Management and Production Engineering Review
Volume 14 • Number 4 • December 2023 • pp. 56–70
DOI: 10.24425/mper.2023.147204

mper

A Combination of Association Rules and Optimization Model to Solve Scheduling Problems in an Unstable Production Environment

Mateo DEL GALLO, Filippo Emanuele CIARAPICA, Giovanni MAZZUTO, Maurizio BEVILACQUA

Università Politecnica delle Marche, Department of Industrial Engineering and Mathematical Science, Italy

Received: 30 June 2023
Accepted: 01 November 2023

Abstract

Production problems have a significant impact on the on-time delivery of orders, resulting in deviations from planned scenarios. Therefore, it is crucial to predict interruptions during scheduling and to find optimal production sequencing solutions. This paper introduces a self-learning framework that integrates association rules and optimisation techniques to develop a scheduling algorithm capable of learning from past production experiences and anticipating future problems. Association rules identify factors that hinder the production process, while optimisation techniques use mathematical models to optimise the sequence of tasks and minimise execution time. In addition, association rules establish correlations between production parameters and success rates, allowing corrective factors for production quantity to be calculated based on confidence values and success rates. The proposed solution demonstrates robustness and flexibility, providing efficient solutions for Flow-Shop and Job-Shop scheduling problems with reduced calculation times. The article includes two Flow-Shop and Job-Shop examples where the framework is applied.

Keywords

Data mining; Association rules; Optimization model; Production scheduling; Job-shop scheduling; Flow-shop scheduling.

Introduction

Machine breakdowns or other types of problems in the production system can generate a high percentage non-conforming parts realized. Such events may lead to a part having to be reworked or discarded, scenarios that in any case cause a delay in production compared to the planned schedule. To increase their competitiveness, companies must be capable of responding fast to several factors that can compromise the production process. In this context, it is important that the scheduling phase can predict such anomalies to avoid errors in the scheduled dates.

For this reason, Data Mining (DM) approaches have been used over the years to create scheduling algorithms that are more flexible because they are able to consider different aspects when choosing the sequence of activities. In particular, Association Rules (ARs) are a powerful tool to support decision-making processes, because they are able to find hidden relationships between different parameters in large datasets (Troncoso-García et al., 2023).

In this context, the present work aims to develop a self-learning framework that combines ARs and optimization models to realize a scheduling algorithm capable of considering the probability of non-conforming items based on several factors. ARs are used to find correlations between different combinations of production aspects (like the type of article, type of material or number of parts to be manufactured) and the likelihood of good parts realized present in historical datasets. Depending on the number and type of parameters chosen to extract ARs, different levels of ARs were defined with different levels of accuracy. From the results of ARs, the list of tasks to be scheduled is compared with them to find one or more rules that describe one or more production processes. If there is a match between the task to be scheduled and the ARs, a corrective factor on the quantity to produce based on the consequent value and the success rate is calculated. From the ARs re-

***Corresponding author**: Mateo Del Gallo – Università Politecnica delle Marche, Department of Industrial Engineering and Mathematical Science, Italy – Ancona Via Brecce Bianche 12, 60131, phone: +39 3291016745, e-mail: m.delgallo@pm.univpm.it*

sults, it is possible to understand if a production process with certain characteristics had a higher or lower success rate based on historical data. Thus, the quantities that will be produced are higher than the actual demands in order to anticipate future failures and respect the delivery date. To solve the scheduling problem, a mathematical model was developed with the aim of minimizing the makespan value. The proposed scheduling algorithm turns out to be flexible for both Flow-Shop Scheduling Problems (FSSP) and Job-Shop Scheduling Problems (JSSP). FSSP and JSSP are known as NP-hard problems (Babor et al., 2023), are one of the most difficult combinatorial optimization problems. It was considered limited availability of resources for processing and each machine has its own matrix setup time.

In the existing literature, numerous scheduling models integrate Artificial Intelligence (AI) techniques or a DM approach with heuristic methods to tackle scheduling problems (Zhang et al., 2022). However, there is a paucity of research employing mathematical models and DM techniques for this purpose. To fill this gap, the scheduling framework proposed in this study aims to introduce a novel decision-making tool that utilizes a data-driven approach and an optimisation model. The aim is to achieve a globally optimal solution to the problem of task sequencing in an unpredictable production environment.

The manuscript is structured as follows: in Literature Review there is a literature review about the use of ARs in scheduling problems. Research Approach presents the research approach and the framework. Framework Application presents the application of the framework. The results and discussion are reported in section Results and discussion. Finally, conclusions of the work are reported in Conclusions.

Literature review

In recent years, the research focus on scheduling problems has significantly increased with the emergence of Industry 4.0. Various methods have been proposed to address the challenges posed by JSSP or FSSP. However, the use of DM approaches combined with the use of mathematical models to solve the sequencing problem in this field is relatively limited. To collect the contributions in the literature on the use of ARs or DM techniques for solving scheduling problems, a systematic literature review was conducted. The renowned scientific database Scopus was selected, considering articles written in English and having full-text availability. To stay up to date with the latest advancements, only articles published between 2019 and 2023 were included in the study. Each article was thoroughly reviewed to assess its relevance and suitability to the established theme of this research. The total number of papers retrieved from Scopus, along with the number of papers selected for the literature study, are presented in Table 1. The literature review was conducted by searching for keyword pairs that would lead back to the topic of the article. Specifically, "Scheduling", "Flow-shop scheduling" and "Job-shop scheduling" were chosen as the first keywords in order to collect publications that deal with the problems discussed in this article. Meanwhile, for the second keyword, "Association Rules" and "Data Mining" were chosen in order to study how other authors have used the same techniques as us to solve scheduling problems.

Table 1
Selected papers for literature contributions

Keywords	Papers found	Relevant papers
"Scheduling" AND "Association Rules"	90	7
"Job shop scheduling" AND "Association Rules"	2	1
"Flow shop scheduling" AND "Association Rules"	2	0
"Job shop scheduling" AND "Data Mining"	17	4
"Flow shop scheduling" AND "Data Mining"	7	0

An interesting approach to the use of ARs is that proposed by (Wu et al., 2018) which focuses on an assembly resource planning strategy that uses the Apriori algorithm to mine historical assembly resources and labour hour data. Similarly, (Farizal & Joelian, 2020) also applied the Apriori algorithm to determine the optimal engine replacement timing using DM techniques on heavy equipment engine condition monitoring data and external parameters. They used clustering to categorise monitoring data, ARs to analyse interactions between variables and time series analysis to estimate the usefulness of condition monitoring.

Nasiri et al. (2019) presents a DM approach that combines 'attribute-oriented induction (AOI) and association rules (AR)' techniques to generate an improved initial population for population-based metaheuristics in JSSP resolution. The authors used AOI to learn rules and knowledge types and the concept tree obtained from AOI was used as input for ARs. The ARs were applied to a dataset of optimal or near-optimal solutions of JSSP instances to identify associations between the attribute class of an operation and its sequence in the solutions. The authors evaluated the effectiveness of the proposed procedure by using the solutions as the initial population for the Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) and obtained significant performance improvements.

Qiu et al. (2019) conducted an experiment on a data mining-based disturbance prediction system for the workshop schedule. They employed a hybrid algorithm in the DM module to generate a disturbance tree, which acted as a classifier of disturbances occurring prior to production. The disturbance tree was then used to classify disturbances and based on the results of the decision-making process; scheduling was planned in advance to avoid disruptions. Instead, Zhao et al. (2022) proposed a reactive scheduling method to manage the uncertainty of the arrival of new jobs. They used makespan and machine utilisation as scheduling criteria and divided the production system period into several sub-criteria. The dynamic scheduling model assigned dispatching rules to sub-scheduling periods in real-time.

