Contents lists available at ScienceDirect

Decision Analytics Journal

journal homepage: www.elsevier.com/locate/dajour

Digital-twin-based decision support of dynamic maintenance task prioritization using simulation-based optimization and genetic programming

Marcus Frantzén [a], Sunith Bandaru [b], Amos H.C. Ng [b,c,*]

[a] Department of Industrial and Materials Science, Chalmers University of Technology Gothenburg, SE-41296, Sweden
[b] Division of Intelligent Production Systems, School of Engineering Science, University of Skövde, PO Box 408, SE-541 28, Skövde, Sweden
[c] Division of Industrial Engineering and Management, Department of Civil and Industrial Engineering, Uppsala University, PO Box 534, Uppsala 75121, Sweden

ARTICLE INFO

Keywords:
Decision support systems
Digital Twin
Short-term corrective maintenance priority
Genetic programming
Simulation-based optimization
Bottleneck

ABSTRACT

Modern decision support systems need to be connected online to equipment so that the large amount of data available can be used to guide the decisions of shop floor operators, making full use of the potential of industrial manufacturing systems. This paper investigates a novel optimization and data analytic method to implement such a decision support system, based on heuristic generation using genetic programming and simulation-based optimization running on a digital twin. Such a digital-twin-based decision support system allows the proactively searching of the best attribute combinations to be used in a data-driven composite dispatching rule for the short-term corrective maintenance task prioritization. Both the job (e.g., bottlenecks) and operator priorities use multiple criteria, including competence, utilization, operator walking distances on the shop floor, bottlenecks, work-in-process, and parallel resource availability. The data-driven composite dispatching rules are evaluated using a digital twin, built for a real-world machining line, which simulates the effects of decisions regarding disruptions. Experimental results show improved productivity because of using the composite dispatching rules generated by such heuristic generation method compared to the priority dispatching rules based on similar attributes and methods. The improvement is more pronounced when the number of operators is reduced. This paper thus offers new insights about how shop floor data can be transformed into useful knowledge with a digital-twin-based decision support system to enhance resource efficiency.

1. Introduction

Industry 4.0 refers to the much anticipated fourth industrial revolution where the aim is to create intelligent factories to cope with individualized products with a short lead time to market and high productivity. Challenges such as shortened product life cycles, unpredictable customer demands, fluctuating production volumes, and shifting bottlenecks make it difficult to achieve high overall equipment efficiency (OEE), which has been noted as rather low in many companies [1]. Poor maintenance prioritization may also extend production downtime causing lower OEE, reduced productivity, and poor utilization of the allocated maintenance labor and resources [2]. Consequently, there is great potential to improve efficiency if manufacturing companies can react more flexibly and swiftly to current conditions, for example, by prioritizing the correct maintenance action, especially on the bottleneck stations. Li et al. [3] argue that further research is needed regarding maintenance scheduling in the short term (e.g., production control), as well as regarding priority between multiple bottlenecks. Bottlenecks are one of the main constraints on productivity [4]. In short-term maintenance priority, the main focus is on process control [5] and can be described as the operational period where a distribution fit is not suitable due to lack of data. As pointed out by Li and Ni [6], both the planning for maintenance opportunities and bottleneck detection could influence the decisions relating to short-term maintenance priority. In short-term maintenance priority regarding corrective maintenance, the bottleneck priority is of utmost importance as a high priority on the bottleneck stations will result in higher productivity.

Given that labor costs have increased disproportionately to productivity and may even constitute as much as 80% of the total maintenance cost, operator allocation has also gained increasing attention [7,8]. Currently, operative maintenance decisions are often based largely on the operator's experience, in spite of substantial research in this [1]. After all, humans are flexible, are able to adapt to different situations [9], and maybe good problem solvers who can grasp large amounts of data [10]. However, during running production, they are not able to process large amounts of information immediately and take decisions from a holistic system point of view. Hence, a decision support system (DSS) using data directly from the shopfloor is needed to support

* Corresponding author at: Division of Intelligent Production Systems, School of Engineering Science, University of Skövde, PO Box 408, SE-541 28, Skövde, Sweden.

E-mail addresses: marcus.frantzen@chalmers.se (M. Frantzén), sunith.bandaru@his.se (S. Bandaru), amos.ng@his.se (A.H.C. Ng).

https://doi.org/10.1016/j.dajour.2022.100039
Received 30 December 2021; Received in revised form 9 March 2022; Accepted 15 March 2022
Available online 18 March 2022

Nomenclature

List of Abbreviations

$\Delta\sigma_{\mathrm{d}}$	the percentage difference of shift utilization delta
ΔTH(%)	the percentage difference of throughput per hour
AM	autonomous maintenance
APBA	active period bottleneck attribute
ARIMA	auto-regressive integrated moving average
ARWP	assumed real-world priority
CDR	composite dispatching
CPS	cyber–physical systems
DA	distance attribute
DES	discrete-event simulation
DSS	decision support system
DT	digital twin
DT-DSS	digital-twin-based decision support system
E	expert operator
FIFO	first in first out
GP	genetic programming
GPSO-HGM	heuristic generation method using genetic programming, and simulation-based optimization
HGM	heuristic generation method
I	intermediate operator
J_	job priority rules
J_FIFO	first in first out
J_HPMA	highest parallel momentary availability
J_HWA	highest work-in-process after machine first
J_HWB	highest work-in-process before machine first
J_LIFO	last in first out
J_LAPB	least steady state active period bottleneck first
J_LMB	least momentary bottleneck first
J_LPMA	lowest parallel momentary availability
J_LWA	lowest work-in-process after machine first
J_LWB	lowest work-in-process before machine first
J_MAPB	most steady state active period bottleneck first
J_MMB	most momentary bottleneck first
J_RAND	random
KPI	key performance indicator
MBA	momentary bottleneck attribute
MES	manufacturing execution system
N	novice operator
n	number of operators
NIST	National Institute of Standards and Technology
O_	operator priority rules
O_HCOM	highest competence first
O_HDIS	highest distance first
O_HUT	highest utilization first
O_LCOM	lowest competence first
O_LDIS	lowest distance first
O_LUT	lowest utilization first
O_RAND	random
OA	order attribute
OEE	overall equipment efficiency
PA	performance attribute
PDR	priority dispatching rule
PM	professional maintenance
PMAA	parallel momentary availability attribute
SO	simulation-based optimization
TH	throughput per hour
TPM	turning point method
UA	utilization attribute
W1	work area 1
W1_2_3_4	work area 1, 2, 3, & 4
W1_4	work area 1 & 4
W2	work area 2
W2_3	work area 2 & 3
W3	work area 3
W4	work area 4
W5	work area 5
W_ALL	work area 1–5
WAA	work-in-process after machine attribute
WBA	work-in-process before machine attribute
WIP	work-in-process

the short-term maintenance priority. Methods that detect bottlenecks by collecting data directly from the manufacturing systems, without using any analytical or simulation-based method, are referred to as data-driven methods [11]. Using data-driven methods for short-term maintenance priority has been proven efficient [12], but all of these methods have their prioritizations pre-defined, without using any optimization methods, which may limit their efficiency in a dynamically changing production environment.

Short-term maintenance priority with regard to corrective maintenance (bottleneck priority), is also affected by the operators chosen to do the task. Several researchers have noted that the time to perform a job depends on the ability of the operator, but a rather common simplification is that any operators may possess the same competence/performance. However, the competence/performance of an operator may have a considerable effect on the throughput of a system [7,13–15]. Operators may also master one or several types of jobs, in which case they are defined as multi-skilled (cross-trained) operators. Multi-skilled operators may be one of the most important factors in allocating operators, including the balancing of their workloads; the effect on the productivity of different competence levels has been investigated by several authors [7,13,15,16]. Furthermore, the downtime of a machine which in turn consists of the waiting time for an operator, is dependent on both the time of noticing the machine failure and the distance to the failed machine.

Hence, based on the above discussions, there are two types of short-term shopfloor decisions related to corrective maintenance prioritizations: (1) job prioritization, when a single operator is available, and the right job needs to be prioritized, and (2) operator prioritization, when a single job requires attention and the right operator needs to be prioritized. In order to handle both "job priority", "operator priority" and adapt to a dynamically changing production environment, a Digital-Twin-based Decision Support System (DT-DSS) can be developed to address this problem. Hence, the aim of this paper is to propose such a DT-DSS for conducting short-term corrective maintenance task prioritization as well as illuminating how different job and operator priorities affect the overall productivity of a production system. A technical novelty embedded in such a DT-DSS is the heuristic generation method (HGM) using genetic programming (GP), and simulation-based optimization (SO), denoted as GPSO-HGM hereafter, to proactively search for the best attribute combinations to be used in a data-driven composite dispatching rule (CDR) for the short-term corrective maintenance task prioritization. The best attribute combinations are determined for jobs and operators based on multiple criteria as well as the attributes and methods identified from the literature, include competence, utilization, operator walking distances on the shop floor, bottlenecks, work-in-process (WIP), and parallel resource availability, as will be described in Section 2. To our best knowledge, the simultaneous job and operator priority are not studied in the literature but is deemed to be important from a practical point of view. Moreover, while there are many research studies that are based on small or theoretical production systems, complexities, such as work areas, various operator tasks and different constraints, are usually not considered. Another practical gap is that algorithms or rules in the research community do not adapt to future scenarios since they are pre-defined and may fall short to be applied when the reality changes.

A real-world application case study has been performed to evaluate the data-driven CDRs generated by the GPSO-HGM. A discrete-event simulation model has been built to simulate and evaluate the decisions made regarding disturbances on this real-world machining line. The experimental study in Section 6 of this real-world production system shows that the CDRs outperform the priority dispatching rules (PDRs), based on similar attributes and methods used in GPSO-HGM identified from literature in Section 2, for the short-term reactive maintenance priority. The experimental study produced clear results and also provided insights into the importance of an optimal prioritization of the short-term reactive maintenance tasks among machine operators, especially when the operators are reduced, e.g., due to high labor costs. Another contribution in Section 6 is the knowledge acquired when analyzing the attributes selected in the generated heuristics.

