness function$_1$, the optimization considers an individual better than another if it has a lower average makespan, no matter how robust the solution is. Indeed, in Fig. 13 it is possible to see that the optimization direction prefers individuals with slightly lower makespan, but with much higher standard deviation (best individuals), than more robust individuals (worst individuals). It is also worth observing that the standard deviation of the individuals significantly varies along the GA iterations, and no trend can be recognized along the generations.

On the whole, the need to also consider the standard deviation in the solution search process appears evident, inte-

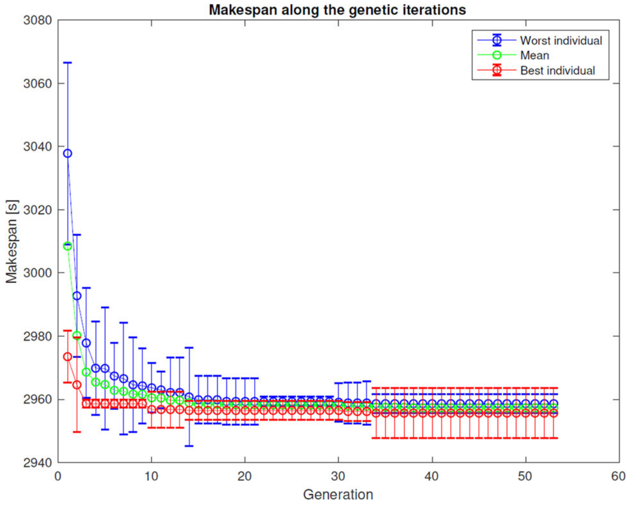

Fig. 13 Evolution of makespan with fitness function$_1$

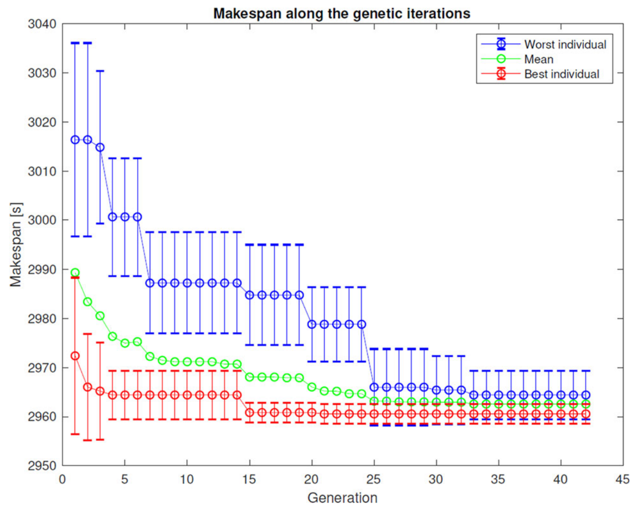

Fig. 14 Evolution of makespan with fitness function$_2$

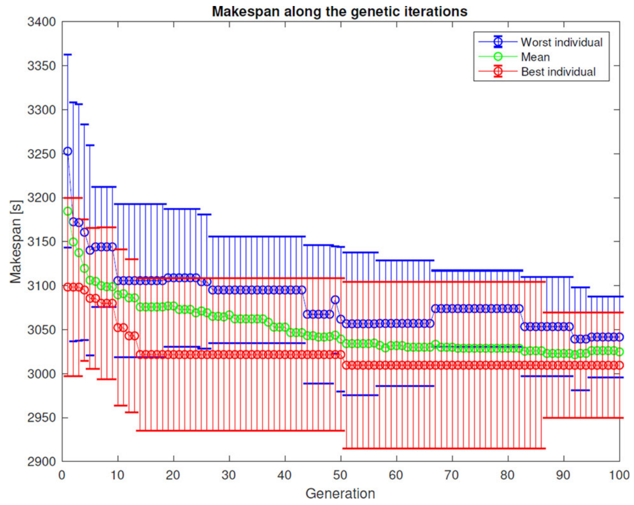

Fig. 15 Evolution of makespan in 100 iterations with fitness function$_2$

grating its evaluation within the GA fitness function in order to ensure a good balance between a good makespan and robustness of the solution.

Results with fitness function$_2$

Fitness function$_2$ optimizes both the average makespan and the standard deviation associated to the repetitions of the simulation of each individual simultaneously. λ is set equal to 0.5 to give the same importance to the two fitness functions—makespan and robustness—so to obtain solutions that behave well under both aspects. Other tests have been performed with different values of λ but are not reported here because they are not meaningful for the purposes of the paper.

