

# Formal Scheduling Method for Zero-Defect Manufacturing

Katarzyna Grobler-Dębska

AGH University of Science and Technology

Edyta Kucharska

AGH University of Science and Technology

Jerzy Baranowski (✉ [jb@agh.edu.pl](mailto:jb@agh.edu.pl))

AGH University of Science and Technology <https://orcid.org/0000-0003-3313-581X>

---

## Research Article

**Keywords:** Formal methods, Zero-defect manufacturing, Scheduling

**Posted Date:** March 29th, 2021

**DOI:** <https://doi.org/10.21203/rs.3.rs-315238/v1>

**License:** © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

---

**Version of Record:** A version of this preprint was published at The International Journal of Advanced Manufacturing Technology on October 22nd, 2021. See the published version at <https://doi.org/10.1007/s00170-021-08104-0>.

# Formal Scheduling Method for Zero-Defect Manufacturing

Katarzyna Grobler-Dębska · Edyta Kucharska · Jerzy Baranowski

Received: date / Accepted: date

**Abstract** Zero-defect manufacturing (ZDM) is a concept of quality management that recently exhibits a renewal of interest. It aims to improve the process efficiency and the product quality while eliminating defects and minimizing process errors. Most of ZDM applications focus on using the technological achievements of Industry 4.0 to detect and predict defects, forgetting to optimize the schedule on the production line.

This paper proposes a formal approach for solving scheduling problems with unexpected events as extension of general frameworks for ZDM strategy. We propose formal method to create predictive-reactive schedule for problems with defect detection and repair. Our formal model combines defect detection, heuristics construction and decision support. We illustrate our results with examples.

**Keywords** Formal methods · Zero-defect manufacturing · Scheduling

## 1 Introduction

In modern market conditions manufacturers need to quickly deliver high quality products. This requires rapid reaction to various events in the production process. This led to development of several methodologies for quality control and improvement. In majority they have been concentrated on single-stage processes such as statistical process control (SPC), design of experiments (DoE), acceptance sampling procedures, six-sigma and

lean tools (Barbosa et al., 2018). In contrast, the Zero-Defect Manufacturing (ZDM) strategy concerns the operations in the entire organization. Its goal is to improve the process efficiency and product quality while eliminating defects and minimizing process errors.

While ZDM as strategy is attractive, theoretical and practical works do not consider its relationship with work scheduling problem. Most works treat the problem of job assigning to machines as a simple queuing system e.g. FIFO (Konrad et al., 2013). It however ignores inherent uncertainty of real systems. Industry needs a coherent framework merging solutions of automatic process data acquisition, data mining and scheduling under uncertainty. There are conceptual propositions for solutions (Dreyfus and Kyritsis, 2018; Psarommatis and Kiritsis, 2018), but there are no concrete results.

In the field of scheduling, most of the approaches for real manufacturing systems and unexpected events are “machine-oriented”. We mean by it that the algorithms mostly consider possibilities of machine failures (Abedinnia et al., 2017; Liu, 2019; Shen and Zhu, 2017). “Product-oriented” and ZDM aligned approaches, which combine disparate disturbances are uncommon. An example of a “product-oriented” disturbance is a quality defect in a manufactured item.

Typically scheduling under uncertainty works considers predictive schedules using probabilistic uncertainty models. Their goal is to determine a schedule with good average performance (Dias and Ierapetritou, 2016). This is not a methodology that can be used in ZDM oriented Decision Support Systems (DSS). Performance analysis of predictive-reactive rescheduling in dynamic environment is still open problem (Cardin et al., 2017).