Habib Zahmani & Atmani (2021) presented an approach combining DM, GA and simulation techniques for the JSSP. Data mining was used to identify the best dispatching rules through a decision tree, which served as a database for the GA. The GA then created and evolved the dispatching rule sets and assigned them to the machines on the shop floor. A simulator replicated the shop floor environment collected the data, filled a database with job attributes, and evaluated the rule sets based on the relevant time interval. Chen (2019) proposed a resource allocation algorithm based on big data association mining in the context of cloud computing. This technique extracted resource ARs from the information management system and employed adaptive optimal resource allocation control using ARs as training sets. On the other hand, Wang et al. (2022) explored the use of ARs in manufacturing systems and applied logic to address production rule problems between production lines and goods in the automotive industry.

Analysis of the publications shows that the use of ARs in combination with optimization algorithms is present in the literature. However, there are no cases of self-learning frameworks that combine the use of ARs with mathematical models for solving scheduling problems. This paper aims to fill this gap by proposing a self-learning framework for solving JSSP and FSSP.

Research approach

The algorithm was created in an attempt to find the global minimum value of makespan in scheduling production orders. However, the processing time often does not correspond to what is expected due to errors made during the production phase. Consequently, it was necessary to use DM techniques such as ARs. Based on historical data, ARs are used to identify the relationships between the characteristics that define the production process and its success rate. Fig 1. shows a diagram of the working process of the proposed framework.

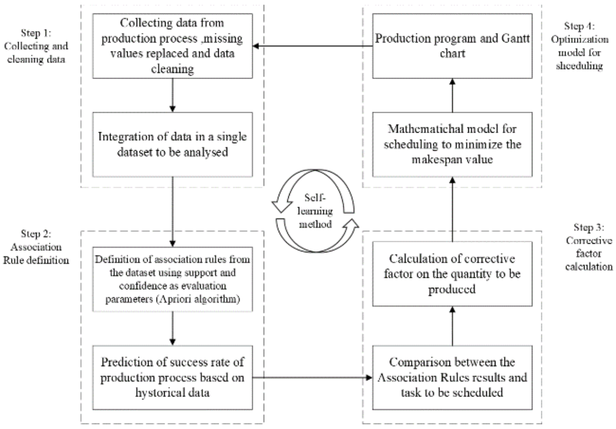
Step 1:
Collecting and cleaning data
Collecting data from production process ,missing values replaced and data cleaning
Integration of data in a single dataset to be analysed
Production program and Gantt chart
Step 4:
Optimization model for sheeduling
Mathematichal model for scheduling to minimize the makespan value
Self-learning method
Step 2:
Association Rule definition
Definition of association rules from the dataset using support and confidence as evaluation parameters (Apriori algorithm)
Prediction of success rate of production process based on hystorical data
Step 3:
Corrective factor calculation
Calculation of corrective factor on the quantity to be produced
Comparison between the Association Rules results and task to be scheduled

Fig. 1. Self-learning framework for scheduling program

A preliminary phase of the framework consists of collecting and cleaning the data of the company under investigation. This phase determines the overall quality of the path. Indeed, any irregularities overlooked at this point can have an impact on the subsequent procedure. It is necessary to ensure the accuracy of the collections, combine the various sources and correct missing values. This process leads to a single dataset incorporating all relevant sources. The dataset contains information on the percentage of successfully processed parts for a given production process.

Application of ARs is crucial in order to find a correlation between the percentage of correctly executed processes and different combinations of factors that may be the type of material, the type of article and the quantity to be produced. ARs return different rules which are defined according to the number of parameters considered as antecedents. In “Association rules definition” this step of the framework will be more accurately explained.

The sequence of operations to be programmed is compared with the ARs results. If there is a match with one or a set of rules, a correction factor is calculated on the quantities to be produced to avoid subsequent rework and to predict possible anomalies during the production process ("Corrective factor calculation").

The last step of the framework is the optimization model for scheduling problems; it is a mathematical model with the aim of finding the global optimum solution that minimises the makespan. ("Optimization Model"). From the outcome of the processing that has been carried out, the dataset can be enriched with new data from the production process in order to make the framework more accurate and create a self-learning method.

The following paragraphs explain the various stages of the algorithm in detail.

Association rules definition

ARs are a type of DM technique used to discover relationships between variables or elements in a dataset. Specifically, ARs analyse transactions in a dataset to discover patterns of co-occurrence between two main elements, antecedent and consequent. The consequent refers to a distinct group of items or attributes that frequently occur together with the antecedent. The antecedent, on the other hand, represents a set of one or more items or attributes that co-occur in a process. To calculate ARs in a dataset, the first step involves identifying frequent itemsets, which are collections of one or more items that appear together. To evaluate the quality of an AR, in this work, support and confidence are utilized (Fani et al., 2023). For an AR of the form $x \to y$ where x is the antecedent and y the consequent, the support $\mathit{Supp}(x \to y)$ is calculated as the number of transactions containing both x and y, divided by the total number of transactions. Support represents the probability of encountering both x and y in a transaction.

For the same AR, the confidence $\mathit{Conf}(x \to y)$ is computed as the support of the rule $\mathit{Supp}(x \to y)$ divided by the support of the antecedent ($\mathit{Supp}(x)$). Confidence indicates the likelihood of finding item y in a transaction that already contains item x. The Apriori algorithm, developed by (Agrawal & Srikant (1994), is an early and widely used algorithm for discovering ARs and has gained significant popularity in the DM community. This algorithm operates on the concept of frequent itemsets, which are sets of items (such as objects, products, and words) that occur together with a certain frequency within a given dataset.

To effectively utilize the Apriori algorithm, it is necessary to define two parameters: minimum support and minimum confidence. These parameters serve as thresholds to determine the significance of itemsets and rules. The algorithm employs a pruning strategy, where infrequent itemsets are removed in order to reduce computational complexity and enhance performance. By eliminating infrequent itemsets, the algorithm focuses on the most relevant and significant associations within the dataset.

Based on the data sources available to construct the case study, specific objectives for the analysis may be developed, and the ARs can then be retrieved to solve the specific issue. For instance, one can focus on drawing connections between the outcomes of specific workflows and the emergence of problems throughout the manufacturing stage. The outcomes from this phase can be used to plan certain corrective actions to improve the process flow, its reliability, and, as a result, the timely delivery of the product.

In the proposed framework, ARs were used to discover hidden relationships between several factors of the production process and the success rate. A production process can be characterized by considering different aspects, which can be divided into two groups:

- Constant parameters: These are attributes that remain unchanged from one scheduling program to the next one. They provide a consistent basis for the production process.
- Variable parameters: These are parameters that may vary from one machining operation to the next. They capture the dynamic nature of the production process and reflect the changes that occur.

Based on these parameters, groups of ARs can be formed. Each group consists of an antecedent, which includes a combination of constant and variable parameters, and a consequent, which represents the associated outcome or behaviour like the percentage of good parts realized. The number of parameters considered in the antecedent determines the level of the ARs.

Is possible to define different Association Rule Levels (ARL) based on the number of parameters involved:

- Association Rules Level 1 (ARL1): This level includes rules that have $m + n$ parameters as the antecedent, where m represents the constant parameters and n represents the variable parameters.
- Association Rules Level 2 (ARL2) includes rules that have $m+(n-1)$ parameters as the antecedent. One variable parameter is excluded compared to ARL1.
- . . .
- Association Rules Level n (ARLn): This level includes rules that have $m + 1$ parameter as the antecedent. Only one variable parameter is considered as antecedent.