The rest of the paper is organized as follows: In Section 2 the literature of short-term corrective maintenance prioritization, different bottleneck detection methods, and data-driven methods are reviewed. Section 3 introduces the DT-DSS. The technique of using GP run on SO data to generate the data-driven CDRs is described in Section 4. Section 5 reveals the details of the real-world machining line and its simulation model used in the application study. Results and analysis of the experiments conducted with the simulation model are provided in Section 6. Finally, conclusions and future research are outlined in Section 7.

2. Literature review

Wedel et al. [17] state that maintenance prioritization uses many of the short-term and long-term bottleneck detection methods. The term "bottleneck" is a matter of definition, and there are a vast number of bottleneck detection methods and definitions. Common definitions use buffer waiting time, starvation/blockage [4], queue length or production rate [18,19]. Hopp and Spearman [20] define the bottleneck of a system as the station with the highest utilization. Shifting bottleneck detection, based on the theory of constraints [21,22] and a similar concept in [23], defines the machine with the current longest, uninterrupted average active period as the momentary bottleneck and the shifting bottleneck as the sum of sole and shifting active periods [24].

Steady-state bottleneck methods, such as active period bottleneck [25] and average shifting bottleneck [24], make use of long-term historical data about previous events. An example of the use of short-term data in combination with real-time data is the identification of momentary bottlenecks [24]. Yang et al. [26] proposed another method using short-term data (one day) to identify short-term bottlenecks. Gopalakrishnan et al. [27] demonstrated that productivity increased by approximately 5% when the momentary bottleneck was identified as the short-term maintenance priority. Another simulation study of short-term bottleneck prioritization showed a throughput increase of up to 4.5% compared to a Priority Dispatching Rule (PDR), namely the first

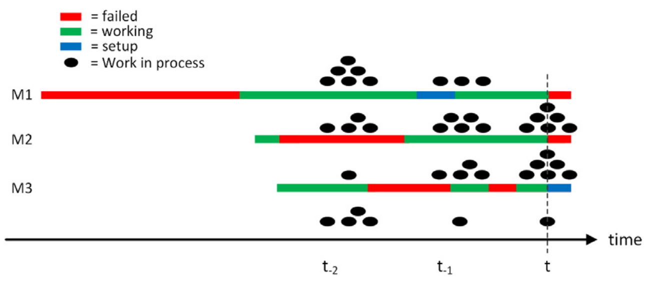

Fig. 1. Active periods and WIP.

come first served approach [17]. Li et al. [5] proposed a bottleneck control strategy that resulted in more efficient reactive maintenance task prioritization in terms of throughput. They also showed that initial buffer adjustments (e.g., after each shift) could further improve productivity. Some authors notice that none of the known bottleneck detection methods are suitable for complex machining lines characterized by parallel machines and presented three new bottleneck detection methods based on short-term, real-time, and near-future bottlenecks and showed a potential OEE increase when these are combined for short-term bottleneck prioritization [28]. The short-term method calculates the possible production loss (number of parts) caused by the bottleneck machine. The real-time bottleneck is defined as the machine that generates the highest loss of the sum of the cycle times of the idle machines. The near-future bottleneck is defined as the machine with the shortest duration until it either gets an empty downstream buffer or a full upstream buffer. Yang et al. [2] and Subramaniyan et al. [12] have pointed out that buffer levels may also be relevant for prioritizing maintenance on the line. Consider the example presented in Fig. 1, where the recent states of machines M1, M2, and M3 in a system with three successive production stages are plotted over time.

At time t, we have to decide which machine to prioritize. Based on bottleneck information, the priority would be in the order M1, M2, M3, since both the active period bottleneck and the average shifting bottleneck methods indicate the same order. However, prioritizing the failures of either M1 or M2 would not improve the system performance. Machine M1 does not have any available work orders, for the buffer upstream is empty and the machine is blocked by the full buffer downstream. Machine M2 has many work orders in the buffer upstream, but it is blocked by the full buffer downstream. Therefore, the optimal maintenance priority in this situation would be M3, even though it is not recognized as a bottleneck machine by the bottleneck methods.

Bottleneck analyzes provide information about the previous state of the machines, while WIP provides information about upcoming production and is, in a sense, a forecast of possible work orders in the buffer upstream and space in the buffer downstream. Roser et al. [29] combined the active period method with data related to WIP and buffer capacity to predict an upcoming bottleneck. Even though simulation has been used to test their methods, they aimed for a real-world implementation that would make the methods data-driven [28].

Methods that detect bottlenecks by collecting data directly from the manufacturing systems, without using any analytical or simulation-based method, are referred to as data-driven methods [11]. The Arrow method [19,30,31] identifies the bottleneck location based on blocking and starvation probabilities. A similar method that also uses blocking and starvation is the turning point method (TPM) developed in [4,5]. Their definition of a bottleneck is when the trend changes from the blockage being larger than starvation to the reverse. The TPM was also extended in [19] to include systems of decoupling buffers, enabling the identification of bottlenecks in different segments of a production line.

Subramaniyan et al. [12] implemented a data-driven bottleneck detection method based on shifting bottleneck detection [24] and validated this based on real-world manufacturing execution system

(MES) data. Subramaniyan et al. [32] extend their previous work to include diagnostic information on the proportion of different active states of the machines which is particularly important in order to find out the root causes of the bottlenecks. Later, they also integrated he data-driven active period technique with the auto-regressive integrated moving average (ARIMA) method to predict bottlenecks [33].

As commented in [34], "previous research efforts on addressing the bottlenecks primarily emphasize on the analysis of data from the physical job-shop, but with little connection and convergence with its virtual models and simulated data", DT is the technology that allows the convergence between virtual and physical spaces to be achieved. Despite DT is often described as one of the most promising enabling technologies for realizing smart manufacturing and Industry 4.0 [35,36] as well as in supporting digital transformation and decision-making in multiple industries [37], it has multifarious definitions in the literature. The National Institute of Standards and Technology (NIST) defines a DT as the electronic/digital representation of a real-world entity, concept or notion, either physical or perceived [38]. While DTs and Cyber–Physical Systems (CPS) share the same essential concepts of intensive cyber–physical connections, real-time interaction, and organization integration, CPS and DTs are not identical from many perspectives, including their origin, development, engineering practices, cyber–physical mapping, and core elements [39]. Some authors define DT as the cyber part of the CPS — a virtual representation/model that interacts with the physical system throughout its lifecycle [40]. While both CPS and DT form closed loops between the cyber/digital and physical worlds based on real-time data analysis, decision-making, and precise execution, by virtue of its virtual models, a DT provides a more intuitive and effective means of system improvement [39]. Specifically, DT allows the physical environment to be reconstructed in the virtual space to allow for simulations, forecasts, optimizations and/or decision-making which will impact the physical system of interest [37]. In terms of DT-based optimization and decision support, recent related works include dynamic job-shop scheduling and re-scheduling [34], iterative decision optimization for highly dynamic production logistics [41] and integrating a DT into a DSS for improving the order management process of manufacturing systems [42]. A recent review of implementing DT through Discrete-Event Simulation (DES) and Agent-Based Simulation for decision support of production processes can be found in [43].

In summary, the common feature of the reviewed literature shows that all of them use pre-defined rules or methods to identify and prioritize the bottlenecks. Different attributes and methods have been identified as important, e.g., shifting bottleneck detection method, WIP, buffers, parallel resources, blocking, starvation, cycle times. Many of these methods may perform well on a certain set of problems but are not likely to perform well over a range of different production systems or if the setting changes. In addition, by relying only on analysis data from the physical shopfloor without using any virtual models for performing prediction and optimization, these methods also fall short of performing prescriptive analytics. To overcome these issues, a DT-DSS for short-term maintenance prioritization is proposed in Section 3 and its GP and SO components embedded in the GPSO-HGM are described in Section 4.

3. Digital-twin-based maintenance prioritization

In order to adapt to a dynamically changing reality at the production shopfloor the concept of DT [44] may be used to address short-term maintenance priority. As discussed in Section 2, there are many definitions of what a DT is. In order not to simply rename technology that has existed for many years [45], e.g., "digital model" there should be a bi-directional data flow between the physical object and the digital object [46]. Hence, in the context of the current paper, a DT is defined as "a digital representation of an existing physical object or system where an automatic real-time (or near real-time) bi-directional data flow is used between the virtual and the physical object or system to enable autonomous decision making or decision support". The virtual entity is of particular importance of a DT and can consist of one or several simulation models [47]. Discrete event simulation modeling can represent complex real-world systems in detail, which is one of its main advantages over other methods. It is also very useful for communicating details, such as a maintenance priority situation, because most simulation software includes visual aids. Once a system has been modeled, simulation allows better predictions of dynamic behavior [48]. The need for simulation models is increasing in order to conduct analyzes without disturbing the real-world system and also allow engineers to analyze and improve future setups of the production system.

The DT-DSS for short-term maintenance prioritization proposed in this paper is shown in Fig. 2 where all different parts of the proposed extended five-dimension digital twin [49] exist, i.e., physical entity, Virtual entity (simulation model), Services (User-interface applications), data and connections. The simulation model can be used to capture the real-world status (Digital Shadow) [46] when integrated with the physical entity, information systems (e.g., manufacturing execution system) and other related knowledge. The generative model (Artificial Intelligence) is used to generate optimal or near-optimal solutions to update the data-driven priority "on-time" which would satisfy the bi-directional data flow of a DT. The data-driven priority may be a computer program, dispatching rule, or other types of heuristics.