We address these issues, by proposing an approach that treats defects and their removal as an inherent

---

K. Grobler-Dębska, E. Kucharska and J. Baranowski  
Department of Automatic Control and Robotics  
AGH University of Science & Technology  
al. Mickiewicza 30, 30-059 Kraków, Poland  
E-mail: {grobler,edyta,jb}@agh.edu.pl

part of the process. Our main contribution is a formal approach to solving scheduling manufacturing problems with defects removal, which is an extension of the ZDM frameworks. In particular, we propose the use of an Algebraic-Logical Meta Model (ALMM). We consider a class of non-deterministic discrete manufacturing problems, in which unexpected events occur during the process execution and impact job results. This general approach is presented with an examples of scheduling problems with product fault detection during quality control and their repair.

The paper is organized as follows. In section 2 we describe ZDM and its frameworks. Section 3 presents algebraic-logical meta model (ALMM) and how it can be used in predictive-reactive scheduling. In section 4 we propose using ALMM switching as ZDM scheduling strategy. In section 5 we discuss applications of this method for flow-shop system with defects along with examples. Paper ends with concluding remarks.

*Remark 1 (Notation)* In this paper by  $(a, b)$  we will consider a tuple (pair), and not an interval. We also use  $\tau_{idle}$  symbol to determine that there is an infinite time to completion, or in other words that machine is not working.

## 2 Zero-Defect Manufacturing

ZDM is a comprehensive strategy and encompasses both short- and long-term perspectives. The short-term perspective ranges a real-time process control system developing and implementing that eliminates the production of a faulty component due to variances in materials, components, and process properties. The long-term perspective is concerned with minimizing all failures by continuously optimizing the production process and the manufacturing system. The basic idea of Zero-Defect Manufacturing was introduced in the early 1960s in connection with the US Army Perishing Missile System. This concept has been implemented only partially due to numerous technological limitations that were prohibiting its implementation. According, to Psarommatis et al. (2020) "currently, with the evolution of Industry 4.0, ZDM concept is easier to be implemented due to the availability of the required amount of data for techniques such as machine learning to work properly but still a lot of effort is needed for better integration and coordination of the capabilities of each technology". Thus, ZDM is now a common manufacturing practice to reduce and minimize the number of defects and errors in a process and to do things right the first time (Wang (2013)).

Nowadays, it is possible to implement the theoretical assumptions of the ZDM strategy because there are many solutions based on advanced technology of automatic data acquisition from machines and devices, and data mining technology focus to obtain automatics defect detection. Therefore, automatic fault prediction and prevention are commonly possible (Bufardi et al. (2017); Chien et al. (2013); Wang (2013); Teti (2015); Ju et al. (2013); Kuo et al. (2013)).

The ZDM strategy is attractive for companies for the following reasons:

- reducing the costs of the company's resources related to the treatment of defective products,
- reducing defective machines and tools, inefficient employees, reduction of scrap production,
- increasing safety and customer satisfaction, making production eco-friendly.

The Zero-Defect Manufacturing can be implemented in two different approaches: the product-oriented ZDM and the process-oriented ZDM. The difference is that a product-oriented ZDM studies the defects on the actual parts and tries to find a solution while on the other hand the process-oriented ZDM studies the defects of the manufacturing equipment, and based on that can evaluate whether the manufactured products are good or not. The latter one lays within the predictive maintenance concept (Psarommatis et al. (2020)).

The concept of zero-defect can be practically utilized in any manufacturing environment to improve quality and reduce cost. Therefore, the whole system should contain the following main phases:

- Data acquisition – automatic capture, cleaning and formatting of relevant data using intelligent sensors system;
- Signal processing – automatic signal processing, filtering and feature extraction;
- Diagnosis – data mining and knowledge discovery for diagnosis;
- Prognostic assessment – data mining and knowledge discovery for prognostic;
- Maintenance scheduling – optimization tool for dynamic scheduling under uncertainty.

There are only a few papers with a proposition of a holistic approach that contains all phrases. The last phase is usually omitted. The table 1 presents the approaches to ZDM strategies and their contribution for ZDM and scheduling maintenance.

There are assumptions in all the papers that all scheduling problems can be modeled and solving using well-known meta-heuristic methods. Any specific model and any proposition of adaptation of these meta-heuristic methods to problems under uncertainty haven't

**Table 1** Survey of the ZDM frameworks illustrates that scheduling is analyzed in an only a general form. There aren't any particular classes of problems which can be solved using the frameworks and there are no specialized methods.