The classification of ARs at different level help to find more rules which describe the process with different level of accuracy. The more parameters that are considered as antecedents, the better that process will be described.

Each ARL can contain multiple groups of rules, depending on the possible combinations of variable parameters. These levels provide a structured framework for analysing and understanding the relationships and patterns within the production process at different levels of detail. There will be d_l possible ARs for a specific level depending on the combination of parameters you want to choose as antecedents. d_l is the binomial coefficient (1) (Jiménez-Pastor & Petkovšek, 2023) and l is the ARs level being considered.

$$d_l = \binom{n}{k} = \frac{n!}{(n-k)! * m!}, \qquad l \in [1,\, n] \tag{1}$$

$$k = n - l + 1 \tag{2}$$

$$\text{No of combinations} = \sum_{l=1}^{n} d_l \tag{3}$$

The total number of possible groups of ARs is expressed by Equation (3). In this way is possible to have several groups of ARs with different numbers of parameters so is easier to find one or a group of rules that can describe the production process. The authors propose an example of a dataset in Table 2 where $m1$ and $m2$ are two constant parameters that describe the production process and $i1, i2$, and $i3$ are variable parameters; the last column is the consequent c.

Table 2
Example of the dataset for the extraction of ARs

$m1$	$m2$	$i1$	$i2$	$i3$	c
a	b	$x1$	$y1$	$z1$	$c1$
a	b	$x1$	$y2$	$z1$	$c2$
a	b	$x2$	$y1$	$z2$	$c3$
a	b	$x2$	$y2$	$z1$	$c1$
a	b	$x1$	$y2$	$z2$	$c3$
a	b	$x3$	$y1$	$z1$	$c1$
a	b	$x1$	$y1$	$z2$	$c3$
a	b	$x2$	$y1$	$z1$	$c2$
a	b	$x2$	$y2$	$z1$	$c3$
a	b	$x1$	$y1$	$z2$	$c1$

The difference between the several ARL is how the antecedent is constructed. An example of different ARL of the data presented in Table 2 is shown in Fig. 2. At the highest level, the antecedent is composed of all the five parameters that describe the process (constants + variables). Meanwhile, the ARL2 have as antecedent only four parameters but at this level, there are three possible combinations of ARs, it depends on the combination of variables parameters. At least, the ARL3 have only one variable parameter as an antecedent that describes the process, in this level, there are three possible combinations too. It can be seen from Equation (3) that there are seven possible combinations on which to calculate ARs.

Corrective factor calculation

The results of ARs provide valuable insights into the probability of success for each production step based on historical data. This information is crucial in the decision-making process, as it allows for informed choices regarding production planning and scheduling.

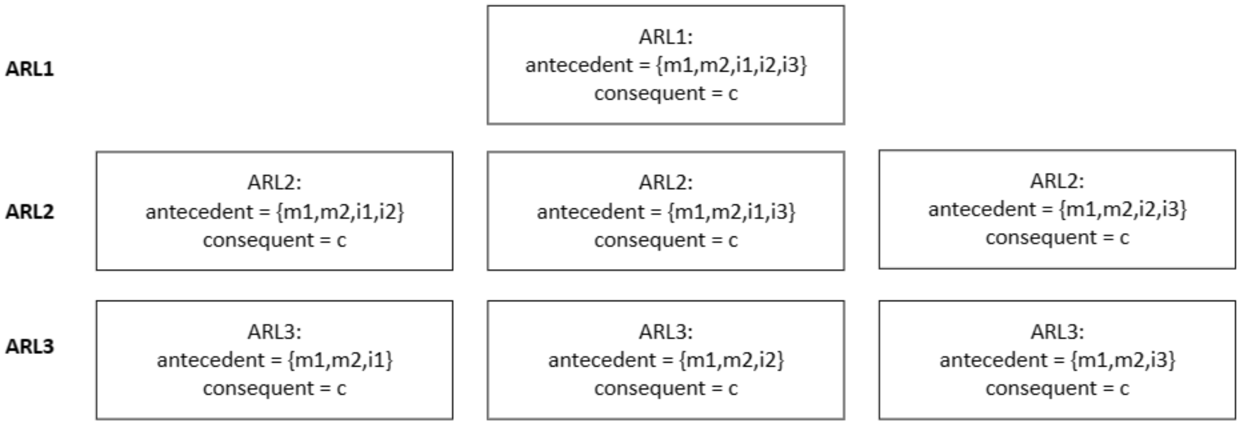

Fig. 2. Example of different ARL

By analysing the ARs, it becomes possible to assess the success rate associated with each operation or step in the production process. If the success rate is found to be low for a particular step, it indicates a higher likelihood of interruptions or failures during that stage. In such cases, it may be advantageous to adjust the production strategy by producing a larger quantity to mitigate potential disruptions in the production chain.

The next step in the framework involves comparing the results of ARs with the sequence of operations to be scheduled. This comparison helps to make more informed decisions on the sequence of tasks and to optimise the overall production schedule.

Figure 3. shows a flow chart which explains the approach used to compare the activities that must be scheduled and the results of ARs.

The first step is trying to find a set of rules in ARL1 that describe one or more operations to be scheduled. The first research must be conducted on the highest level of ARs results previously calculated because ARL1 is considering as antecedent all the variable parameters so rules present in this level characterize better the process. If there is a match means that there has been one or more processes with the same characteristics in the historical data so is possible to see the success rate and that operation will not be searched for in the lower levels. If no correlations are found on the higher level, one will look for them in one of the possible results of the lower level (ARL2). This process iterates up to the last calculated level of ARs.

Once a correlation has been found between the operations to be scheduled and a set of rules, a correction factor is calculated based on the confidence value and the value of the consequent.

If there are no correlations, it means that there is no information about the production process in historical data.

$$\text{No of part to be manufactured} = n_i + \sum_j n_i * (sc_{ij}) * conf_j \qquad (4)$$

Equation (4) is used to calculate the correction factor that must be added to the theoretical demand of the part to be manufactured. The terms of the equation are:

- n_i represents the units of item i to be realised (customer demand).
- sc_i represents the percentage of non-conforming parts (1 – consequent).
- $conf_i$ represents the confidence of the rule article $i \rightarrow$ Succes Rate.

This correction factor helps prevent under-dimensioning during production. For convenience, if the resulting correction factor is not a whole number, it is rounded up to the nearest whole number.

The sequence of steps described earlier can be applied to all operations that need to be scheduled. However, it is advisable to prioritize its application to the most critical production processes. By focusing on critical processes, excessive and unnecessary calculation times can be avoided improving overall efficiency. By incorporating the correction factor into customer demands and considering the specific needs of each production process, more accurate and reliable scheduling can be achieved. This approach helps in reducing the risk of under-dimensioning and assures a smooth and efficient production workflow.