As a part of the DT-DSS, adaption to a dynamically changing reality at the production shopfloor is possible. The GPSO-HGM using GP and SO can address the shortcomings of other data-driven methods, namely that they usually are pre-defined. The system also supports proactively exploring and prescribing new optimal or near-optimal prioritizations based on future scenarios, such as changes in bill of process, layout, or way of working.

4. Genetic Programming based Simulation-Optimization (GPSO-HGM)

Instead of deciding which method or priority rule to use for short-term corrective maintenance priority, as is the case in most of the reviewed literature, the idea here is to autonomously generate optimal or near-optimal Composite Dispatching Rule (CDR) using a combination of genetic programming (GP) [50] and simulation. This approach is similar in principle to typical simulation-based optimization tasks, where a simulation model is integrated with metaheuristic methods, such as a genetic algorithm (GA) [51,52] or Tabu Search [53]. For example, GAs have specifically been used to find the optimal combination of Priority Dispatching Rules (PDRs) [54–56].

4.1. Genetic Algorithms versus Genetic Programming

The basic framework of a GP is very similar to that of a GA, in that it consists of four steps, namely initialization, selection, reproduction, and replacement. The initialization step involves generating random candidate solutions to form the initial population. These solutions are evaluated with respect to one or more objectives in the selection step to identify the best solutions, which form the so-called mating pool. The reproduction step consists of two operations — crossover and mutation. The purpose of crossover is to recombine the information present in the "parents" from the mating pool and generate new (and hopefully better) "offspring" solutions. This is followed by a mutation operation that introduces possible random alterations in the offspring with the purpose of preserving/promoting diversity in the population. The new offspring are evaluated with respect to the objective(s) and form the next population of solutions in the replacement step. These four steps are repeated until a termination criterion is met, whether this be the number of iterations, the algorithmic runtime limit, or the convergence rate.

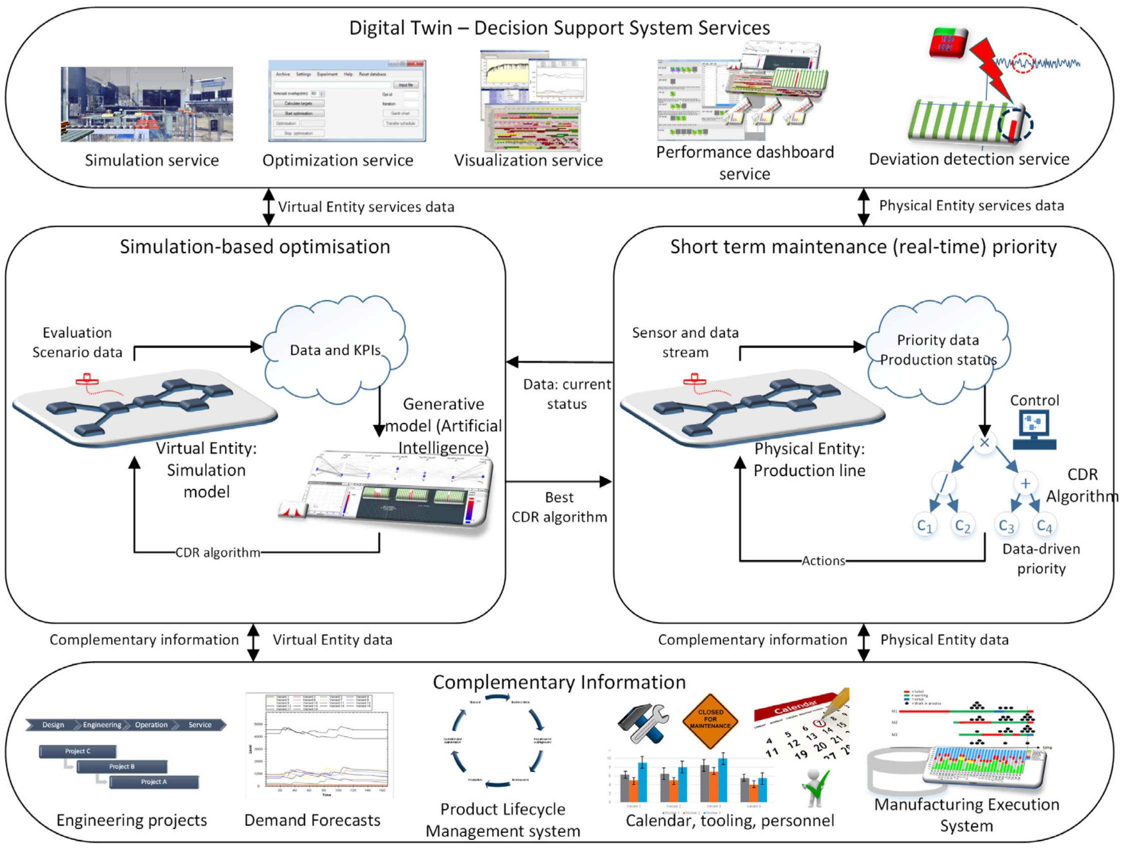
Digital Twin – Decision Support System Services
Simulation service
Optimization service
Visualization service
Performance dashboard service
Deviation detection service
Virtual Entity services data
Physical Entity services data
Simulation-based optimisation
Evaluation Scenario data
Data and KPIs
Generative model (Artificial Intelligence)
Virtual Entity: Simulation model
CDR algorithm
Data: current status
Best CDR algorithm
Short term maintenance (real-time) priority
Sensor and data stream
Priority data Production status
Control
Physical Entity: Production line
CDR Algorithm
Data-driven priority
Actions
Complementary information
Virtual Entity data
Complementary information
Physical Entity data
Complementary Information
Engineering projects
Demand Forecasts
Product Lifecycle Management system
Calendar, tooling, personnel
Manufacturing Execution System

Fig. 2. A DT-DSS for short-term maintenance prioritization.

The main difference between a typical GA and GP is that while the former only allows search over variable values for a fixed solution structure, GP even allows changes to the structure using a special solution representation that encodes this structure. Thus, GP is a lot more versatile than GA. One of the most common applications of GP is symbolic regression. In a typical regression task (such as polynomial regression), the underlying regression model remains fixed (say quadratic), and the problem is essential to estimate the coefficients of this predetermined model. In symbolic regression, the underlying model is not assumed but is rather derived from data by combining "symbols" from a given *function set* and *terminal set*. The function set may consist of mathematical operators like '+', '-', '*' and '/', or functions like 'sin()', 'tanh()' and 'exp()'. The terminal set may consist of variables, ephemeral random constants, and nullary functions/operators. While the typical regression task can be easily performed using the statistical method of multiple linear regression, the symbolic regression problem can only be effectively solved through GP. More generally, GP systems can also be used for automatically generating computer programs to perform specified tasks. In that case, the function set usually includes conditional functions {IF-THEN-ELSE}, Boolean functions {AND, OR, NOT}, looping functions {FOR, REPEAT}, while the terminal set includes external program inputs.

The most common way to represent GP programs is using parse trees, whose intermediate nodes come from a function set, while all leaf nodes come from the terminal set. The tree is typically interpreted from left to right in depth-first order [57], as shown in the examples in Fig. 3. Evolving parse tree without constraining GP can sometimes lead to the problem of *bloating*, where the tree contains large redundant parts. Bloating can be avoided by constraining the maximum depth (number of levels), or the maximum length (total number of nodes) of the generated trees.

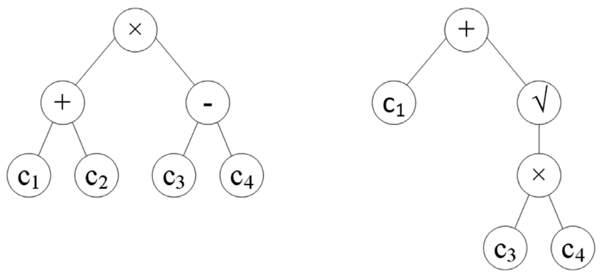

Fig. 3. Examples of parse trees formed using the function set F = {+, -, *, $\sqrt{}$} and the terminal set T = {c1, c2, c3, c4}. The parse tree on the left is interpreted as (c1 + c2) * (c3–c4), and the tree on the right as c1 + $\sqrt{}$(c3 * c4).

4.2. Solution representation in GPSO-HGM

In the proposed GPSO-HGM approach, the parse trees represent CDRs for short-term corrective maintenance. They are encoded in the form of integer strings (genotypes) to simplify the genetic operations of crossover and mutation. Fig. 4 shows how a given genotype can be decoded to obtain the parse tree, which is then interpreted as a CDR (phenotype). In this particular example, the number of levels of the parse tree is restricted to three levels to avoid the problem of bloating mentioned above. This is done by constraining the number of operators that can be used from the function set. As shown in the figure, the first three bits are reserved for the function set, while the next four are reserved for the terminal set. The function set consists of the operators {$\sqrt{}$, *,/, +, -}, with the constraint that operator '$\sqrt{}$' cannot be placed in the first bit. The terminal set consists of variables and constants. Variables can be any of the priority attributes described later in Section 4.3, but for the purpose of illustration, we assume in