Paper	Main contribution for ZDM	Maintenance scheduling
Psarommatis and Kiritsis (2018)	Conceptual framework of scheduling tool for achieving zero-defect manufacturing. The propose decision support system (DSS) which will consist of three individual components: (a) the defect detection module, (b) the defect prediction module and (c) decision making algorithms for suggesting a solution based on user-defined rules. When real time events, that disrupt the production process, are happening the output of the DSS is fed into the dynamic scheduling tool along with the shop floor status and specification.	Lack of information which algorithms or methods will be used in scheduling tool and which optimization criteria will be implemented. Only the Key Performance Indicators (KPIs) will be calculated and if they will be acceptable the updated schedule is released.
Dreyfus and Kyritsis (2018)	Combine three technologies, i.e. tuning assistant, predictive maintenance and automatic scheduling algorithm and it is now implemented in a case study of a high-end production company. The macro solution merge three solutions that will reshape production floors such as machine-oriented, product-oriented and production-oriented. An optimal production floor can be defined with those three ideas: Machine producing, Product conform and Production optimized.	In this solution the automatic scheduling algorithm is proposed, but the author only mentioned existing general methods without proposal of using specific algorithms which can optimize this kind of problems such as two-stage stochastic programming, parametric programming, fuzzy programming, chance constraint programming, robust optimization techniques, conditional value-at-risk.
Wang (2013)	Proposition of general framework of ZDM. The basic components of a ZDM system include intelligent sensor and data collection, signal processing and feature extraction, ZDM for diagnosis and prognosis, decision-making and feedback control.	The framework module for maintenance optimization (or maintenance scheduling) is describe only briefly. There are mentioned many literature's implementation of traditional methods such as Genetic Algorithm (GA), Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) in maintenance optimization. In such problem these methods should be specific version to deal with scheduling under uncertainty.
Myklebust (2013)	General framework of ZDM is developed. Proposed framework has five modules: Sensors for data acquisition; Signal processing (based on Fourier Transformation, Discrete Fourier Transform (DFT), Fast Fourier Transform (FFT), Wavelets Transformation); Fault Classification and Diagnosis (based on Statistical Matching, Evaluation Algorithm, Data Mining and Knowledge Discovery); Fault Prognosis (based on Maintenance Optimization, Autoregressive moving average, ANN prediction, Fuzzy Logic Prediction, Match Matrix Prediction)	Maintenance Scheduling Fault (based on Meta-Heuristic approaches: Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO)). Scheduling is based on general methods without stochastic components.
Lindström et al. (2019)	The paper makes a contribution to the literature by outlining a ZDM approach which combines and integrates: monitoring process parameters, collaborative manufacturing, continuous quality control (input and output), on-line predictive maintenance, data storage and analytic, and re-configuration/re-organization/re-scheduling of manufacturing/production.	In this paper there are any specific solution how re-scheduling of manufacturing should be done.

proposed. Furthermore, it hasn't mentioned that the scheduling task should take into account various aspects of interference to both machines and products. This is still a rare problem in scheduling due to the difficulty of uncertainty problems. As part of the rescheduling approach, algorithms are proposed for solving problems

of scheduling with machine failures, new incoming jobs during simulation, and stochastic job processing times (Liu (2019); Shen and Zhu (2017)). However, less often it can find approaches combining several of these disturbances and other disturbances such as canceling the order, changing the priority of the order of execution of

jobs, inability to perform the jobs due to unavailability of materials, or an unsuccessful technological process and the need to repair an incorrectly performed job.