As shown in Fig. 14, the generations of the GA lead to the definition of increasingly more efficient solutions with respect to both the performances (best individuals). Accordingly, the comparison between Figs. 13 and 14 shows the algorithm converges to an average makespan slightly higher with fitness function$_2$ than with fitness function$_1$. The solutions that strongly favor the makespan minimization are discarded by the GA in favor of robust solutions with more balanced performances, thus giving importance also to a reduced standard deviation. This can better suit the practicality of scheduling in industry.

Additional tests to analyze the effect of failures

To the reader a clear vision of the scheduling framework behavior, three additional tests were performed using the fitness function$_2$, with different MTTR values and different sets of vibrations (i.e. leading to different failure probabilities): (i) test 1 raises both the MTTR and the failure probability; (ii) test 2 keeps the initial failure probability as in the "Results with fitness function2" section, but increases the MTTR; (iii) test 3 keeps the initial MTTR, but increases the failure probability (see Table 2 for the value of these parameters).

Aggregated results are provided in Tables 3 and 4. They respectively show the improvement of makespan and standard deviation obtained by the individual with the highest fitness compared to the worst individual and the best individual of the first generation. It is possible to get an overview of how advantageous the proposed simheuristics framework is, compared to the best and the worst schedules from to a random population (like the first generation). The results are logically justifiable: the best sequence is improving both average makespan and standard deviation with respect to the best and worst individuals of the first generation. In addition, the results clearly show that the percentage of improvement of the standard deviation are one order of magnitude higher than the percentage of improvements in the average makespan. This means that, with the fitness function$_2$, the best solutions show less dispersed performances against the uncertainty due to failures. This is valid for all combinations of the MTTR and failure probabilities.

It is possible to note that the makespan of the found solution does not offer a high percentage decrease with respect to the best individual of the first generation (Table 4). A reason for this is rooted in the fact that the testing facility has strict constraints. Moreover, in both the comparisons (Tables 3 and

Table 3 Improvement of the proposed sequence compared to the worst individual of the first generation

	MTTR = 60 s	MTTR = 120 s
Failure probability = 0.149	Makespan = – 18.60% Std dev = – 95.20%	Makespan = – 4.00% Std dev = – 87.00%
Failure probability = 0.564	Makespan = – 2.80% Std dev = – 94.00%	Makespan = – 10.10% Std dev = – 31.40%

Table 4 Improvement of the proposed sequence compared to the best individual of the first generation

	MTTR = 60 s	MTTR = 120 s
Failure probability = 0.149	Makespan = − 4.00% Std dev = − 93.80%	Makespan = − 0.30% Std dev = − 16.30%
Failure probability = 0.564	Makespan = − 0.40% Std dev = − 63.00%	Makespan = − 1.60% Std dev = − 11.60%

4), considering as reference the initial values of low MTTR and low failure probability, the benefit obtainable in terms of makespan decreases as the failure probability or the MTTR increases. This is because the increase of one of the two parameters leads to the addition of higher processing times, which means a higher total makespan. Instead, the robustness of the solution improves greatly, when the MTTR or the failure probability decrease. This is explainable by the fact that, increasing either the MTTR or the failure probability, it is more difficult to find a robust solution that can absorb the delays due to failures. Overall, the highly different behaviors of makespan and standard deviation confirm the need to evaluate both in the fitness function.

Results with one hundred iterations for the genetic algorithm

A final test has been carried out, with the peculiarity of being forced to reach one hundred generations. A set of vibrations, producing a lower failure probability, has been used together with a higher MTTR with respect to the initial tests (see Table 2 for the value of these parameters).

Similarly to Figs. 13, 14, and 15 shows the evolution of the average makespan of the worst and best individuals within the interval of twice their relative standard deviation. Sometimes, the blue colored line (worst individuals) goes to a higher makespan value than the previous generation, but with a reduction of standard deviation. In general, both blue and red colored lines (the worst and the best individuals) are showing similar trends of convergence, always considering a balanced reduction of both makespan and standard deviation of the individuals. It confirms previous results in terms of evolution of the optimization algorithm.

Human machine interface

The visualization in Figs. 13, 14, and 15 is not intuitive when choosing the best sequence. Each generated schedule has characteristics that make it different from the others. However, all of them are the results of an optimization search aimed at balancing both the performances related to makespan and standard deviation. To simplify the choice, the proposed framework implementation considers the visualization of three best alternatives, each aimed at optimizing a different characteristic. Figure 16 correspondingly shows a set of three sequences output by the framework to the human decision-maker, as a result of the last generated population:

- the best fitness value sequence, corresponding to the best trade-off between good and robust makespan, according to the input weights, based on Eq. (4);
- the lower average makespan sequence, minimizing the makespan regardless of the robustness of the solution;
- the lower standard deviation sequence, corresponding to the most robust solution, with the least fluctuation of makespan; it may present a higher average makespan.