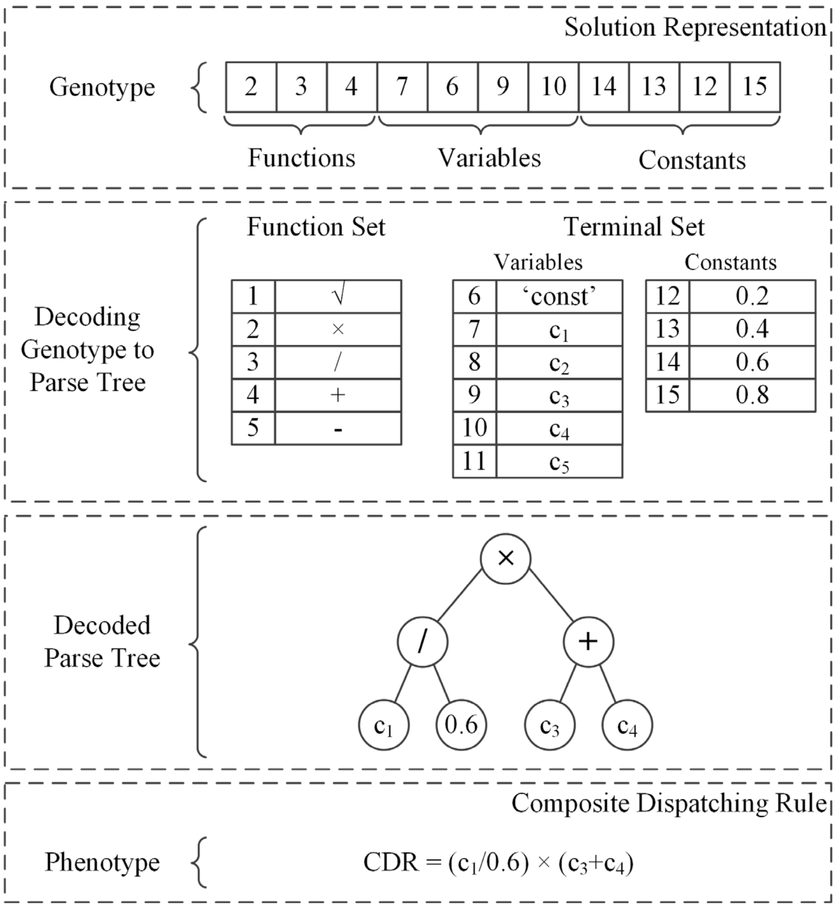

Fig. 4. Steps in decoding the integer string genotype into a parse tree which is interpreted as a CDR.

Fig. 4 that there are five priority attributes (c1, c2, c3, c4, c5) available. The terminal 'const' is used to symbolically represent a constant value. Thus, the terminal set is {'const', c1, c2, c3, c4, c5}. The number of possible values that 'const' can take needs to be at least the number of possible terminal nodes. In this instance, there are four constant values, namely (0.2, 0.4, 0.6, 0.8), since with three levels, the parse tree can have a maximum of four-terminal nodes. If one or more of the terminals is 'const', the values are taken from the last four bits of the genotype.

Once the genome has been created, it is used to generate a parse tree by going left to the right in depth-first order using the mapping of the functions, variables, and constants. The list of functions, variables, or constants can easily be extended if necessary. The variables need to be normalized in order to have similar weights when learning begins and so that the rules can be analyzed more easily afterward. The constants were pre-specified to between zero and one to match the variable normalization. When the parse tree has been generated, it is interpreted into a CDR by going left to right in depth-last order. The CDR or function (phenotype) is then used as a dispatching rule in the simulation model when executed. An example of the process of going from a genotype to a decoded parse tree to finally generate a CDR (phenotype) is shown in Fig. 4.

Even if an integer representation is used to avoid invalid solutions, there are still situations where the genome is valid, but the combination of values in a formula is not. Because some variable values are dependent on the current status of the system, it is possible to end up with an expression that becomes zero or even negative, which is a problem when using division or square roots. Therefore, all invalid calculations need to be returned as invalid solutions. To handle invalid solutions, all anomalies are logged as performance measures to be minimized. A value of zero is returned for division by zero; the protected square root of negative numbers is returned by using absolute values.

4.3. Priority attributes for CDRs and PDRs

Some of the most promising data methods identified in the literature review in Section 2 are the active period bottleneck, momentary bottleneck, WIP, buffer capacity, and parallel machines for job prioritization. Furthermore, the performance of the operators and the time to reach the machine based on distance and workload balancing (utilization) are important for the operator prioritization. The GP needs different variables/attributes in order to generate different CDRs. The list of attributes could be anything from processing time and capacity to queue length, but the objective here is to select the vital few attributes needed for operator and job priority. The list of the priority attributes proposed to find operator- and job CDRs can be found in Table 1.

All of these attributes were normalized to a value between 0 and 1. The attribute PMAA is used to describe the proportion of parallel stations currently available, that is, those not setting up or failing. Stations that are not part of or affecting the main production line are assumed to have a 100% PMA regardless of status. The dispatching rules proposed are also based on the criteria identified in Section 2 and, consequently, map well to the CDR attributes proposed. Common rules, such as "random" and first in first out (FIFO) have also been added. When applicable, each attribute generates two dispatching rules, e.g., lowest and highest levels. The following dispatching rules for prioritizing operators and jobs are proposed in Table 2.

5. Application study on a real-world machining line

In order to evaluate the CDRs generated by the GPSO-HGM, a real-world case study has been used. A discrete-event simulation model has been built to simulate and evaluate the decisions made with regard to

Table 1
List of operator and job priority attributes.

Priority attributes	Formula	Priority type
Performance (PA)	$Performance\ factor/Novice\ performance\ factor$	Operator
Utilization (UA)	$(Working\ time + Walking\ time)/Available\ time$	Operator
Distance (DA)	$Distance/(Xdim + Ydim)$	Operator
Order (OA)	$FIFO\ number/Num\ jobs\ pending$	Job
Active period bottleneck (APBA)	$Working\ time + Failed\ time + Setup\ Time$	Job
Momentary bottleneck (MBA)	$Momentary\ Bottleneck\ Time/Primary\ Momentary\ Bottleneck\ Time$	Job
WIP before machine (WBA)	$WIP\ before\ machine/Capacity\ before\ machine$	Job
WIP after machine (WAA)	$WIP\ after\ machine/Capacity\ after\ machine$	Job
Parallel momentary availability (PMAA)	$Num\ available\ machines/Total\ num\ machines$	Job

Table 2
List of operator and job priority dispatching rules.

Priority Dispatching Rules (PDRs)	Priority type
lowest competence first (O_LCOM)	Operator
lowest utilization first (O_LUT)	Operator
lowest distance first (O_LDIS)	Operator
highest competence first (O_HCOM)	Operator
highest utilization first (O_HUT)	Operator
highest distance first (O_HDIS)	Operator
random (O_RAND)	Operator
first in first out (J_FIFO)	Job
last in first out (J_LIFO)	Job
most steady state active period bottleneck first (J_MAPB)	Job
least steady state active period bottleneck first (J_LAPB)	Job
most momentary bottleneck first (J_MMB)	Job
least momentary bottleneck first (J_LMB)	Job
highest WIP before machine first (J_HWB)	Job
lowest WIP before machine first (J_LWB)	Job
random (J_RAND)	Job
lowest WIP after machine first (J_LWA)	Job
highest WIP after machine first (J_HWA)	Job
lowest parallel momentary availability (J_LPMA)	Job
highest parallel momentary availability (J_HPMA)	Job

disturbances on this real-world machining line. Additional details of the real production system and its simulation model are provided in the next sub-sections.

5.1. The production system

The real-world production system selected is a machining line in the automotive industry. It has a high production quantity with low product variety, manufacturing tens of thousands of parts every year in a two-shift production schedule. There are three different areas in the production line, namely, rough machining, fine machining, and inspection. At the start of the line, manual work is involved in packing and loading parts onto the production line, but the rest of the line is semi-automated. One cell, consisting of two or more resources from one or more production stages, uses gantry robots that load and unload the resources for milling, drilling, grinding, and washing. The machines are arranged in sequence in the flow shop production line, also referred to as a product-oriented layout.

Preventive maintenance activities are divided into either professional maintenance (PM) or autonomous maintenance (AM) procedures. The engineers and technicians of the maintenance department carry out the PM activities for four hours once a week. The AM activities occur at regular intervals and are carried out by the shop floor operators during the workweek. In this sense, the preventive maintenance tasks are calendar-based while corrective maintenance tasks in response to such things as tool breakage are handled by the operators after receiving an alarm on their handheld phones.

Currently, production personnel on the shop floor (operators and shift leaders) decide which of the operators' tasks to prioritize (failures, tool changes, quality controls, and material handling). Given the difficulty of grasping all the information, their decisions are not made with the overall performance of the line in mind. As can be seen in Fig. 5 which depicts production flow and floor areas of the studied system, the line is currently divided into five work areas with usually seven operators on duty per shift, covering the eight gantry areas, including 19 production stages and 28 machines. The manufacturer has noted that current operator utilization is uneven over time, work areas, and shifts.

In Fig. 5 the different work areas are denoted as W1–W5 and the number of machines in each area are shown by the dark blue squares. Due to a higher workload of W1 and W5, these areas normally have two operators per area. Different work areas may be grouped together on the basis of proximity and types of competence needed, for example, W1 and W4 can become W1_4, and W2 and W3 can become W2_3. Furthermore, the rough machining area can be grouped into one large work area, namely, W1_2_3_4; is the fine machining area is also included, the group is named W_ALL. The competence/performance levels of the operators are defined on three levels, namely, novice (N), intermediate (I), and expert (E). These three levels of competence are formulated as time factors multiplied by the length of activities/tasks, that is, a lower time factor corresponds to a higher level of competence and less time to execute the task. On the basis of interviews at the company, the performance/competence factors were set to 1.5 for "N", 1 for "I" and 0.8 for "E".