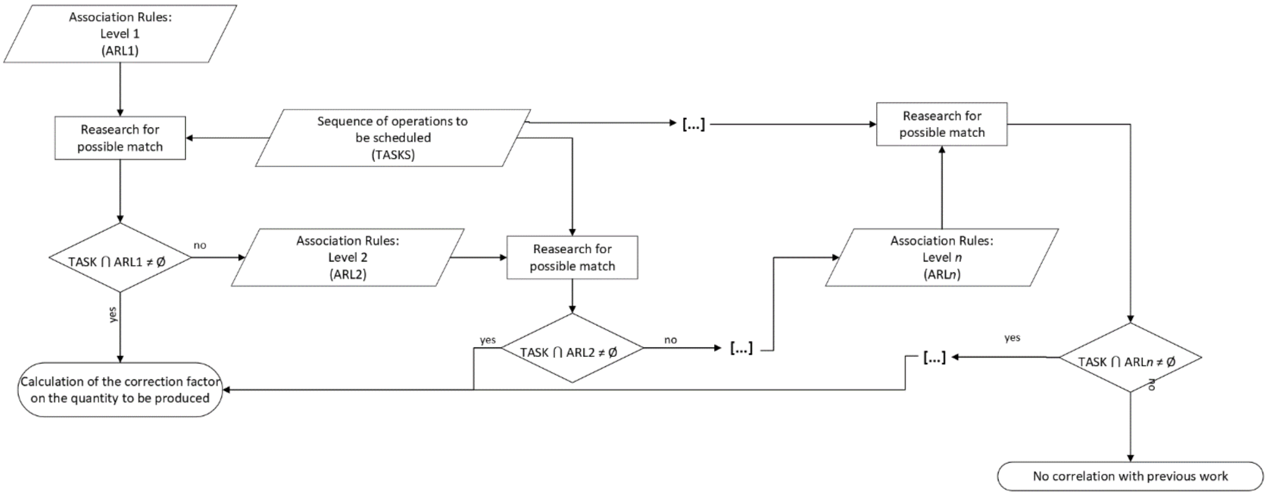
Association Rules:
Level 1
(ARL1)
Reasearch for possible match
Sequence of operations to be scheduled
(TASKS)
[…]
Reasearch for possible match
TASK ∩ ARL1 ≠ Ø
no
Association Rules:
Level 2
(ARL2)
Reasearch for possible match
Association Rules:
Level n
(ARLn)
yes
yes
TASK ∩ ARL2 ≠ Ø
no
[…]
[…]
yes
TASK ∩ ARLn ≠ Ø
no
Calculation of the correction factor on the quantity to be produced
No correlation with previous work

Fig. 3. Research of a match between ARs results and sequence of operation to be scheduled

Optimization model

This section presents in detail the mathematical model that implements the Scheduling Problem (SP) to find the optimal solution for resource allocation. The proposed model is applicable for JSSP and FSSP with the assumption that a machine can only process one article at a time and an article can only be processed by one machine at a given time instant. There is no constraint on the availability of material and human resources at instant zero.

A typical SP is described by the allocation of a set of jobs J to a group of machines M and the various article to be manufactured can be composed by a set of different materials O.

- $j \in J$, the elements and set of jobs.
- $s \in S$, the elements and set of jobs that have already started but not yet completed at the time of scheduling.
- $m \in M$, the elements and set of machines.
- $o \in O$, the elements and set of materials.
- $(i, n) \in [1, len(J)]$ indices for range of set J.
- $(k, v) \in [1, len(M)]$ indices for range of set M.
- $(y, z) \in [1, len(O)]$ indices for a range of set O.

A single task to be scheduled can be characterized by the triad (j, m, o): product j with material o processed on the machine m.

Table 3 shows how the information of the task (j, m, o) is structured: for each task is reported information about the duration for single part production, any precedence constraints, customer demand and the percentage of completion. This last column is necessary because, at the instant of the schedule, it is possible that activities are already in progress.

It is important to know the set-up time necessary to pass from the production of j_i with material o_y to article j_n with material o_z in machine m_k for every (j, m, o) to be scheduled. The group of all set-up times is indicated with:

- $t_{j_i,m_k,o_y,j_n,o_z} \in T$, elements and set of set-up times for machine m_k.

Every task to be scheduled is characterized by two decision variables:

- $dur_{j,m,o}$ = duration of task j-th on machine m-th with o-th material.
- $start_{j,m,o}$ = time to start of task j-th on machine m-th with o-th material.

The proposed mathematical model for solving the scheduling problem is illustrated below.

The aim of this optimization problem is to minimize the makespan value (Equation (5)). The makespan is a metric used in task scheduling and planning problems and it indicates the total amount of time required to complete all tasks in a work schedule or production system. The proposed model has different constraints (Equations (6)–(10)).

$$\begin{aligned}&\mathit{minimize(makespan)}\\&\quad = \mathit{minimize}(\max)\,\forall\, z\{\in R|\\&\quad z = start_{j,m,o} + dur_{j,m,o}, \forall(j, m, o)\end{aligned} \tag{5}$$

$$start_{j,m,o} \geq 0 \tag{6}$$

$$dur_{j,m,o} \geq 0 \tag{7}$$

$$start_{j,m,o} + dur_{j,m,o} \leq makespan \tag{8}$$

$$\begin{aligned}&start_{j_i,m_k,o_y} + dur_{j_i,m_k,o_y} \leq start_{j_i,m_v,o_y};\\&\text{for} \quad \left(x_{j_i,m_k,o_y}\right) = prec(x_{j_i,m_v,o_y})\end{aligned} \tag{9}$$

$$start_{s,m,o_y} + dur_{s,m,o_y} \leq start_{j,m,o} \tag{10}$$

$$\begin{aligned}&[start_{j_i,m_k,o_y} + dur_{j_i,m_k,o_y} +\\&t_{j_i,m_k,o_y,j_n,o_y} \leq start_{j_n,m_k,o_z}]\\&\vee\\&[start_{j_n,m_k,o_z} + dur_{j_n,m_k,o_z} +\\&t_{j_n,m_k,o_z,j_i,o_z} \leq start_{j_i,m_k,o_y}]\end{aligned} \tag{11}$$

Equations (6) and (7) are used to specify that the duration and start time must be non-negative numbers and (8) requires that the end time of a single activity must be equal or less than makespan. The constraint Equation (9) says that an activity cannot begin if the activity preceding it has not been completed; it is an equation that refers to those activities subject to precedence constraints with other activities. Meanwhile, Equation (10) serves to give precedence to tasks that have a higher percentage of completion than other tasks to be performed on the m-th machine, this is because it has been assumed that a task cannot be stopped if it has already begun.

Table 3
Example of a task structure for scheduling

Task (job, machine, material)	Dur	Prerequisite Task	Demand	% Complete
(‘Job1’, ‘$M1$’, ‘material1’)	10	None	4	100
(‘Job1’, ‘$M2$’, ‘material1’)	12	(‘Job1’, ‘$M1$’, ‘material1’)	4	20
(‘Job1’, ‘$M3$’, ‘material1’)	3	(‘Job1’, ‘$M2$’, ‘material1’)	4	0

A disjunction equation is a type of equation that contains at least two expressions separated by the word 'or'. The objective of solving a disjunction equation is to find all values that satisfy at least one of the two equations separated by the word 'or'. Equation (11) describes the behaviour whereby if a machine is engaged in a machining operation, it cannot start an activity on another item until the previous activity has been completed. Thanks to this expression there is no possible overlapping of the activity in the scheduling program.

Regarding the availability of machines is necessary to introduce a machine availability calendar in which it is stated day by day whether the day is working or not and at what times the machines are available as shown in Table 4. The last column is a binary number indicating whether the day is a holiday with 0 and whether the day is a working day with 1.

Table 4
Machine availability calendar

Month	Day	H_start	min_start	min_available	Work_day
5	1	6	0	0	0
5	2	6	0	960	1
5	3	6	0	960	1
...	...	...	...	...	...
5	31	6	0	960	1

Framework application

The proposed framework was tested in different scenarios, to give the reader a clear understanding of the algorithm and an idea of its robustness, two examples are given. The first refers to a classic job shop scheduling problem while the second to a flow-shop scheduling problem. In order to compare better the two proposed scenarios, a configuration with 5 articles that must be manufactured on 5 machines is presented. Next, the results in terms of calculation times will be shown for different scenarios in which various factors are varied, such as the number of machines, the number of articles to be processed, the number of operations to be scheduled, the number of correlations found between the activities to be planned and the calculated ARs. The list of tasks to be scheduled is shown in Table 5 and the scheme of the JSSP is illustrated in Fig. 4.