In Fig. 16, for each of the three individuals (three sequences) the typical boxplot is displayed: the blue circle indicates the average makespan and the red line indicates the median. The rectangle includes values within the 25° and the 75° percentile and the whiskers reach the extreme values (the isolated red cross indicates an outlier).

As Fig. 16 is drawn considering the experiments reported in “Results with one hundred iterations for the genetic algorithm” section, it is worth remarking that the individual with the best fitness corresponds to the best individual for fitness function_2 shown in Fig. 15 in the last generation; the other two best individuals—best makespan and best standard deviation—provides additional choices for the decision-maker.

Generalizing, this human–machine interface is also part of the proposed framework. Herein, the display of the best individuals is shown, to let the human decision-makers decide which is the most appropriate schedule to apply according to his/her expertise and contingent production needs. This reflects a precise choice of the authors that preferred not to fully automatize the re-scheduling driven by the proposed DT-based framework. This leaves a degree of flexibility to the human decision-makers that is reasonable in practice.

Discussion

The present article proposes the use of a DT-based framework for real-time scheduling in a flowshop environment. The converging results presented from “Results with fitness function1” to “Results with one hundred iterations for the genetic algorithm” sections demonstrate the viability of the framework that integrates the DT module—embedding a DES model and the real time data gathering and analysis into the EPHM module—and the GA-based optimization module. The applicability of the framework to flowshops is demonstrated by the application case in the I4.0 Lab; however, by changing the simulation model and the GA encoding it is possible to adapt the same framework also to other types of production systems.

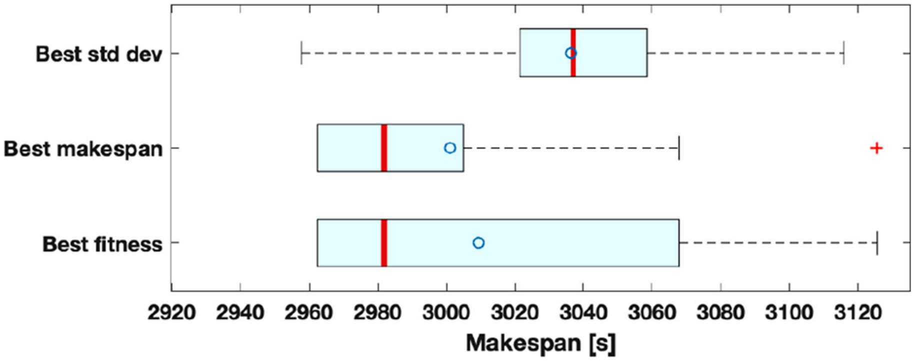

Fig. 16 Human-machine interface

Overall, the outcome of this article can be discussed with concern to the Digital Twin development in manufacturing and to the robust scheduling.

Looking at the Digital Twin development in manufacturing, the proposed synchronized simulation can be inserted in the classification by (Kritzinger et al. 2018) as a Digital Shadow, according to the capabilities shown by the experiments and implementation presented in this article: the proposed framework does not feedback into the field of decision-making autonomously; the operator is instead asked to manually choose the preferred schedule among the best ones proposed by the human–machine interface. However, the step forward to build a full DT is conceptually short: the implementation of an automated passage here is not conceptually different, as the optimization module may be already provided with the rules to choose the preferred schedules autonomously, and the decision may be sent to the control software module directly as a result of the schedule optimization.

Regarding the robust scheduling, as outcome of interest to industrial practice this work fosters a decision-making support with a preventive approach.

- According to the proposed framework, the use of real-time data from accelerator sensors allows to compute the failure probability of the equipment and to use it in the simulation model, which is used to compute the performances of the single production schedule. Besides, considering the experimental evidence, the results show that, according to the number of repetitions and according to the magnitude of MTTR, the standard deviation of the performances may highly change. On the whole, for an effective decision-making support it is therefore advisable to define the criteria for the selection of the best schedule: only considering the makespan is a too narrow perspective, the standard deviation should be considered, finally leaving the decision to the human decision-maker depending on the contingencies.
- Concerning the time constraints to actionable information, it must be considered that, although the proposed framework fosters a real-time scheduling approach, it is not purely reactive; conversely, the scheduling solutions found need to be robust against uncertainty in a preventive approach. This should be considered to justify the computational time of the proposed framework, which is not in the order of seconds but may take a few minutes to run. It is indeed true that the framework relies on the DES, which requires a longer computational time than the use of programming languages, such as C-based languages. Despite this limit, the authors see in the use of DES the possibility to accurately simulate - and therefore to elaborate production schedules—of complex real manufacturing systems. For this reason, the additional computational time required by the DES is justified. This is particularly valid when considering that scheduling is timely triggered to run the preventive approach to robust scheduling, which means feeding the real-time data to the data analytics and elaborating the simheuristic in a time window of a few minutes before achieving the preventive decision.