5.2. The discrete-event simulation model

The discrete-event simulation model, built using Siemens Plant Simulation software, represents the real-world production line and generates maintenance events based on real-world data gathered from the equipment. The simulation model contains the detailed logic of equipment such as gantry robots and machines, as well as other logic, such as the different activities carried out by the operators. The fidelity of the model is high so that events such as failures, frequent tool changes, quality controls, material handling, and other tasks can be simulated. The operators work in different work areas, as mentioned in the previous section. The simulation model allows operators with different levels of performance to be assigned to different pre-defined work areas but also allows operators to have individual work areas and different performance per activity. The walking speed (1.3 m/s) of the operators was modeled according to the company's data that they use according to their rules and regulations. It is well in line with physiological data studies of what speed that represents a normal walking pace, i.e., 1.33 m/s [58].

When simulating maintenance activities, it is important to include the other tasks carried out by the operators since these activities may affect their availability. Thus, the model needs to include planned maintenance, corrective maintenance, material handling, tool handling, and quality control. Other tasks carried out by the operators that do not affect production directly involve the grinding of tools (i.e., external setup). It is crucial that the online maintenance priority system get real-time data from systems such as MES. Online information about WIP, breakdowns, tool changes, operator position, and momentary (shift) operator utilization was implemented in the simulation model. Due to the operative nature of this study, long-term improvements such as skills development and learning are excluded, but these will be considered in future work.

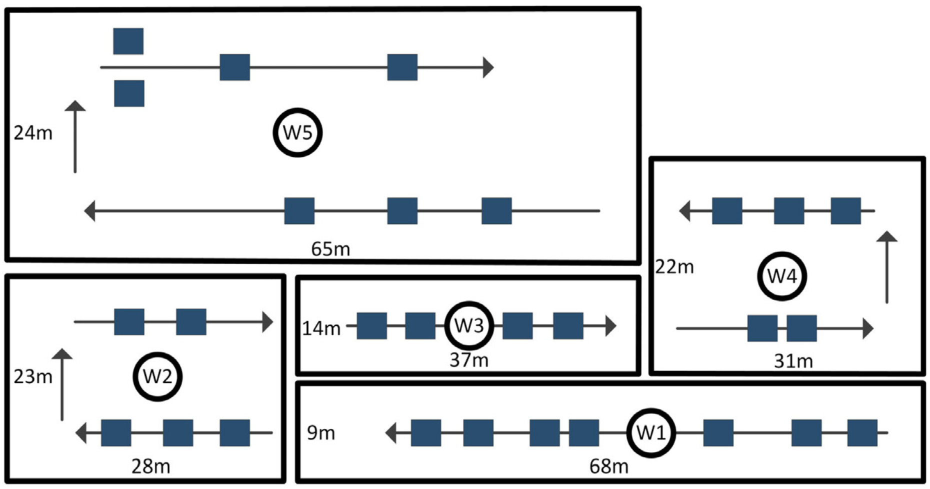

Fig. 5. Work areas and production flow on the snapshot of the simulation model.

Validation of the simulation model, that is, the comparison of the results of the simulation model with real-world data, was based mainly on throughput (TH). It is, however, impossible to know exactly how the operators prioritize their tasks and how this affects the line performance. As previously mentioned, the short-term corrective maintenance prioritization can be divided into (1) job priority, when one operator is available, and the jobs need to be prioritized, and (2) operator priority, when one job is available, and the right operator needs to be prioritized. The job priority is assumed to be first in first out (J_FIFO), and the operator priority is assumed to be random (O_RAND). With regard to the operator prioritization in the simulation model, the TH output was found to be within 3% of the data collected for the same three-month period. This deviation is not considered significant. Utilizations of the individual operators were validated as well. Additionally, validation using a structured model walkthrough of the simulation model to approve model assumptions was carried out together with the production engineers. Manhattan distance, that is, the distance between two points based on the sum of the horizontal and vertical distance instead of merely the diagonal distance, was used to calculate the walking distance for the operators. The raw material is assumed to be available at the beginning of the line. The simulation model supports a library of priority dispatching rules (PDRs) as well as CDRs generated by the GPSO-HGM.

6. Experimental results and analysis

The aim of this experimental study is to compare the results of the generated CDRs with PDRs based on the identified attributes and rules in Section 5.1. Furthermore, the aim is also to gain knowledge of how different attributes and levels of operators affect the results. In the following sub-sections, the experimental settings are summarized, and the numerical results of the experimental study are presented and analyzed.

6.1. Experimental settings

The main objective of the study is to maximize the percentage increase of TH, referred to as TH delta or ΔTH(%), compared to the original TH obtained from the real line using the current manual operator prioritization. In other words, the jobs per hour in the validation scenario (see Section 6.2) are compared with assumed real-world priority. In the study, 0 means the same throughput as the validation scenario, and 1 means 1% higher TH than the validation scenario. An objective called shift utilization delta, denoted as $\Delta\sigma_d$, is introduced into the optimization to minimize the deviation of the operator utilization. It is calculated by taking the standard deviation of the operator utilization per shift and then taking the average value of all standard deviations.

The main scenario represents the current real-life situation in which seven operators are employed in five work areas. However, the initial simulation experiments found that this scenario is inherently problematic because the work areas are not very large and many operators can take the jobs, leading to poor utilization of the workforce. Therefore, experiments were also conducted to assess the impact on the priorities when the number of operators is reduced and the size of the work area is altered. The number of work areas and the operator competence level needed in each scenario were estimated together with the engineers at the company, for one of the areas generally requires a higher level of competence. The following work area setups were used for the different scenarios (N = novice, I = intermediate, and E = expert):

- Seven operators using the work area setup: W1: 1N, 1E; W2: 1E; W3: 1I; W4: 1I; W5: 1I, 1E.
- Six operators using the work area setup: W1_2_3_4: 1N, 1I; W5: 1I, 1E; W_ALL: 1I, 1E.
- Five operators using the work area setup: W1_2_3_4: 1N, 1I; W5: 1E; W_ALL: 1I, 1E.
- Four operators using the work area setup: W1_2_3_4: 1I; W5: 1E; W_ALL: 1N, 1E.
- Three operators using the work area setup: W_ALL: 1N, 2E.

Two types of experiments were carried out for each scenario, namely:

- Simulation experiments of full factorial experimental design with PDRs.
- GPSO-HGM to generate CDRs.

The GPSO-HGM used a training set of five replications and the best solutions were verified by another 100 replications (validation set). Each replication is based on 20 production weeks, that is, 100 days (7-day warm-up time and 107-day simulation horizon). The t-distribution, also known as Student's t-distribution, is a type of normal distribution used to determine whether the results are statistically significant when the variance in the data is unknown. We want to be 95% confident that the average results of two groups (obtained different PDRs) differ, which is measured by the resulting "p-value". If the p-value is less than or equal to 0.05 (1–0.05 = 0.95 = 95%) when using the t-distribution we can be confident (i.e., 95% confidence level) that the average results of two groups, e.g., results of different PDRs, differ.

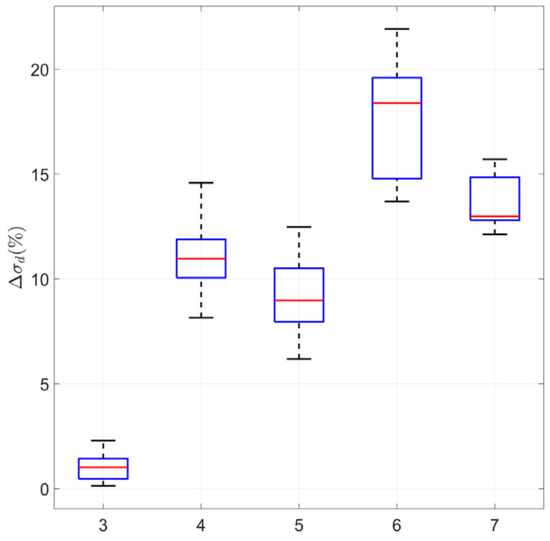

Fig. 6. Utilization difference with scenarios having n operators.

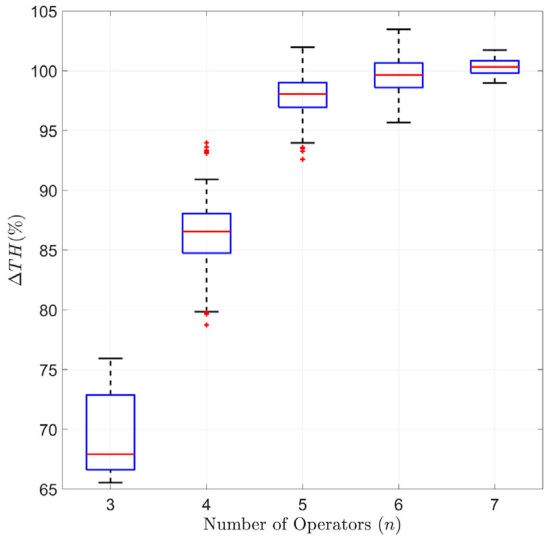

Fig. 7. Difference in overall throughput per hour.

6.2. Maintenance prioritization

In this sub-section, the results of the scenario with seven operators are first presented in detail for both full factorial and GPSO-HGM. After that, the overall results of all the experiments are presented and analyzed.

6.2.1. Full factorial experimental design with PDRs

Five scenarios of two-factor full factorial experimental designs with a total of 91 different priority rule combinations in each scenario were evaluated. They had 7 PDRs of operator priority and 13 PDRs of job priority. A boxplot of the utilization difference ($\Delta\sigma_d$) is shown in Fig. 6.

In Fig. 6, $\Delta\sigma_d$(%) is the utilization difference with scenarios having n operators. Each one of the boxplots is the summarized result for a full factorial experiment. The utilization difference is affected by different rules, but it is mainly the scenario that seems to set the range of which the values vary. All scenarios, except the scenario with three operators, cannot reach a low utilization difference because of the different workloads of the work areas.

A boxplot of the TH delta (ΔTH) is shown in Fig. 7.