### 3 Reactive-predictive scheduling with Algebraic-Logical Meta-Model (ALMM)

If we want to consider scheduling problems that satisfy the ZDM strategy we need to consider scheduling problems under uncertainty. This kind of schedule should be predictive-reactive. Predictive-reactive scheduling consists of two stages: the planning stage and the execution stage. Predictive scheduling consists of the planning stage and it is off-line scheduling which taking into account uncertainty and flexibility of the executed process. Reactive scheduling refers to the execution stage and it is an on-line phase in which the schedule is created or modified in production when an event occurs. In this section, we present ALMM methodology and resulting from it a formal mechanism for generating predictive-reactive scheduling. This allows early detection of product defects, the adaptation to operating condition changes and the optimisation of manufacturing processes. In this section we will show how our methodology allows creation of predictive-reactive schedule.

#### 3.1 Predictive scheduling

Considered approach is based on a multistage decision-making process, where the decision is made jointly and considers the current situation (current process state). This approach is the process trajectory-based simulation, where the solution is constructed in stages. Thus, there is a possibility to consider the existing constraints or dynamic input data at a stage (given state of the problem). Moreover, possible decisions are also state-dependent. Using ALMM allows us to reconstruct the process of decision making as well as to monitor and track decisions during the manufacturing process.

ALMM of the discrete deterministic process was defined by Dudek-Dyduch (1989, 1990). According to this methodology, an optimization problem is modelled as a multistage decision process together with optimization criterion  $Q$ . It is defined as a sex-tuple which elements are: a set of decisions  $U$ , a set of generalized states  $S = X \times T$  ( $X$  is a set of proper states and  $T$  is a subset of non-negative real numbers representing the time instants), an initial generalized state  $s_0 = (x_0, t_0)$ , a transition function  $f(s, u)$ , a set of not admissible generalized states  $S_N \subset S$ , a set of goal generalized states  $S_G \subset S$ .

*Transition function*  $f$  is defined by means of two functions:  $f_x : U \times X \times T \rightarrow X$  determines the next state and  $f_t : U \times X \times T \rightarrow T$  determines the next time instant. As a result of the decision  $u$  that is taken at some proper state  $x$  and a moment  $t$ , the state of the process changes to  $x' = f_x(u, x, t)$  that is observed at the moment  $t' = f_t(u, x, t) = t + \Delta t$ . All limitations concerning the control decisions in a given state  $s$  can be defined in a convenient way by means of so-called sets of possible decisions  $U_p(s)$ .

The optimization task is to find an admissible decision sequence  $\tilde{u}$  that optimizes criterion  $Q$ . The consecutive process states are generated as follows. A process starts with an initial state  $s_0$ . Each next state depends on the previous state and the decision made at this state. The decision is chosen from different decisions, which can be made at the given state. Generation of the state sequence is terminated if the new state is a goal state, a non-admissible state, or a state with an empty set of possible decisions. The sequence of consecutive process states from the given initial state to a final state (goal or non-admissible) form a process trajectory.

The advantages of ALMM are that values of particular co-ordinates of a state or/and decision may be names of elements (symbols) as well as some objects, e.g. a finite set, sequence, etc. (thus they do not have to be numerical) and the set of possible decisions  $U_p$ , set of not admissible generalized states  $S_N$  and a set of goal generalized states  $S_G$  are formally defined with the use of logical formula. It allows representing all kinds of information regarding the mentioned problem (including various temporal relationships and restriction of the process) in a convenient way (Dudek-Dyduch, 1990). It should be noted that the set of possible states is dependent on previously made decisions and in any stage decision is made based on current data and system limitations. During a simulation any decision can't be made that would lead the system to not admissible state. The heuristic optimization algorithm based on the ALMM is designed to maximize the criterion function and determine the next state out of the set of not admissible states. Moreover, considered model gives a possibility for a formal analysis of admissibility.

Considered approach allows creation of a predictive schedule. A set of possible decisions should be designed in such a way as to take into account additional information (not yet possible in a given state) but predicted and possible in the near future. It is not mandatory to assign a new job to a machine or resource as soon as possible (as soon as the machine finishes an operation and is ready to start the next one). It is possible to design a decision not assigning any new job to the machine if

that is not preferred. Preferences can be predicted. This is a different approach than the typical queuing scheduling system. Moreover, as shown by Kucharska (2019) instead of only using information about currently available jobs or machines, ALMM can include additional information about the expected availability in the future. In such a case time can be calculated based on the given rules and restrictions of the problem or predicted based on historical data.