Conclusions and future works

The production scheduling remains an open challenge for research and industry. The present article brings forward an innovative framework for robust scheduling that embeds traditional instruments such as simulation and metaheuristics (GA) with Industry 4.0 data-driven capabilities. Through the synchronization of sensors data of the real system, it is possible to consider the simulation and the simheuristics implemented according to the DT paradigm.

The application in the I4.0 Lab demonstrates the integration of various knowledge backgrounds from the different research areas—robust production scheduling, Digital Twin development, and PHM—into a unique framework as an initial proof-of-concept. The results reach the declared objectives of the paper: they show that it is possible to offer the inclusion of uncertainty in the scheduling activity in a flowshop context, that is still mostly treated in a deterministic way in literature. In addition, the uncertainty is not modelled according to historical data and to similar systems' behavior, but the uncertainty reflects the real-life randomness of disruptions. As demonstrated in the application of the framework, the failure probability of equipment is computed in real time, according to the operating conditions from sensor data read from the field and elaborated through the EPHM in the Digital Twin module. The results of the scheduling then offer the makespan and the standard deviation of the best and worst individuals of each generation, evaluating the robustness of the solution in the evolution of the optimization algorithm.

Overall, the present work contributes to the research on DT in manufacturing, proposing an innovative use of Digital Shadow/Twin (i.e. synchronized simulation) for the production scheduling with consideration of uncertainty. In this way, a data-driven statistical model is adopted to compute the failure probability, implemented through the EPHM module of the Digital Shadow/Twin. The application of the proposed framework is demonstrated in a flowshop in an Industry 4.0-compliant physical laboratory, in order to have a real system in which the schedule could be applied and to have the real time data gathering from sensors. The approach proposes to avoid the full automatization of the best schedule selection; instead, it leaves the choice to the operator to select the best preferred option, according to the results of the scheduling framework, his/her experience and contingent operating conditions. The proposed framework also contributes to the robust scheduling research stream, by substituting the "traditional" statistical distributions to describe the stochastic behavior, with data-driven statistical models that are obtained both from historical data and real-time data from the sensors and are integrated into the DT paradigm.

The paper finally opens new opportunities for research in the DT area. In particular, high potentials reside in the data-driven analyses that can be made in the EPHM module. The framework proposed and demonstrated a statistical description of the failure behavior of the drilling station, while a future direction of investigation can be to model the behavior prediction soon. Thus, a more complete EPHM module can be developed that not only detects a failure (and a corresponding failure probability) but can also predict the future trend of the failure probability. It will allow a comparison between different failure conditions and their effect on dynamic production scheduling. Further, the proposed framework is considering a centralized decision-making approach, based on a centralized DT simulation of the whole production system. Another future direction of investigation may lead to study the impact on decisions when the DT is distributed and associated to the local control of the single workstations. By constructing an agent-based architecture, new possibilities of decision-making would be opened for further research to this end. Lastly, it is worth remarking that data communication protocols, such as OPC-UA, provide limited sampling rate which may not be suitable for detecting faults in their incipient stages. To cope with this issue, additional sensors, and computational architectures such as Edge computing can be integrated into the framework.

Acknowledgements This work was supported by the Ministry of Education, University and Research (Italy), in relation to the project "Smart Manufacturing 2020" (Grant Number CTN01_00163_216744). It has been possible thanks to the use of the Industry 4.0 Laboratory of the Manufacturing Group at the School of Management of the Politecnico di Milano (lab site: www.industry40lab.org) and of the NSF Industry/University Cooperative Research Center on Intelligent Maintenance Systems (IMS) at the University of Cincinnati. The authors express their gratitude to Lorenzo Cesarini for his support in the presented work.

Funding Open access funding provided by Politecnico di Milano within the CRUI-CARE Agreement. This work was supported by the Ministry of Education, University and Research (Italy), in relation to the project "Smart 1136 Manufacturing 2020" (Grant Number CTN01_00163_216744).

References

Abedinnia, H., Glock, C. H., Grosse, E. H., & Schneider, M. (2017). Machine scheduling problems in production: A tertiary study. *Computers & Industrial Engineering, 111,* 403–416. https://doi.org/10.1016/j.cie.2017.06.026.

Allahverdi, A., Aydilek, H., & Aydilek, A. (2018). No-wait flowshop scheduling problem with two criteria; total tardiness and makespan. *European Journal of Operational Research, 269*(2), 590–601. https://doi.org/10.1016/j.ejor.2017.11.070.

Aramon Bajestani, M., & Beck, J. C. (2015). A two-stage coupled algorithm for an integrated maintenance planning and flowshop scheduling problem with deteriorating machines. *Journal of Scheduling, 18*(5), 471–486. https://doi.org/10.1007/s10951-015-0416-2.