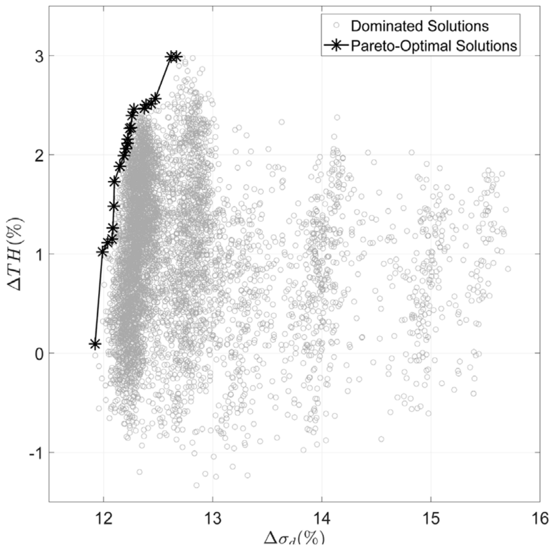

Fig. 8. Optimization results of using seven operators with CDRs.

In the results presented with Fig. 7, ΔTH(%) is the jobs per hour in relation to the validation scenario with assumed real-world priority (ARWP) representing 100% with 7 operators. The TH is reduced when the number of available operators is reduced. It is interesting to note the wide range of results in all the scenarios except the scenario with seven operators (n7). Consequently, there is a potential for improvement.

Studying the scenario of seven operators in detail, the results vary from –1.0 to 1.5 in ΔTH(%) related to the validation scenario with the assumed real-world priority (J_FIFO and O_RAND) and from 12.1 to 15.7 in $\Delta\sigma_d$. The average utilization (work) of operators is approximately 50%–55% of the available time (excluding breaks and planned maintenance), which is quite a low utilization level. Table 3 gives an insight into the overall main effects of the different rules relative to the whole set of solutions/responses from each scenario.

As presented in Table 3, the main effects show the main percentage effect that the rules affect TH (higher is better) and $\Delta\sigma_d$(lower is better) in each one of the operator scenarios. It seems that the job priority rules (J_) do not affect the utilization difference to a great extent unless the operators have high overall utilization, i.e., when operators are below six. The operator priority rules (O_) have a greater impact on the utilization difference, but it is possible to see that this impact is minor when the number of operators is few. The job priority rules have a much greater impact on TH compared to the operator priority rules, especially when the operators are few. The main objective ΔTH(%) will be analyzed in further details in the following sub-sections.

6.2.2. Optimizing CDRs using GPSO-HGM

Fig. 8 presents the Pareto-optimal solutions obtained by using GPSO-HGM to generate CDRs based on the scenario with seven operators.

It is possible to achieve a ΔTH(%) of 2.8 and still retain the $\Delta\sigma_d$ at 12.8%. The best CDRs (BCDRs), generated by GPSO-HGM, yield a significantly higher TH compared to the best PDRs (BPDRs) found (O_LDIS and J_MAPB). Consequently, an analysis of what makes the CDRs generated by GPSO-HGM better than ordinary PDRs is in order.

The operator PDRs that on average achieve the highest ΔTH(%) are also part of the CDR generated by GPSO-HGM, namely, performance attribute (PA) and distance attribute (DA). The best CDR found from that experiment yielded the following equation for operator CDR: $((0.4-PA) * (PA * DA))$. The results of the CDRs are sorted in decreasing order, so a higher value is prioritized. Studying the formula in greater detail reveals that an operator with a higher level of skill/performance

Table 3
Main effects of the dispatching rules (an explanation of the abbreviated dispatching rules can be found in Table 2).

PDR	n_7		n_6		n_5		n_4		n_3	
	Main effect TH	Main effect σ_d	Main effect TH	Main effect σ_d	Main effect TH	Main effect σ_d	Main effect TH	Main effect σ_d	Main effect TH	Main effect σ_d
O_LCOM	0,15%	1,83%	−0,18%	1,92%	−0,07%	1,55%	0,01%	0,47%	−0,24%	0,24%
O_LUT	−0,10%	−1,28%	0,18%	−3,88%	0,03%	−2,41%	−0,08%	−0,74%	−0,13%	−0,33%
O_LDIS	0,45%	−0,80%	1,27%	−2,98%	0,96%	−1,21%	1,16%	−0,57%	1,59%	0,02%
O_HCOM	−0,21%	0,37%	0,14%	0,65%	0,07%	−0,44%	−0,13%	−0,35%	−0,04%	−0,06%
O_HUT	0,12%	1,31%	−0,30%	3,64%	−0,09%	2,12%	−0,04%	0,74%	−0,18%	0,36%
O_HDIS	−0,30%	−0,63%	−0,97%	1,15%	−0,87%	0,53%	−0,87%	0,45%	−0,95%	−0,13%
O_RAND	−0,12%	−0,80%	−0,13%	−0,51%	−0,04%	−0,14%	−0,05%	0,02%	−0,05%	−0,10%
J_FIFO	−0,18%	0,01%	−0,01%	0,00%	−0,08%	0,08%	0,62%	−0,83%	−1,90%	0,17%
J_LIFO	0,03%	−0,02%	0,25%	0,01%	0,37%	−0,07%	0,16%	0,65%	−4,15%	0,57%
J_MAPB	0,90%	0,12%	2,66%	−0,21%	3,69%	−0,93%	8,11%	−2,60%	0,92%	−0,41%
J_LAPB	−1,01%	−0,14%	−3,01%	0,19%	−4,36%	0,98%	−7,71%	2,33%	−3,24%	0,38%
J_MMB	0,56%	0,09%	1,15%	0,01%	1,28%	−0,03%	1,95%	0,12%	5,52%	−0,74%
J_LMB	−0,67%	−0,10%	−1,81%	−0,01%	−2,26%	0,11%	−5,09%	0,76%	−4,67%	0,60%
J_HWB	0,64%	0,11%	1,94%	−0,06%	2,33%	−0,34%	4,29%	−1,33%	−1,43%	0,10%
J_LWB	−0,52%	−0,06%	−1,58%	0,04%	−1,75%	0,31%	−2,85%	1,04%	5,70%	−0,39%
J_RAND	0,15%	0,02%	0,07%	−0,02%	0,29%	−0,13%	0,35%	−0,34%	−2,07%	0,26%
J_LWA	0,65%	0,01%	0,50%	0,05%	0,46%	0,19%	−0,87%	0,75%	5,86%	−0,84%
J_HWA	−0,57%	−0,01%	−0,32%	−0,01%	−0,25%	−0,12%	0,23%	−0,66%	−4,67%	0,59%
J_LPMA	0,41%	0,05%	0,60%	−0,05%	0,90%	−0,17%	2,41%	−0,66%	7,85%	−0,74%
J_HPMA	−0,39%	−0,08%	−0,46%	0,06%	−0,62%	0,13%	−1,58%	0,77%	−3,74%	0,43%

Table 4
Results comparison of the best rules found (an explanation of the abbreviated dispatching rules can be found in Table 2).

	ARWP		BPDRs			BCDRs		
	Rules	ΔTH(%)	BPDRs	ΔTH(%)	ΔTH_n(%)	BCDRs	ΔTH(%)	ΔTH_n(%)
$n = 7$	J_FIFO O_RAND	0.00	J_MAPB O_LDIS	1.48	1.48	((APBA + APBA)+(WBA - OA)) ((0.4-PA)*(PA*DA))	2.82	2.82
$n = 6$	J_FIFO O_RAND	−0.67	J_MAPB O_LDIS	3.09	3.79	((APBA/0.4)+(WBA*0.4)) ((0.7/UA)-(DA*PA))	3.51	4.21
$n = 5$	J_FIFO O_RAND	−2.29	J_MAPB O_LDIS	2.11	4.50	((APBA+APBA)+(WBA+APBA)) ((1-UA)-(DA*0.5))	2.49	4.89
$n = 4$	J_FIFO O_RAND	−13.15	J_MAPB O_LDIS	−6.07	8.15	((APBA*OA)+(WBA+APBA)) ((UA-0.6)-(DA-0.8))	−5.55	8.75
$n = 3$	J_FIFO O_RAND	−31.88	J_LPMA O_LDIS	−24.14	11.36	((1+MBA)+(WBA+WBA))((0.1-DA)-(DA*0.9))	−15.79	23.63

is prioritized when operators have the same position, although the distances of the operators are also considered in the CDR.

The best CDR from that experiment yielded the following equation: $((APBA + APBA) + (WBA - OA))$. One of the main attributes in the job CDR is thus active period bottlenecks, which were previously identified as the PDR (J_MAPB) with the highest average ΔTH(%). Furthermore, the WIP before a machine is another important attribute of the job CDR, which also matches the results of the full factorial experiment in which WIP before and after are important in order to achieve a higher TH. FIFO order, that is, the order attribute (OA), is the last ingredient in the job CDR. A machine/resource with long, average active periods and (secondary) high WIP before the machine is prioritized, unless many jobs are pending and the current machine recently required service.

6.2.3. Overall comparison of results

Table 4 shows the overall results of the different operator scenarios for the main objective ΔTH(%). The columns show the assumed real-world priority (ARWP), best PDRs (BPDRs), best CDRs (BCDRs).

In Table 4, ΔTH(%) is the jobs per hour in relation to the validation scenario (7 operators) with ARWP, and ΔTH_n(%) is the jobs per hour with scenarios having n operators, compared to the real-world priority with the current number of operators. The best PDRs (BPDRs) consequently achieve significantly higher ΔTH(%) than the assumed real-world priority (ARWP) (confirmed with a t-test), while the best CDRs (BCDRs) have significantly higher ΔTH(%) than the BPDRs (confirmed with a t-test), especially when the number of operators, n, is decreasing. The average results and the confidence intervals of the ARWP, best PDRs, and best CDRs, are plotted in Fig. 9.