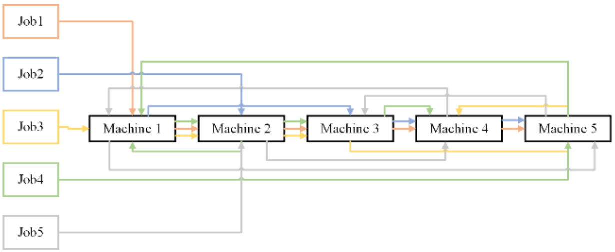

Fig. 4. Job-Shop scheme

Job-shop scheduling problem

The proposed JSSP, a classic NP-hard problem, is shown in Fig. 4. Part of the list of tasks to be scheduled is shown in Table 5.

The proposed JSSP consists of 5 articles to be produced on 5 machines with 4 different types of materials.

The next step is to search if similar processes exist in the historical data. For this purpose, ARs were used and to optimise the calculation time of the algorithm, a good strategy is to search for past production anomalies only for the critical process. This strategy thus makes it possible to avoid increasing the calculation time by analysing established processes that do not generate faults. In this case, in order to make the

Table 5
Tasks to schedule in the Job-Shop plant

ID	Task (job, machine, material)	Dur	Prerequisite task	Demand	% complete
1	(Job1, Machine1, Material 1)	10	None	10	5
2	(Job1, Machine2, Material 1)	10	(Job1, Machine1, Material 1)	10	0
...	...	...	...	...	...
24	(Job5, Machine5, Material 2)	15	(Job5, Machine1, Material 2)	3	0
25	(Job5, Machine3, Material 2)	8	(Job5, Machine5, Material 2)	3	0

example clearer, Machine 2 has been assumed to be the most critical of the path, and therefore the extraction of ARs will only be done for this process. If there are several critical processes within a production process, the extraction of ARs will also be done for the other processes.

Flow-shop scheduling problem

The FSSP proposed is illustrated in Fig. 5. In this case, all 5 items to be scheduled must be processed on the same 5 machines in the same order. This last aspect is the difference between FSSP and JSSP.

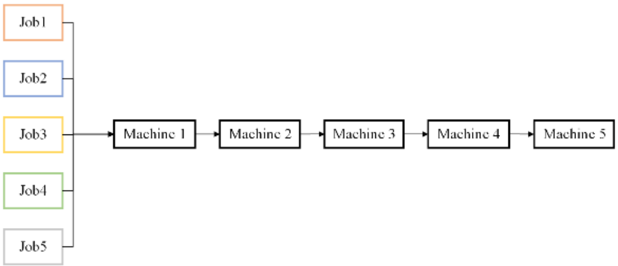

Fig. 5. Flow-Shop scheme

In order to be able to compare the calculation times of the algorithm in the case of FSSP and JSSP, the same number of articles (5), the same number of machines (5) and the same number of available materials (4) were used and the manufacturing time for each process is the same for both the problems. Customer demand and unit production themes were left unchanged in both cases. Table 6 reports part of the tasks to be scheduled.

For both cases, the set-up time matrices for the five machines are the same and the Machine 2 process was considered as the most critical in the path.

Association rules application

In these cases, the ARs are calculated to find the correlation between the characteristic of the Machine 2 operation and its success rate. Five parameters were considered to describe the process, divided as follows:

- Type of process (Machine 2) and period of the year (May) were assumed as constant parameters.
- Type of article, type of material and number of parts to be manufactured as variable parameters.

According to Equation (3) is possible to consider seven combinations of parameters that can describe the Machine 2 process divided into three levels of accuracy. Fig. 6 shows the difference between the antecedents chosen for the different levels of ARs.

In the present case study ARL1 is defined as the set of rules that describe the Machine 2 process with all the five parameters (type of process, period of the year, type of article, type of material and number of parts to be manufactured) as antecedent and the success rate as consequent.

ARL2 is composed of three groups of ARs that have different combinations of parameters as antecedents:

- ARL2_1: have as antecedent the type of process (Machine 2 process), month (May), type of article and type of material.
- ARL2_2: have as antecedent the type of process (Machine 2 process), month (May), type of article and number of parts to be manufactured.
- ARL3_3: have as antecedent the type of process (Machine 2 process), month (May), type of material and number of parts to be manufactured.

In the same way, ARL3 have three groups of ARs too characterized in the following way:

- ARL3_1: have as antecedent the type of process (Machine 2 process), month (May) and type of article.
- ARL3_2: have as antecedent the type of process (Machine 2 process), month (May) and type of material.
- ARL3_3: have as antecedent the type of process (Machine 2 process), month (May) and the number of parts to be manufactured.

To extract the ARs from the dataset was used Mlxtend library (Raschka, 2018), an open-source library used in Python 3.8.10.

Table 6
Tasks to schedule in the Flow-Shop plant

ID	Task (job, machine, material)	Dur	Prerequisite task	Demand	% complete
1	(Job1, Machine1, Material 1)	10	None	10	5
2	(Job1, Machine2, Material 1)	10	(Job1, Machine1, Material 1)	10	0
. . .	. . .	. . .	. . .	. . .	. . .
24	(Job5, Machine4, Material 2)	4	(Job5, Machine3, Material 2)	3	0
25	(Job5, Machine5, Material 2)	15	(Job5, Machine4, Material 2)	3	0

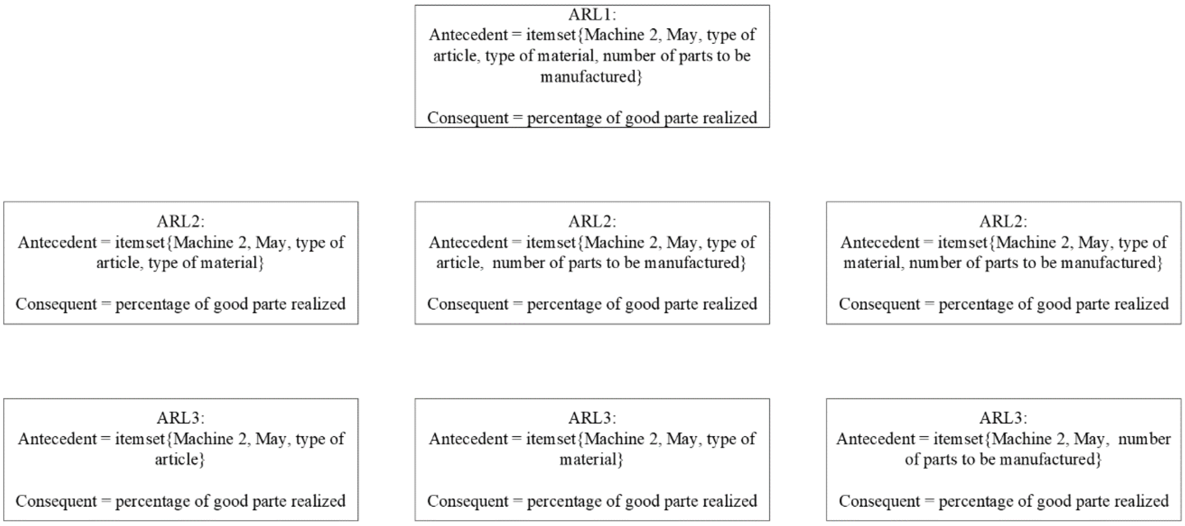
ARL1:
Antecedent = itemset{Machine 2, May, type of article, type of material, number of parts to be manufactured}
Consequent = percentage of good parte realized
ARL2:
Antecedent = itemset{Machine 2, May, type of article, type of material}
Consequent = percentage of good parte realized
ARL2:
Antecedent = itemset{Machine 2, May, type of article, number of parts to be manufactured}
Consequent = percentage of good parte realized
ARL2:
Antecedent = itemset{Machine 2, May, type of material, number of parts to be manufactured}
Consequent = percentage of good parte realized
ARL3:
Antecedent = itemset{Machine 2, May, type of article}
Consequent = percentage of good parte realized
ARL3:
Antecedent = itemset{Machine 2, May, type of material}
Consequent = percentage of good parte realized
ARL3:
Antecedent = itemset{Machine 2, May, number of parts to be manufactured}
Consequent = percentage of good parte realized