Bagheri, A., Zandieh, M., Mahdavi, I., & Yazdani, M. (2010). An artificial immune algorithm for the flexible job-shop scheduling problem. *Future Generation Computer Systems, 26*(4), 533–541. https://doi.org/10.1016/j.future.2009.10.004.

Baheti, R., & Gill, H. (2011). Cyber-physical Systems. In T. Samad & A. Annaswamy (Eds.), *The Impact of Control Technology* (Issue 12, pp. 161–166). IEEE Control Systems Society.

Behnamian, J., & Zandieh, M. (2011). A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties. *Expert Systems with Applications, 38*(12), 14490–14498. https://doi.org/10.1016/j.eswa.2011.04.241.

Borangiu, T., Oltean, E., Raileanu, S., Anton, F., Anton, S., & Iacob, I. (2020). Embedded Digital Twin for ARTI-type control of semi-continuous production processes. *International Workshop on Service Orientation in Holonic and Multi-Agent Manufacturing, 2019,* 113–133.

Castelli, M., Cattaneo, G., Manzoni, L., & Vanneschi, L. (2019). A distance between populations for n-points crossover in genetic algorithms. *Swarm and Evolutionary Computation, 44,* 636–645. https://doi.org/10.1016/j.swevo.2018.08.007.

Cimino, C., Negri, E., & Fumagalli, L. (2019). Review of Digital Twin applications in manufacturing. *Computers in Industry, 113,* 103130. https://doi.org/10.1016/j.compind.2019.103130.

Della Croce, F., Tadei, R., & Volta, G. (1995). A genetic algorithm for the job shop problem. *Computers & Operations Research, 22*(1), 15–24. https://doi.org/10.1016/0305-0548(93)E0015-L.

Dias, L. S., & Ierapetritou, M. G. (2016). Integration of scheduling and control under uncertainties: Review and challenges. *Chemical Engineering Research and Design, 116,* 98–113. https://doi.org/10.1016/j.cherd.2016.10.047.

Eddaly, M., Jarboui, B., & Siarry, P. (2016). Combinatorial particle swarm optimization for solving blocking flowshop scheduling problem. *Journal of Computational Design and Engineering, 3*(4), 295–311. https://doi.org/10.1016/j.jcde.2016.05.001.

Enders, M. R., & Hoßbach, N. (2019). Dimensions of Digital Twin applications—A literature review. In *25th Americas conference on information systems, AMCIS 2019*, Code 151731.

Falkenauer, E., & Bouffouix, S. (1991). A genetic algorithm for job shop. In *Proceedings of 1991 IEEE International Conference on Robotics and Automation*, pp. 824–829. https://doi.org/10.1109/ROBOT.1991.131689.

Framinan, J. M., Fernandez-Viagas, V., & Perez-Gonzalez, P. (2019). Using real-time information to reschedule jobs in a flowshop with variable processing times. *Computers & Industrial Engineering, 129,* 113–125. https://doi.org/10.1016/j.cie.2019.01.036.

Framinan, J. M., Leisten, R., & García, R. R. (2014). *Manufacturing scheduling systems: An integrated view on models, methods and tools*. London: Springer. https://doi.org/10.1007/978-1-4471-6272-8.

Fumagalli, L., Macchi, M., Negri, E., Polenghi, A., & Sottoriva, E. (2017). Simulation-supported framework for job shop scheduling with genetic algorithm. In *Proceedings of the XXII summerschool of industrial mechanical plants "Francesco Turco", Palermo (Italy), 13–15th September 2017* (pp. 1–8).

Fumagalli, L., Macchi, M., Pozzetti, A., Tavola, G., & Terzi, S. (2016). New methodology for smart manufacturing research and education : The lab approach. In *21st Summer School Francesco Turco 2016* (pp. 42–47).

Fumagalli, L., Negri, E., Sottoriva, E., Polenghi, A., & Macchi, M. (2018). A novel scheduling framework: Integrating genetic algorithms and discrete event simulation. *International Journal Management and Decision Making, 17*(4), 371–395.

Fumagalli, L., Polenghi, A., Negri, E., & Roda, I. (2019). Framework for simulation software selection. *Journal of Simulation*, *13*(4), 286–303. https://doi.org/10.1080/17477778.2019.1598782.

Garey, M. R., & Jonhson, D. S. (1979). *Computers and intractability: A guide to the results of the study. Within the range of the used scenarios, theory of NP-completeness*. San Francisco: Freeman.