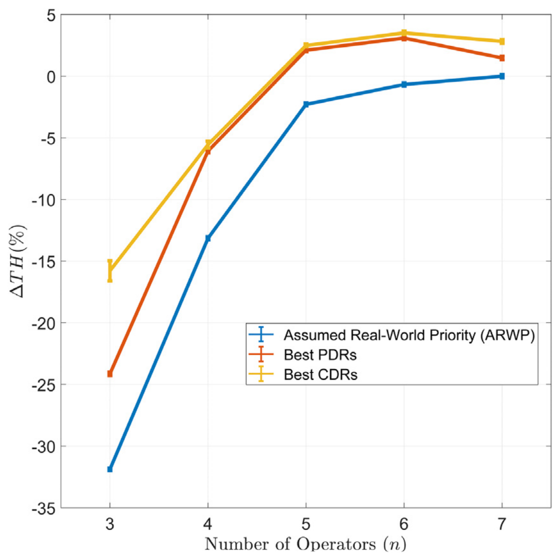

Fig. 9. Real-world priority compared to the best PDRs and CDRs.

Comparing the results of the dispatching rules to the ARWP reveals that short-term corrective maintenance priority has a major impact on TH, especially when the operators have higher utilizations. The differences between the assumed real-world priority and the best rules found are substantial. The resemblance, in terms of attributes, between the best PDRs and best CDRs is generally considerable. However, the optimization results using GPSO-HGM to generate CDRs consistently achieve significantly better results than the best PDRs found for each scenario because GPSO-HGM is able to use an optimal combination of different attributes. It is interesting to note that the scenario of six operators ($n = 6$) can get a slightly higher throughput compared to seven operators ($n = 7$) due to a minor work area flexibility in $n = 6$ as described in the experimental settings of Section 6.1. Furthermore, the throughput is only slightly decreased when going from six ($n = 6$) to five ($n = 5$) operators. On top of that, by using a better priority (BPDRs or BCDRs) a higher throughput with five ($n = 5$) operators can be attained when compared to seven ($n = 7$) operators using ARWP. Hence, it is possible to have a big labor cost reduction whilst maintaining throughput (or even increasing throughput).

7. Conclusions and future work

Maintenance activities on manufacturing shop floors need smart prioritization approaches to enhance resource efficiency, productivity, and indirectly the profitability of companies. Maintenance digitalization using sensory data, data analytics, and mobile visualization in the development of DSSs is important in order to achieve smart maintenance in line with the Industry 4.0 initiative. However, one obstacle to fully digitalizing the gathering of online manufacturing data using MES is that companies may not know the full potential of the data or grasp how it should be processed and transformed into usable knowledge in a digital twin DSS, even though they may have full access, in principle, to any data.

This research study has shown that by using online data together with SBO and advanced data analysis technologies, a shop floor DSS is able to adapt to disturbances in a production system and support the making of optimal or near-optimal decisions in real-time. While Industry 4.0 in general refers to the intelligent networking of machines and processes for the industry with the state-of-the-art information and communication technologies [59], the full potential of smart maintenance for improving overall system performance has not been clearly demonstrated, especially in a quantified way. This paper has contributed toward smart, optimized maintenance by proving that optimization of the core part of short-term corrective maintenance prioritization can affect and improve the overall line performance — an important aspect that has not been addressed explicitly in other smart maintenance studies. More specifically, this paper successfully introduces and implements a way of prioritizing short-term corrective maintenance activities (planned and unplanned), based on multiple criteria, including bottleneck, WIP level, competence and utilization of operators, and walking distances on the shop floor with a DT-based SO and GP approach. The novel GPSO-HGM employs a combination of SO and GP to generate data-driven composite dispatching rules has also been introduced. The results generated from a simulation model of a real-world automotive machining line show statistically significant improvements of the system throughput when current priorities based mainly on experience are changed to customized rules generated by GP. The experimental results also clearly show how optimal short-term corrective maintenance prioritization can have a significant impact on the production rate, especially when the number of operators is decreased. The DT-DSS, especially the simulation model, would also allow the evaluation of changed work areas over time and greater attention to the skills of the operators. As noted in the results presented in Section 6.2.3, it is possible to have big labor cost reductions whilst maintaining throughput when the work areas and the priorities are changed or optimized.

Future research will study how different levels of operator utilization affect the results. Different ways of generating CDRs using GPSO-HGM could also be studied. Due to the operative nature of this study, skills development and learning, which are related to strategic, long-term improvement, were not considered. Obviously, skills or competence development could be improved over time in order to maintain high productivity. Therefore, the ways in which learning can be incorporated into the DT-DSS is another interesting area of future research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The co-authors would like to acknowledge the Knowledge Foundation (KKS), Sweden, for their funding through the research profile Virtual Factories with Knowledge-Driven Optimization at the University of Skövde which is partially related to this work.

References

[1] A. Ingemansson, On reduction of production disturbances in manufacturing systems based on discrete-event simulation, 2004, http://lup.lub.lu.se/record/467483. (Accessed: 27 January 2016).

[2] Z. Yang, Q. Chang, D. Djurdjanovic, J. Ni, J. Lee, Maintenance priority assignment utilizing on-line production information, J. Manuf. Sci. Eng. 129 (2) (2006) 435–446, http://dx.doi.org/10.1115/1.2336257.

[3] J. Li, J, E.D. Blumenfeld, N. Huang, M.J. Alden, Throughput analysis of production systems: recent advances and future topics, Int. J. Prod. Res. 47 (14) (2009) 3823–3851, http://dx.doi.org/10.1080/00207540701829752.

[4] L. Li, L, Q. Chang, J. Ni, G. Xiao, S. Biller, Bottleneck detection of manufacturing systems using data driven method, in: 2007 IEEE International Symposium on Assembly and Manufacturing, 2007, pp. 76–81, http://dx.doi.org/10.1109/ISAM.2007.4288452.

[5] L. Li, Q. Chang, J. Ni, S. Biller, Real time production improvement through bottleneck control, Int. J. Prod. Res. 47 (21) (2009) 6145–6158, http://dx.doi.org/10.1080/00207540802244240.

[6] L. Li, J. Ni, Short-term decision support system for maintenance task prioritization, Int. J. Prod. Econ. 121 (1) (2009) 195–202, http://dx.doi.org/10.1016/j.ijpe.2009.05.006.

[7] A. Azadeh, M. Sheikhalishahi, M. Koushan, An integrated fuzzy DEA–fuzzy simulation approach for optimization of operator allocation with learning effects in multi products CMS, Appl. Math. Model. 37 (24) (2013) 9922–9933, http://dx.doi.org/10.1016/j.apm.2013.05.039.

[8] J. Ni, X. Jin, Decision support systems for effective maintenance operations, CIRP Ann. - Manuf. Technol. 61 (1) (2012) 411–414, http://dx.doi.org/10.1016/j.cirp.2012.03.065.

[9] J.W. Payne, J.R. Bettman, E.J. Johnson, The Adaptive Decision Maker, Cambridge University Press, 1993, http://dx.doi.org/10.1057/jors.1994.133.

[10] K.N. McKay, F.R. Safayeni, J.A. Buzacott, Common sense realities of planning and scheduling in printed circuit board production, Int. J. Prod. Res. 33 (6) (1995) 1587–1603, http://dx.doi.org/10.1080/00207549508930230.

[11] L. Li, Q. Chang, J. Ni, Data driven bottleneck detection of manufacturing systems, Int. J. Prod. Res. 47 (18) (2009) 5019–5036, http://dx.doi.org/10.1080/00207540701881860.

[12] M. Subramaniyan, A. Skoogh, M. Gopalakrishnan, H. Salomonsson, A. Hanna, D. Lämkull, An algorithm for data-driven shifting bottleneck detection, Cogent Eng. 3 (1) (2016) http://dx.doi.org/10.1080/23311916.2016.1239516.

[13] M. Holm, A.C. Garcia, G. Adamson, L. Wang, Adaptive decision support for shop-floor operators in automotive industry, Proc. CIRP 17 (2014) 440–445, http://dx.doi.org/10.1016/j.procir.2014.01.085.

[14] B.L. Song, W.K. Wong, J.T. Fan, S.F. Chan, A recursive operator allocation approach for assembly line-balancing optimization problem with the consideration of operator efficiency, Comput. Ind. Eng. 51 (2006) 585–608, http://dx.doi.org/10.1016/j.cie.2006.05.002.

[15] T. Ertay, D. Ruan, Data envelopment analysis based decision model for optimal operator allocation in cms', European J. Oper. Res. 164 (2005) 800–810, http://dx.doi.org/10.1016/j.ejor.2004.01.038.

[16] L.C. Mak, W.K. Wong, Y.S. Leung, A simulation analysis of the impact of production lot size and its interaction with operator competence on manufacturing system performance, Simul. Model. Pract. Theory 49 (2014) 203–214, http://dx.doi.org/10.1016/j.simpat.2014.09.008.

[17] M. Wedel, M. von Hacht, R. Hieber, J. Metternich, E. Abele, Real-time bottleneck detection and prediction to prioritize fault repair in interlinked production lines, Proc. CIRP 37 (2015) 140–145, http://dx.doi.org/10.1016/j.procir.2015.08.071.

[18] S.R. Lawrence, A.H. Buss, Economic analysis of production bottlenecks, Math. Probl. Eng. Hindawi 1 (4) (1995) 341–363, http://dx.doi.org/10.1155/S1024123X95000202.

[19] C.T. Kuo, J.T. Lim, S.M. Meerkov, Bottlenecks in serial production lines: A system-theoretic approach, Math. Probl. Eng. (1996) http://dx.doi.org/10.1155/S1024123X96000348.

[20] W.J. Hopp, M.L. Spearman, Factory physics. McGraw-hill/Irwin/Irwin, 2008.