Fig. 6. Structure of antecedents and consequents of different ARL

For each level, the first step is to find the most frequent sets in the dataset. The minimum support and minimum confidence threshold is set to min_support = 0.005 and min_confidence = 0.000. These two values are set low to try to find all possible rules to better identify the process, as the size of the dataset is not large enough. It is of interest to be able to appropriately define time intervals for updating the database. In this way, it will also be possible to define when it is appropriate to update the new ARs. In fact, since these are data-driven processes, it is essential that the results of the analyses are up to date, with a view to being able to adapt production as promptly as possible. Given that there is no time interval defined by the state of the art, it makes sense to propose a time interval of at least one month between successive updates. This allows the company time to assimilate the new suggestions provided by the data-driven analysis and, at the same time, does not leave too wide an interval that would compromise the validity of the results themselves.

Part of the results of ARs for the seven levels presented are shown in Table 7. The total number of rules found, based on available historical data, is 528 divided into the three accuracy levels of ARs.

For example, line 502 of Table 7 describes the Machine 2 process in May with only one article to be produced and the 100% of good parts realized have a support of 0,31. That means that this pair of events is present in 31% of the transaction. Confidence value explains that with 85% the Machine 2 process described in line 502 has a success rate of 100%. Line 501 presents the same antecedent but a different consequent. Rule 501-st has a support value of 0,074 so is less frequent than the rule in lane 502. Confident value tells that the 15% of probability that the process describes in lane 501 has a 0% of success rate.

Corrective factor calculation

From the ARs results, it is possible to understand the percentage of good parts produced for a specific working condition of the Machine 2 process with a given confidence value. To avoid having to carry out rework that would lead to an excessive increase in time to market, a correction factor on the quantity to be produced was calculated. The first step is to try to identify a set of ARs that can describe the tasks that must be scheduled.

The list of the tasks which must be scheduled (Table 5 and Table 6) and the results of ARL1 were compared in order to find one or more rules that characterize a specific Machine 2 process. If there is a match, a corrective factor on the quantity to be produced will be calculated according to the consequent and confidence values. In the event that there are no rules in ARL1 with the same antecedent compared to the parameters of a specific Machine 2 process that must be scheduled, other rules will be searched for in one of the three groups forming ARL2. Again, if there is a match between a task to be scheduled and rules present in one of the ARL2 groups, the corrective factor will be calculated, otherwise, the search for potential rules will be carried out in the lower level, ARL3. If there are no matches in ARL3 either, it will mean that there is no historical record of the Machine 2 process with the selected parameters.

Table 7
Part of results from each ARL

	Level	Antecedent	Consequent	Support	Confidence
1	ARL1	Machine2/May/Product 1/Material 1/NP5'	0	0.006369	1
2	ARL1	Machine2/May/Product 4/Material 1/NP1'	100	0.003299	0.333333
3	ARL1	Machine2/May/Product 5/Material 3/NP1'	100	0.019849	0.25
…	…	…	…	…	…
155	ARL2_1	Machine2/May/Product 1/Material 4'	25	0.0069427	0.333333
156	ARL2_1	Machine2/May/Product 5/Material 3'	33	0.0029325	0.5
157	ARL2_1	Machine2/May/Product 4/Material 1'	100	0.0093253	0.8
…	…	…	…	…	…
220	ARL2_2	Machine2/May/Product 1/NP5'	60	0.0025161	0.666667
221	ARL2_2	Machine2/May/Product 2/NP3'	67	0.019563	0.75
222	ARL2_2	Machine2/May/Product 2/NP8'	100	0.0052141	1
…	…	…	…	…	…
289	ARL2_3	Machine2/May/Material 1/NP3'	33	0.0295612	1
290	ARL2_3	Machine2/May/Material 3/NP6'	83	0.0094215	0.666667
291	ARL2_3	Machine2/May/Material 4/NP1'	0	0.0076124	0.333333
…	…	…	…	…	…
390	ARL3_1	Machine2/May/Product 2'	50	0.031251	0.235294
391	ARL3_1	Machine2/May/Product 2'	25	0.0041252	0.058824
392	ARL3_1	Machine2/May/Product 1'	83	0.012523	0.25
…	…	…	…	…	…
452	ARL3_2	Machine2/May/Material 1'	100	0.0195251	0.05
453	ARL3_2	Machine2/May/Material 3'	75	0.0096122	0.235294
454	ARL3_2	Machine2/May/Material 4'	100	0.0305215	0.471321
…	…	…	…	…	…
501	ARL3_3	'Machine2/May/NP1'	0	0.07432	0.15
502	ARL3_3	'Machine2/May/NP1'	100	0.31273	0.85
…	…	…	…	…	…
528	ARL3_3	'Machine2/May/NP3'	67	0.092412	0.333333

Table 8 shows all the correlations found for the Machine 2 process between the list of tasks that must be scheduled (Table 5 and Table 6) and the results of the different levels of ARs (Table 7).

The first line highlights a correlation between the task on Product 1 and one rule present in ARL1. That rule has a consequence value of 75% of success rate with a confidence of 1. Thanks to this value is possible to calculate the corrective factor based on Equation (4).

$$\begin{aligned}\text{No of part to be manufactured} &= 10\\ &+ (10 * 0.25 * 1) = 10 + 2.5 = 13\end{aligned}$$

In this case, decimal numbers are rounded up.

Instead, lines 2 and 3 show two rules in ARL2 that characterise the process for Product 2. This means that in ARL1 there was no rule describing such a process while a group of ARL2 did. Thus, the corrective factor will be the contribution of two rules:

$$\begin{aligned}&\text{No of part to be manufactured}\\ &\quad = 10 + [(4 * 0.5 * 0.5) + (4 * 0 * 0.5)]\\ &\quad = 4 + 1 + 0 = 5\end{aligned}$$

Mathematical model application

The mathematical model illustrated in Section 3.3 was used to solve both the FSSP and the JSSP. The

Table 8
Correspondence between ARs and the task to be scheduled and its correction factor

ID	ARL	Antecedent	Consequent (Succes rate)	Confidence	Corrective factor
1	ARL1	(Machine 2, May, Product 1, Material 1, NP10)	75%	1	3
2	ARL2$_1$	(Machine 2, May, Product 2, Material 2)	50%	0.5	1
3	ARL2$_1$	(Machine 2, May, Product 2, Material 2)	100%	0.5	0
4	ARL2$_2$	(Machine 2, May, Product 3, NP5)	0%	0.25	1
5	ARL2$_2$	(Machine 2, May, Product 3, NP5)	100%	0.75	0
6	ARL2$_2$	(Machine 2, May, Product 4, NP5)	0%	0.33	2
7	ARL2$_2$	(Machine 2, May, Product 4, NP5)	100%	0.66	0
8	ARL3$_1$	(Machine 2, May, Product 5)	0%	0.5	2
9	ARL3$_1$	(Machine 2, May, Product 5)	100%	0.5	0

output of this phase consists of the scheduling plan with the related Gantt charts shown in Fig. 7 and Fig. 8. The figure shows a double Gantt diagram in which the first shows the schedule from the point of view of the articles, while the second shows the sequence of articles to be processed by each machine.