### 3.2 Reactive scheduling

Reactive scheduling can be built using *ALMM switching method*. This method provides a switch model when some event occurs and creates a partial re-schedule. A switching method is a formal approach to the model discrete multi-stage decision process in which some unexpected events have occurred. This method was considered in Grobler-Dębska et al. (2013); Grobler-Dębska (2019); Kucharska et al. (2017b).

Here we focus on processes in which unexpected events include detection of a defect in the product. A practical examples of such event are: irregular application of paint on a painted element, the bad balance of springs in the mattress, or back filling of the hollow headings. This type of defect should be detected and repaired to achieve the postulates of ZDM.

The ALMM switching method consists of the following elements:

- analysis of types of disturbances,
- determination of the set of switching states  $\mathcal{S}_{switch}$ ,
- division of problem to subproblems,
- algebraic-logical models (*ALM*) creation for all subproblems,
- switching type specification,
- determination of the switching rules  $\mathcal{R}_{switch}$ ,
- definition of the switching function .

**Analysis of the problem.** This step deals with determining some characteristic problem features. In particular, limitations or requirements reflected in the problem are detected. The preferences of certain types of states or decisions are determined. Moreover, both kinds of disturbances and the moment of occurrence are identified.

**Determination of the set of switching states specification  $\mathcal{S}_{switch}$ .** The set of switching states includes three types of states:

- all states when a new disturbance occurs,
- the states when previously detected disturbance is removed for all types of disturbance

- the states when previously detected disturbance is removed and a new disturbance occurs in the same process state.

In the discussed problem with product quality control, the set of switching states includes the states when 1) at least one machine with quality control finishes its job and quality control detects some fault and 2) any additional machine has just finished repairing a job with the defect.

**Division of problem to subproblems.** Adequate division of the initial non-deterministic problem into subproblems is crucial and related with the types of defects. For manufacturing problems with quality defects, the initial problem includes all types of disturbances and all repair machines dedicated to these disturbances. Division of problem to subproblems reflects the possible consequences of the event. In this problem, it is necessary to introduce the quality repair process. Subproblems are constructed as follows:

- The first problem is the classic deterministic scheduling problem in which the occurrence of quality defect is not considered, but only the possibility of detecting them i.e. a quality control station is considered (within the  $M_q$  machine or several machines with quality control  $\mathcal{M}_Q$ ). The problem defines how a given set of jobs without disturbances is processed for given processing times at a given set of machines.
- Further scheduling problems in a given system consider the following issues: specific types of quality defects repairment and a set or sequence of dedicated repair machines for these types of deficiencies.

The division into subproblems causes that in a given state of the system the proper subproblem is considered, which includes only required repair machines.

**Algebraic-logical models (*ALM*) creation for all subproblems.** All subproblems are modeled as algebraic-logical models *ALM*. The structure of ALMs depends on the type of data changes and may be the same or different. The advantage is to specify simpler models of simpler problems than the original initial problem. In the simplest case, we can use the same model with different parameters. Furthermore, for some problems, with various types of data changes we can specify several models of simpler problems (subproblems of initial problem).