Gonzalez-Neira, E. M., Ferone, D., Hatami, S., & Juan, A. A. (2017). A biased-randomized simheuristic for the distributed assembly permutation flowshop problem with stochastic processing times. *Simulation Modelling Practice and Theory, 79,* 23–36. https://doi.org/10.1016/j.simpat.2017.09.001.

Gupta, J. N. D., & Stafford, E. F. (2006). Flowshop scheduling research after five decades. *European Journal of Operational Research, 169*(3), 699–711. https://doi.org/10.1016/j.ejor.2005.02.001.

Hatami, S., Calvet, L., Fernández-Viagas, V., Framiñán, J. M., & Juan, A. A. (2018). A simheuristic algorithm to set up starting times in the stochastic parallel flowshop problem. *Simulation Modelling Practice and Theory, 86*(April), 55–71. https://doi.org/10.1016/j.simpat.2018.04.005.

Heng, A., Zhang, S., Tan, A. C. C., & Mathew, J. (2009). Rotating machinery prognostics: State of the art, challenges and opportunities. *Mechanical Systems and Signal Processing, 23*(3), 724–739. https://doi.org/10.1016/j.ymssp.2008.06.009.

Herroelen, W., & Leus, R. (2005). Project scheduling under uncertainty: Survey and research potentials. *European Journal of Operational Research, 165*(2), 289–306. https://doi.org/10.1016/j.ejor.2004.04.002.

Johnson, S. M. (1954). Optimal two and three stage production schedules with set-up time included. *Naval Research Logistics Quarterly,1*(1), 61–68. https://doi.org/10.1002/nav.3800010110

Juan, A. A., Barrios, B. B., Vallada, E., Riera, D., & Jorba, J. (2014). A simheuristic algorithm for solving the permutation flow shop problem with stochastic processing times. *Simulation Modelling Practice and Theory, 46,* 101–117. https://doi.org/10.1016/j.simpat.2014.02.005.

Juan, A. A., Faulin, J., Grasman, S. E., Rabe, M., & Figueira, G. (2015). A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives, 2,* 62–72. https://doi.org/10.1016/j.orp.2015.03.001.

Kritzinger, W., Karner, M., Traar, G., Henjes, J., Sihn, W., Kritzinger, W., et al. (2018). Digital Twin in manufacturing: A categorical literature review and classification. *IFAC-PapersOnLine, 51*(11), 1016–1022. https://doi.org/10.1016/j.ifacol.2018.08.474.

Krug, W., Wiedemann, T., Liebelt, J., & Baumbach, B. (2002). Simulation and optimization in manufacturing, organization and logistics. In *Simulation in industry, 14th European simulation symposium* (pp. 423–429).

Lee, E. A. (2008). Cyber physical systems: Design challenges. In *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)* (pp. 363–369). https://doi.org/10.1109/ISORC.2008.25.

Lee, C. K. H. (2018). A review of applications of genetic algorithms in operations management. *Engineering Applications of Artificial Intelligence, 76,* 1–12. https://doi.org/10.1016/j.engappai.2018.08.011.

Lee, J., Ardakani, H. D., Yang, S., & Bagheri, B. (2015a). Industrial big data analytics and cyber-physical systems for future maintenance and service innovation. *Procedia CIRP, 38,* 3–7. https://doi.org/10.1016/j.procir.2015.08.026.

Lee, J., Bagheri, B., & Kao, H. A. (2015b). A cyber-physical systems architecture for Industry 4.0-based manufacturing systems. *Manufacturing Letters, 3,* 18–23. https://doi.org/10.1016/j.mfglet.2014.12.001.

Lee, J., Lapira, E., Bagheri, B., & Kao, H. A. (2013). Recent advances and trends in predictive manufacturing systems in big data environment. *Manufacturing Letters, 1*(1), 38–41. https://doi.org/10.1016/j.mfglet.2013.09.005.

Leitão, P., Karnouskos, S., Ribeiro, L., Lee, J., Strasser, T., & Colombo, A. W. (2016). Smart agents in industrial cyber-physical systems. *Proceedings of the IEEE, 104*(5), 1086–1101. https://doi.org/10.1109/JPROC.2016.2521931.

Lim, K. Y. H., Zheng, P., & Chen, C. H. (2020). A state-of-the-art survey of Digital Twin: Techniques, engineering product lifecycle management and business innovation perspectives. *Journal of Intelligent Manufacturing, 31*(6), 1313–1337. https://doi.org/10.1007/s10845-019-01512-w.

Liu, C. H. (2016). Mathematical programming formulations for single-machine scheduling problems while considering renewable energy uncertainty. *International Journal of Production Research, 54*(4), 1122–1133. https://doi.org/10.1080/00207543.2015.1048380.

Lolli, F., Balugani, E., Gamberini, R., & Rimini, B. (2017). Stochastic assembly line balancing with learning effects. *IFAC-PapersOnLine, 50*(1), 5706–5711. https://doi.org/10.1016/j.ifacol.2017.08.1122.