The proposed algorithm finds the minimum makespan value to realize a list of tasks and a correlation between the list of activities to be scheduled with relative historical data on a given production process.

The framework was tested on a computer with processor Intel 12th Gen Intel(R) Core(TM) i9-12900KF 3.20 GHz and 32 Gigabytes RAM and 'gurobi' as solver.

Concerning the JSSP, the algorithm returns the global optimum solution to the makespan value minimisation problem (760 minutes) in 1.39 seconds. With regard to the FSSP, both the makespan value (919 minutes) and, above all, the calculation time, which amounts to 1.41 seconds, are superior.

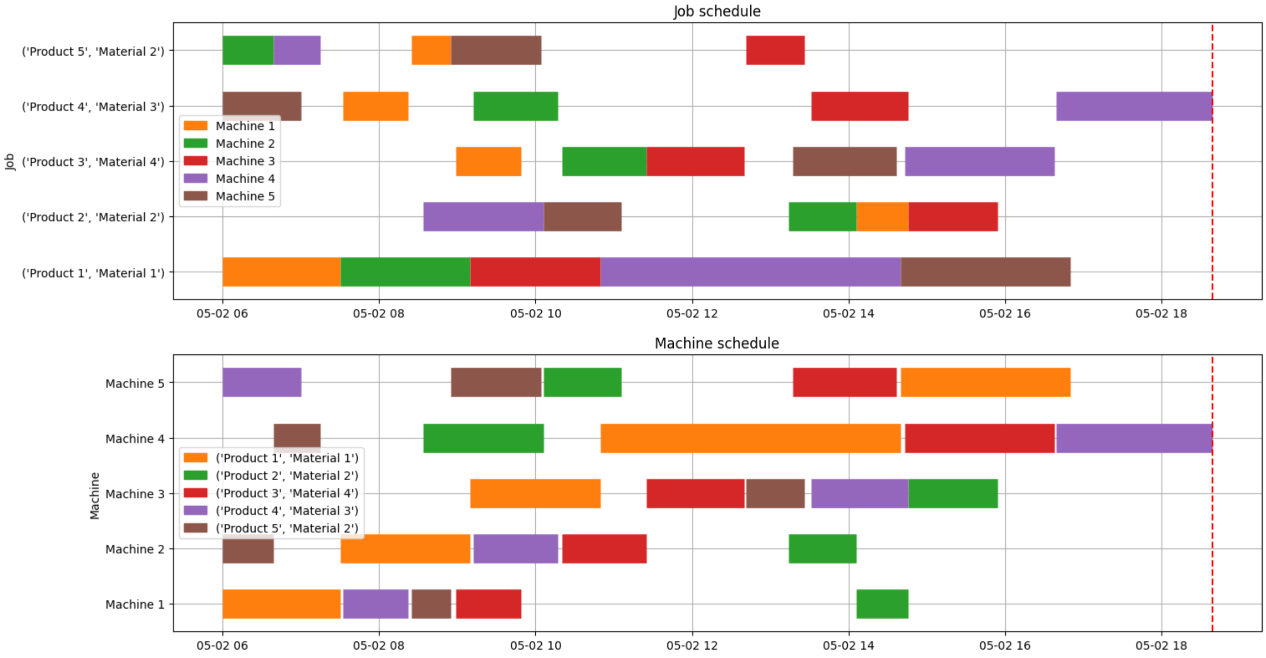

Fig. 7. Gantt chart relative to the JSSP

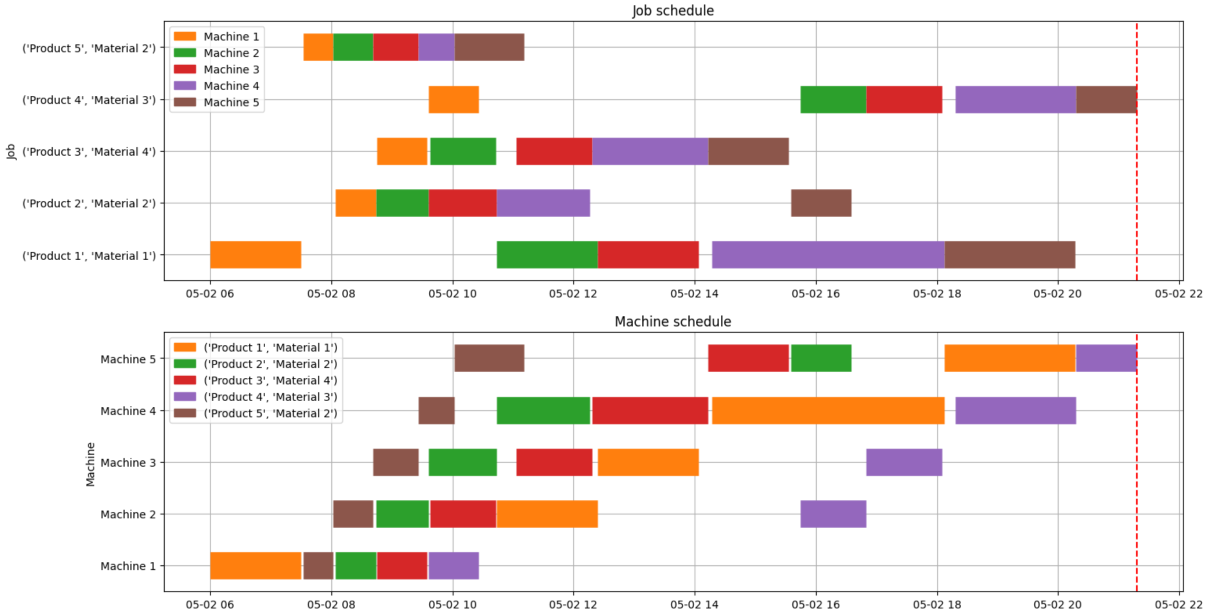

Fig. 8. Gantt chart relative to the FSSP

Once the scheduled products have been produced, the results of the machining will be entered into the database from which ARs are calculated so as to always provide continuous data and create a self-learning cycle.

Results and discussion

Determining and controlling the probability of success of one or more processes in an unstable production environment is of crucial importance to optimise the development of an accurate production planning program.

To comprehensively identify the connections among various factors and causes that influence the production program, ARs offer a suitable analysis approach for examining the parameters of the scheduling process. By considering both constant antecedents (attributes that remain unchanged throughout) and variable antecedents (attributes that may vary across machining operations), the algorithm presented in Section 3.1 allows managers to assess the probability of success of a given production process.

The extraction of ARs is influenced by the defined support and confidence levels. Setting a threshold greater than 0 may lead to the exclusion of certain rules, particularly those with a lower probability of occurrence. The significance of this parameter becomes more pronounced as the dataset size increases since the number of association rules to be mined directly impacts the algorithm's efficiency.

Using the results obtained from the ARs, it was possible to launch a scheduling plan that considers the high probability of non-conforming parts in a given process, thus suggesting a higher quantity to be produced based on historical data. This allows companies to meet delivery times in more detail, avoiding the need to frequently change the scheduling plan due to production problems. As far as calculation time is concerned, the two theoretical cases illustrated in the previous section show that in both cases calculation times are very low and can therefore also be implemented in real business cases. Table 9 shows the calculation times for different scenarios assumed. The first column shows the characteristics of the problem in terms of the number of products to be scheduled and the number of machines required for processing. The fourth column shows the number of total operations to be scheduled, while the third column shows the number of ARs found. Finally, the last two columns show the calculation times for the FSSP and JSSP problems.