[21] J.H. Blackstone, Theory of constraints - A status report, Int. J. Prod. Res. 39 (2001) 1053–1080, http://dx.doi.org/10.1080/00207540010028119.

[22] E.M. Goldratt, J. Cox, The goal: A process of ongoing improvement, 2004, http://dx.doi.org/10.2307/3184217.

[23] J. Adams, E. Balas, D. Zawack, The shifting bottleneck procedure for job shop scheduling, Manag. Sci. 34 (3) (1988) 391–401, http://dx.doi.org/10.1287/mnsc.34.3.391.

[24] C. Roser, M. C, M. Tanaka, Shifting bottleneck detection, in: Winter Simulation Conference 2002, pp. 1079–1086, http://dx.doi.org/10.1109/WSC.2002.1166360.

[25] C. Roser, M. Nakano, M. Tanaka, A practical bottleneck detection method, in: Winter Simulation Conference 2001, pp. 949–953, http://dx.doi.org/10.1109/WSC.2001.977398.

[26] K. Yang, Y. Chung, S.C. Park, Short-term bottleneck detection for process planning in a FAB, Int. J. Comput. Commun. Eng. 3 (6) (2014) 442–445, http://dx.doi.org/10.7763/IJCCE.2014.V3.365.

[27] M. Gopalakrishnan, A. Skoogh, C. Laroque, Simulation-based planning of maintenance activities by a shifting priority method, in: Proceedings of the Winter Simulation Conference 2014. Piscataway, NJ, USA, http://dx.doi.org/10.1109/WSC.2014.7020061.

[28] M. Wedel, P. Noessler, J. Metternich, Development of bottleneck detection methods allowing for an effective fault repair prioritization in machining lines of the automobile industry, Prod. Eng. 10 (3) (2016) 329–336, http://dx.doi.org/10.1007/s11740-016-0672-9.

[29] C. Roser, K. Lorentzen, D. Lenze, J. Deuse, F. Klenner, R. Richter, J. Schmitt, P. Willats, Bottleneck prediction using the active period method in combination with buffer inventories, in: H. Lödding, et al. (Eds.), Advances in Production Management Systems. The Path to Intelligent, Collaborative and Sustainable Manufacturing, Springer International Publishing, Cham, 2009, pp. 374–381.

[30] S.Y. Chiang, C.T. Kuo, S.M. Meerkov, C-bottlenecks in serial production lines: Identification and application, Math. Probl. Eng. (2001) http://dx.doi.org/10.1155/S1024123X01001776.

[31] S. Ching, S, S.M. Meerkov, L. Zhang, Assembly systems with non-exponential machines: Throughput and bottlenecks, Nonlinear Anal. TMA 69 (3) (2008) 911–917, http://dx.doi.org/10.1016/j.na.2008.02.068.

[32] M. Subramaniyan, A. Skoogh, H. Salomonsson, P. Bangalore, M. Gopalakrishnan, Data-driven algorithm for throughput bottleneck analysis of production systems, Prod. Manuf. Res. 6 (1) (2018) 225–246, http://dx.doi.org/10.1080/21693277.2018.1496491.

[33] M. Subramaniyan, A. Skoogh, H. Salomonsson, P. Bangalore, J. Bokrantz, A data-driven algorithm to predict throughput bottlenecks in a production system based on active periods of the machines, Comput. Ind. Eng. 125 (2018) 533–544, http://dx.doi.org/10.1016/j.cie.2018.04.024.

[34] M. Zhang, F. Tao, A.Y.C. Nee, Digital twin enhanced dynamic job-shop scheduling, J. Manuf. Syst. 58 (2021) 146–156, http://dx.doi.org/10.1016/j.jmsy.2020.04.008.

[35] F. Tao, F, H. Zhang, A. Liu, A.Y.C. Nee, Digital twin in industry: State-of-the-art, IEEE Trans. Ind. Inf. 15 (4) (2018) 2405–2415, http://dx.doi.org/10.1109/TII.2018.2873186.

[36] E. Negri, E, L. Fumagalli, M. Macchi, A review of the roles of digital twin in CPS-based production systems, in: Value Based and Intelligent Asset Management, Vol. 29, 2020, pp. 1–307, http://dx.doi.org/10.1016/j.promfg.2017.07.198.

[37] E. VanDerHorn, S. Mahadevan, Digital twin: Generalization, characterization and implementation, Decis. Support Syst. 145 (2021) 113524, http://dx.doi.org/10.1016/j.dss.2021.113524.

[38] J. Voas, P. Mell, V. Piroumian, Considerations for digital twin technology and emerging standards (No. NIST internal or interagency report (NISTIR) 8356 (Draft)), 2021, http://dx.doi.org/10.6028/NIST.IR.8356-draft, National Institute of Standards and Technology.

[39] F. Tao, Q. Qi, L. Wang, A.Y.C. Nee, Digital twins and cyber–physical systems toward smart manufacturing and industry 4.0: Correlation and comparison, Engineering 5 (4) (2019) 653–661, http://dx.doi.org/10.1016/j.eng.2019.01.014.

[40] C. Semeraro, M. Lezoche, H. Panetto, M. Dassisti, Digital twin paradigm: A systematic literature review, Comput. Ind. 130 (2021) 103469, http://dx.doi.org/10.1016/j.compind.2021.1034690166-3615.

[41] H. Jiang, H, T. Qu, M. Wan, L. Tang, G.Q. Huang, Digital-twin-based implementation framework of production service system for highly dynamic production logistics operation, IET Collab. Intell. Manuf. 2 (2) (2020) 74–80, http://dx.doi.org/10.1049/iet-cim.2019.0065.

[42] M. Kunath, H. Winkler, Integrating the digital twin of the manufacturing system into a decision support system for improving the order management process, Proc. CIRP 72 (2018) 225–231, http://dx.doi.org/10.1016/j.procir.2018.03.192.

[43] C.H. dos Santos, J.A.B. Montevechi, J.A. de Queiroz, M.R. de Carvalho, F. Leal, Decision support in productive processes through DES and ABS in the digital twin era: a systematic literature review, Int. J. Prod. Res. (2021) 1–20, http://dx.doi.org/10.1080/00207543.2021.1898691.

[44] M. Grieves, Digital Twin: Manufacturing Excellence Through Virtual Factory Replication, A Whitepaper, 2014, http://dx.doi.org/10.5281/zenodo.1493930.

[45] L. Wright, S. Davidson, How to tell the difference between a model and a digital twin, Adv. Model. Simul. Eng. Sci. 7 (1) (2020) 1–13, http://dx.doi.org/10.1186/s40323-020-00147-4.

[46] W. Kritzinger, M. Karner, G. Traar, J. Henjes, W. Sihn, Digital twin in manufacturing: A categorical literature review and classification, IFAC-PapersOnLine 51 (11) (2018) 1016–1022, http://dx.doi.org/10.1016/j.ifacol.2018.08.474.

[47] F. Tao, F, M. Zhang, A.Y.C. Nee, Background and concept of digital twin, in: Digital Twin Driven Smart Manufacturing, Elsevier, 2019, pp. 3–28, http://dx.doi.org/10.1016/b978-0-12-817630-6.00001-1.

[48] T. Baines, S. Mason, P.O. Siebers, J. Ladbrook, Humans: the missing link in manufacturing simulation? Simul. Model. Pract. Theory 12 (7–8) (2004) 515–526, http://dx.doi.org/10.1016/S1569-190X(03)00094-7.

[49] F. Tao, F, M. Zhang, Y. Liu, A.Y.C. Nee, Digital twin driven prognostics and health management for complex equipment, CIRP Ann. 67 (1) (2018) 169–172, http://dx.doi.org/10.1016/j.cirp.2018.04.055.

[50] J.R. Koza, Genetic Programming: on the Programming of Computers by Means of Natural Selection, Cambridge, MA: MIT Press, 1992.

[51] J.H. Holland, Outline for a logical theory of adaptive systems, J. ACM (1962) http://dx.doi.org/10.1145/321127.321128.

[52] J.H. Holland, Adaptation in Natural and Artificial Systems, Ann Arbor MI University of Michigan Press, 1975, http://dx.doi.org/10.1137/1018105.

[53] M. Laguna, R. Martí, C. Rafael, Scatter Search: Methodology and Implementations, C. Kluwer Academic Publishers, 2003.

[54] I.T. Tanev, T. Uozumi, Y. Morotome, Hybrid evolutionary algorithm-based real-world flexible job shop scheduling problem: application service provider approach, Appl. Soft Comput. 5 (1) (2004) 87–100, http://dx.doi.org/10.1016/j.asoc.2004.03.013.

[55] G. Ochoa, G, J.A. Vázquez-Rodríguezz, S. Petrovic, E. Burke, Dispatching rules for production scheduling: A hyper-heuristic landscape analysis, in: IEEE Congress on Evolutionary Computation, CEC, 2009, http://dx.doi.org/10.1109/CEC.2009.4983169.

[56] M. Frantzén, A.H.C. Ng, P. Moore, A simulation-based scheduling system for real-time optimization and decision-making support, Robot. Comput.-Integr. Manuf. 27 (4) (2011) 696–705, http://dx.doi.org/10.1016/j.rcim.2010.12.006.

[57] R. Poli, W.B. Langdon, N.F. McPhee, A Field Guide To Genetic Programming, Lulu Enterprises, UK Ltd., 2008.

[58] G.Z. Bedny, Application of Systemic-Structural Activity Theory to Design and Training, first ed., CRC Press, 2014.

[59] Plattform industrie 4.0, what is industrie 4.0? 2021, Available at: https://www.plattform-i40.de/I40/Navigation/EN/Industrie40/WhatIsIndustrie40/what-is-industrie40.html. (Accessed: 6 September 2021).