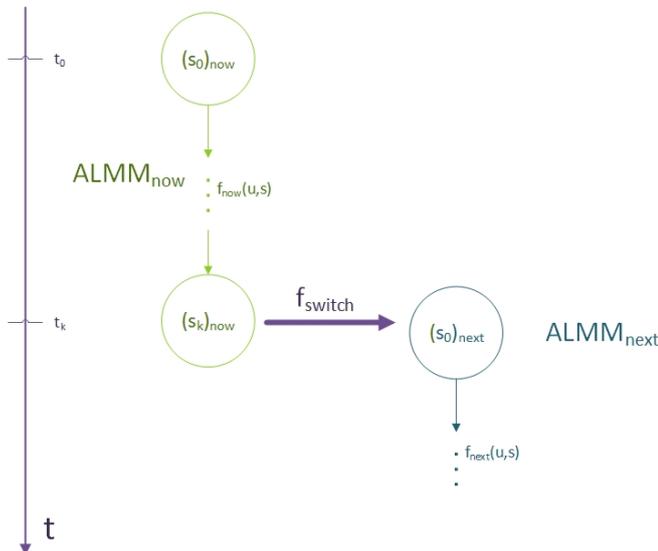
**Switching type specification.** It depends on the number of disturbance types. There are four types of switching for one type of disturbance:

- the system switches from the process without the disturbance to process with disturbance,

- the system switches from the process with disturbance to the same process with modified parameters and data, in the case of new disturbance detection when the earlier disturbance have not been removed yet or only the part of disturbance has just been removed,
- the system switches from the process with disturbance to process without disturbance when all disturbance have just been removed.

In the problems with more then one type of disturbances the system can switch from the process with one type of disturbance to process with the other type of defect. In such a case it should be analysed all the possibilities of switching when the system switches from a process with one kind of disturbance to the process with another type of disturbance or to the process without any disturbances.

**Determination of the switching rules  $\mathcal{R}_{switch}$ .** Switching rules determine how, as a result of an event, the current state of the production process (represented by  $s_k$  of the current model  $ALM_{now}$ ) goes to the next state of the process (represented as the initial state  $s_0$  usually in another model  $ALM_{next}$ ). The new state determined in this way includes changes that are a consequence of the event. The set of these rules will determine the switching function.



**Fig. 1** The idea of generating a single process trajectory with switching between models. Switching occurs as a result of an event in the current state of the production process (represented by  $s_k$  of the current model  $ALM_{now}$ ) and goes to the next state of the process (represented as the initial state  $s_0$  usually in another model  $ALM_{next}$ .)

**Definition of the switching function  $f_{switch}$ .** Determining the rules depends on types of switching. It consists of the determination of algebraic formulas to recalculate a value of parameters in case of switching between the same  $ALM$  or calculate a new state and new value of parameters in case of switching between different  $ALMs$ . The switching rules usually don't have to be set for all combination of switching between models. It should be excluded the switching between those models, which are not related.

Let us consider the following notation. Let  $\mathcal{S}_{switch}$  denote the set of switching states from all the models under consideration. Let  $\mathcal{D}$  denote the space of data sets. Let  $\mathcal{ALM}$  denote the set of all considered models for the initial problem. Let  $f_{switch}$  denote the switching function, which is defined as follows:

**Definition 1** Switching function is a function which to the model  $ALM \in \mathcal{ALM}$  in the current switching state  $s \in \mathcal{S}_{switch}$  and the data of subproblem determines the appropriate model and the corresponding data of subproblem:

$$f_{switch} : \mathcal{ALM} \times \mathcal{S}_{switch} \times \mathcal{D} \rightarrow \mathcal{ALM} \times \mathcal{D}. \quad (1)$$

The switching function is defined by means of two functions:

$$f_{switch} = (f_{switch|_{\mathcal{ALM}}}, f_{switch|_{\mathcal{D}}}), \quad (2)$$

where:

$f_{switch|_{\mathcal{ALM}}} : \mathcal{ALM} \times \mathcal{S}_{switch} \times \mathcal{D} \rightarrow \mathcal{ALM}$  determines the next model,

$f_{switch|_{\mathcal{D}}} : \mathcal{ALM} \times \mathcal{S}_{switch} \times \mathcal{D} \rightarrow \mathcal{D}$  determines the next set of data for subproblem.

The switching function determines the next system state based on the previous state and detected disturbances, without taking decision in this state whereas the transition function in the model determines the next system state based on the previous state and the decision.