Note how the calculation times remain very low (< 10 seconds) for all proposed problems with an increase in these times as the complexity of the problem increases.

The realization of a self-learning framework that combine a level structure of ARs with mathematical models for scheduling production orders is a novelty in literature. The level structure of the ARs guarantees different levels of accuracy of the solution depending on the availability of data. This aspect provides good

Table 9
Calculation times for problems of varying complexity

Test (no articles × no machines)	No rules	No operations	FSSP time [s]	JSSP time [s]
Test (3 × 3)	5	9	1.16	1.15
Test (5 × 3)	4	15	1.34	1.25
Test (7 × 3)	12	21	1.83	1.73
Test (9 × 3)	25	27	3.92	8.31
Test (3 × 5)	5	15	1.19	1.18
Test (5 × 5)	6	25	1.42	1.39
Test (7 × 5)	4	35	1.64	1.99
Test (9 × 5)	17	45	3.19	8.76
Test (3 × 7)	5	21	1.37	1.17
Test (5 × 7)	7	35	1.51	1.41
Test (7 × 7)	12	49	2.44	2.68
Test (9 × 7)	10	63	3.81	4.62
Test (3 × 9)	5	27	1.28	1.31
Test (5 × 9)	13	45	1.7	1.41
Test (7 × 9)	21	63	4.33	3.57
Test (9 × 9)	17	81	8.78	7.41

flexibility and robustness to the framework because it can also be applied to industrial realities where very large amounts of data on the production process are not available. Also, the possibility to update the AR extraction dataset with new production data allows for ever better levels of accuracy of the solution, increasing the overall performance value.

The proposed algorithm lends itself well to industrial applications due to its robustness and flexibility, but above all due to the low computation times found in different configurations of the scheduling problem.

Conclusions

This paper proposes a self-learning framework able to schedule production orders considering, from historical company data, the success rate of one or more stages of the production process. ARs are used to find correlations between a combination of different parameters about the production process and the success rate. Based on the activities to be scheduled, a set of ARs was searched for that best describes the production process. If among the various levels of ARs, there was one or more rules that described the process, a demand correction factor was calculated. This correction factor is calculated as a function of the confidence value of the rule and the percentage of good pieces realised. To solve the scheduling problem, the authors propose a mathematical model to find the global optimal solution which minimizes the makespan. In the modern company, the management of the production process has a strategic importance. The ability to predict errors during one or more stages of the production process is of fundamental importance, especially in decision-making processes supporting the production planner. The proposed framework assists the production planner in due fundamental steps: risk analysis in the production process and scheduling of production orders using a mathematical approach. This approach increases the possibility of being able to deliver a production batch on time and avoid delays caused by non-conformities during the production process.

The main limitation of this work concerns the information contained in the dataset used for the extraction of ARs. The more information one has about a given production process or the entire production line, the better one can characterise the process under investigation through the use of ARs.

Another missing information concerns the availability of raw materials and semi-finished products in stock that can help in the decision-making process to understand the real quantity that can be produced.

Further development may consider these two important aspects, the first one is to try to describe the production process more accurately. On the other hand, stock availability is important in order to consider the real availability of material and thus to understand whether the quantities to be produced suggested by the correction factor are possible. Future development will be to compare the results obtained in terms of solution quality and calculation time with other DM or AI techniques. One could envisage replacing the mathematical model, which requires high computation times for large problems, with an AI approach such as Reinforcement Learning algorithms, so that a scheduling program can be realised in real-time or near real-time, even for more complex problems.

Acknowledgements

We are very grateful for the support of the European Union's Horizon Europe research and innovation programme under grant agreement No. 101057294, project AIDEAS (AI-Driven industrial Equipment product life cycle boosting Agility, Sustainability and resilience).

References

Agrawal, R., & Srikant, R. (1994). Fast Algorithms for Mining Association Rules.

Babor, M., Paquet-Durand, O., Kohlus, R., & Hitzmann, B. (2023). Modeling and optimization of bakery production scheduling to minimize makespan and oven idle time. *Scientific Reports*, 13(1). DOI: 10.1038/s41598-022-26866-9.

Chen, Y. (2019). Research on Resource Allocation Optimization of Information Management System Based on Big data Association Mining. *Journal of Physics: Conference Series*, 1345(2). DOI: 10.1088/1742-6596/1345/2/022073.

Fani, V., Antomarioni, S., Bandinelli, R., & Bevilacqua, M. (2023). Data-driven decision support tool for production planning: a framework combining association rules and simulation. *Computers in Industry*, 144. DOI: 10.1016/j.compind.2022.103800.

Farizal, & Joelian, A. (2020). Engine replacement scheduling optimization using Data Mining. *Journal of Physics: Conference Series*, 1500(1). DOI: 10.1088/1742-6596/1500/1/012111.

Habib Zahmani, M., & Atmani, B. (2021). Multiple dispatching rules allocation in real time using data mining, genetic algorithms, and simulation. *Journal of Scheduling*, 24(2), 175–196. DOI: 10.1007/s10951-020-00664-5.

Jiménez-Pastor, A., & Petkovšek, M. (2023). The factorial-basis method for finding definite-sum solutions of linear recurrences with polynomial coefficients. *Journal of Symbolic Computation*, 117, 15–50. DOI: 10.1016/j.jsc.2022.11.002.

Nasiri, M.M., Salesi, S., Rahbari, A., Salmanzadeh Meydani, N., & Abdollai, M. (2019). A data mining approach for population-based methods to solve the JSSP. *Soft Computing*, 23(21), 11107–11122. DOI: 10.1007/s00500-018-3663-2.

Qiu, Y., Sawhney, R., Zhang, C., Chen, S., Zhang, T., Lisar, V.G., Jiang, K., & Ji, W. (2019). Data mining–based disturbances prediction for job shop scheduling. *Advances in Mechanical Engineering*, 11(3). DOI: 10.1177/1687814019838178.

Raschka, S. (2018). MLxtend: Providing machine learning and data science utilities and extensions to Python's scientific computing stack. *Journal of Open Source Software*, 3(24), 638. DOI: 10.21105/joss.00638.

Troncoso-García, A.R., Martínez-Ballesteros, M., Martínez-Álvarez, F., & Troncoso, A. (2023). A new approach based on association rules to add explainability to time series forecasting models. *Information Fusion*. DOI: 10.1016/j.inffus.2023.01.021.

Wang, L., Lin, B., Chen, R., & Lu, K.H. (2022). Using data mining methods to develop manufacturing production rule in IoT environment. *Journal of Supercomputing*, 78(3), 4526–4549. DOI: 10.1007/s11227-021-04034-6.

Wu, Y., Yao, L., Liu, J., & Zhuang, C. (2018). A New Method of Resource-Scheduling-Strategy Generation for the Assembly of Complex Products Based on the Apriori Algorithm (IEEE). IEEE.

Zhang, Y., Zhu, H., Tang, D., Zhou, T., & Gui, Y. (2022). Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. *Robotics and Computer-Integrated Manufacturing*, 78. DOI: 10.1016/j.rcim.2022.102412.

Zhao, A., Liu, P., Gao, X., Huang, G., Yang, X., Ma, Y., Xie, Z., & Li, Y. (2022). Data-Mining-Based Real-Time Optimization of the Job Shop Scheduling Problem. *Mathematics*, 10(23). DOI: 10.3390/math10234608.