Lopes, M. R., Costigliola, A., Pinto, R., Vieira, S., & Joao, M. C. (2019). Pharmaceutical quality control laboratory Digital Twin—A novel governance model for resource planning and scheduling. *International Journal of Production Research, Article in press.*. https://doi.org/10.1080/00207543.2019.1683250.

Macchi, M., Roda, I., Negri, E., & Fumagalli, L. (2018). Exploring the role of Digital Twin for asset lifecycle management. *IFAC-PapersOnLine, 51*(11), 790–795. https://doi.org/10.1016/j.ifacol.2018.08.415.

Marmolejo-Saucedo, J. A., Hurtado-Hernandez, M., & Suarez-Valdes, R. (2019). Digital Twins in supply chain management: A brief literature review. In *International conference on intelligent computing and optimization* (pp. 653–661).

Napoleone, A., Macchi, M., & Pozzetti, A. (2020). A review on the characteristics of cyber-physical systems for the future smart factories. *Journal of Manufacturing Systems, 54,* 305–335. https://doi.org/10.1016/j.jmsy.2020.01.007.

Negri, Elisa, Berardi, S., Fumagalli, L., & Macchi, M. (2020). MES-integrated Digital Twin frameworks. *Journal of Manufacturing Systems, 56,* 58–71. https://doi.org/10.1016/j.jmsy.2020.05.007.

Negri, E., Davari Ardakani, H., Cattaneo, L., Singh, J., Macchi, M., & Lee, J. (2019a). A Digital Twin-based scheduling framework including equipment health index and genetic algorithms. *IFAC-PapersOnLine, 52*(10), 43–48. https://doi.org/10.1016/j.ifacol.2019.10.024.

Negri, E., Fumagalli, L., Cimino, C., & Macchi, M. (2019b). FMU-supported simulation for CPS digital twin. *Procedia manufacturing*, *28*, 201–206. https://doi.org/10.1016/j.promfg.2018.12.033.

Negri, E., Fumagalli, L., & Macchi, M. (2017). A review of the roles of Digital Twin in CPS-based production systems. *Procedia Manufacturing, 11,* 939–948. https://doi.org/10.1016/j.promfg.2017.07.198

Neufeld, J. S., Gupta, J. N. D., & Buscher, U. (2016). A comprehensive review of flowshop group scheduling literature. *Computers & Operations Research, 70,* 56–74. https://doi.org/10.1016/j.cor.2015.12.006.

Nguyen, D., & Bagajewicz, M. (2008). Optimization of preventive maintenance scheduling in processing plants. *Computer Aided Chemical Engineering, 25,* 319–324. https://doi.org/10.1016/S1570-7946(08)80058-2.

Orozco-Romero, A., Arias-Portela, C. Y., & Marmolejo- Saucedo, J. A. (2019). The use of agent-based models boosted by Digital Twins in the supply chain: A literature review. In *International conference on intelligent computing and optimization* (pp. 642–652).

Oztemel, E., & Gursev, S. (2020). Literature review of Industry 4.0 and related technologies. *Journal of Intelligent Manufacturing, 31*(1), 127–182. https://doi.org/10.1007/s10845-018-1433-8.

Pan, Q. K., & Wang, L. (2012). Effective heuristics for the blocking flowshop scheduling problem with makespan minimization. *Omega, 40*(2), 218–229. https://doi.org/10.1016/j.omega.2011.06.002.

Pessoa, L. S., & Andrade, C. E. (2018). Heuristics for a flowshop scheduling problem with stepwise job objective function. *European Journal of Operational Research, 266*(3), 950–962. https://doi.org/10.1016/j.ejor.2017.10.045.

Rossit, D., & Tohmé, F. (2018). Scheduling research contributions to smart manufacturing. *Manufacturing Letters, 15,* 111–114. https://doi.org/10.1016/j.mfglet.2017.12.005.

Shi, J., Wan, J., Yan, H., & Suo, H. (2011). A survey of cyber-physical systems. In *2011 international conference on wireless communications and signal processing, WCSP 2011* (pp. 1–6). https://doi.org/10.1109/WCSP.2011.6096958.

Snaselova, P., & Zboril, F. (2015). Genetic algorithm using theory of chaos. *Procedia Computer Science, 51*(1), 316–325. https://doi.org/10.1016/j.procs.2015.05.248.

Teschemacher, U., & Reinhart, G. (2016). Enhancing constraint propagation in ACO-based schedulers for solving the job shop scheduling problem. *Procedia CIRP, 41,* 443–447. https://doi.org/10.1016/j.procir.2015.12.071.

Uriarte, A. G., Ng, A. H. C., & Moris, M. U. (2018). Supporting the lean journey with simulation and optimization in the context of industry 4.0. *Procedia Manufacturing, 25,* 586–593. https://doi.org/10.1016/j.promfg.2018.06.097.

Uslu, M. F., Uslu, S., & Bulut, F. (2019). An adaptive hybrid approach: Combining genetic algorithm and ant colony optimization for integrated process planning and scheduling. *Applied Computing and Informatics*. https://doi.org/10.1016/j.aci.2018.12.002. **(in press)**.

Utne, I. B., Brurok, T., & Rødseth, H. (2012). A structured approach to improved condition monitoring. *Journal of Loss Prevention in the Process Industries, 25*(3), 478–488. https://doi.org/10.1016/j.jlp.2011.12.004.

Vieira, G. E., Kück, M., Frazzon, E., & Freitag, M. (2017). Evaluating the robustness of production schedules using discrete-event simulation. *IFAC-PapersOnLine, 50*(1), 7953–7958. https://doi.org/10.1016/j.ifacol.2017.08.896.

Wardhani, R., Mubarok, K., Mucha, C., Kubota, T., Lu, Y., & Xu, X. (2018). A review on Digital Twin in manufacturing process. In *Proceedings of international conference on computers and industrial engineering, CIE* (pp. 1–15).

Wu, D., Greer, M. J., Rosen, D. W., & Schaefer, D. (2013). Cloud manufacturing: Strategic vision and state-of-the-art. *Journal of Manufacturing Systems, 32*(4), 564–579. https://doi.org/10.1016/j.jmsy.2013.04.008.

Wu, Z., Sun, S., & Xiao, S. (2018). Risk measure of job shop scheduling with random machine breakdowns. *Computers & Operations Research, 99,* 1–12. https://doi.org/10.1016/j.cor.2018.05.022.

Xu, L. Da, He, W., & Li, S. (2014). Internet of things in industries: A survey. *IEEE Transactions on Industrial Informatics, 10*(4), 2233–2243. https://doi.org/10.1109/TII.2014.2300753.

Yang, S., Bagheri, B., Kao, H. A., & Lee, J. (2015). A unified framework and platform for designing of cloud-based machine health monitoring and manufacturing systems. *Journal of Manufacturing Science and Engineering, Transactions of the ASME, 137*(4), 1–6. https://doi.org/10.1115/1.4030669.

Yu, A. J., & Seif, J. (2016). Minimizing tardiness and maintenance costs in flow shop scheduling by a lower-bound-based GA. *Computers & Industrial Engineering, 97,* 26–40. https://doi.org/10.1016/j.cie.2016.03.024.

Zandieh, M., Mozaffari, E., & Gholami, M. (2010). A robust genetic algorithm for scheduling realistic hybrid flexible flow line problems. *Journal of Intelligent Manufacturing, 21*(6), 731–743. https://doi.org/10.1007/s10845-009-0250-5.

Zhai, Y., Biel, K., Zhao, F., & Sutherland, J. W. (2017). Dynamic scheduling of a flow shop with on-site wind generation for energy cost reduction under real time electricity pricing. *CIRP Annals—Manufacturing Technology, 66*(1), 41–44. https://doi.org/10.1016/j.cirp.2017.04.099.

Zhang, J., Ding, G., Zou, Y., Qin, S., & Fu, J. (2019a). Review of job shop scheduling research and its new perspectives under industry 4.0. *Journal of Intelligent Manufacturing, 30*(4), 1809–1830. https://doi.org/10.1007/s10845-017-1350-2.

Zhang, L., Luo, Y., Tao, F., Li, B. H., Ren, L., Zhang, X., et al. (2014a). Cloud manufacturing: a new manufacturing paradigm. *Enterprise Information Systems, 8*(2), 167–187. https://doi.org/10.1080/17517575.2012.683812.

Zhang, H., Ma, L., Sun, J., Lin, H., & Thürer, M. (2019b). Digital Twin in services and industrial product service systems: Review and analysis. *Procedia CIRP, 83,* 57–60. https://doi.org/10.1016/j.procir.2019.02.131.

Zhang, H., Zhao, F., Fang, K., & Sutherland, J. W. (2014b). Energy-conscious flow shop scheduling under time-of-use electricity tariffs. *CIRP Annals—Manufacturing Technology, 63*(1), 37–40. https://doi.org/10.1016/j.cirp.2014.03.011.

Zhang, H., Zhao, F., & Sutherland, J. W. (2017). Scheduling of a single flow shop for minimal energy cost under real-time electricity pricing. *Journal of Manufacturing Science and Engineering, Transactions of the ASME, 139*(1), 1–5. https://doi.org/10.1115/1.4034275.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.