**Switching method** specifies the 5-tuple of

$$(\mathcal{D}, \mathcal{ALM}, \mathcal{R}_{switch}, \mathcal{S}_{switch}, f_{switch}),$$

where:

- $\mathcal{D}$  - set of data and parameters of problem, which is the job set  $\mathcal{J}$ , the machine set  $\mathcal{M}$ , the resources set  $\mathcal{R}$  and their parameters
- $\mathcal{ALM}$  - set of models of considered subproblems,
- $\mathcal{R}_{switch}$  - a set of switching rules between models
- $\mathcal{S}_{switch} = (X_{switch}, T_{switch})$  is a set of states in which switching occurs,
- $f_{switch}$  is a switching function.

Therefore, the switching algorithm for the production problems with quality deficiencies can be represented as follows:

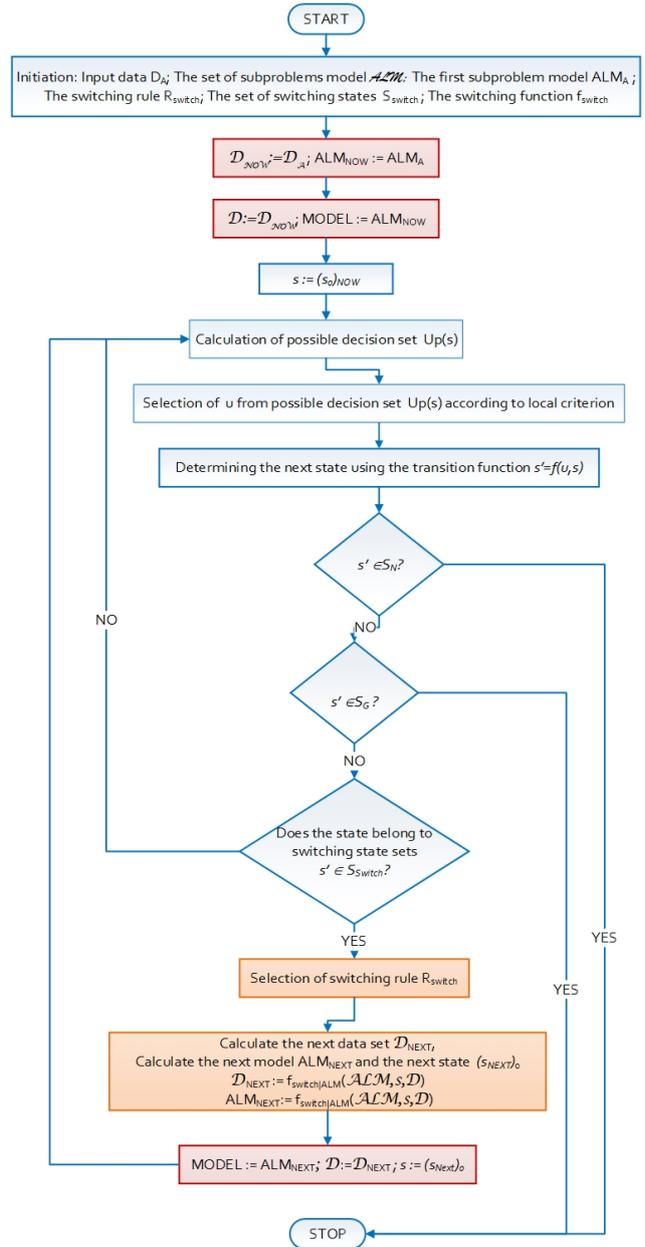
1. Determining all components of the switching method, i.e.:
  - input data  $\mathcal{D}_A$  for the first subproblem: defining the job set  $\mathcal{J}_A$ , the machine set  $\mathcal{M}_A$ , the resources set  $\mathcal{R}_A$  and their parameters,
  - algebraic-logical models of subproblems and determination of  $ALM_A$  - the model of the first subproblem,
  - the switching rules  $R_{switch}$ ,
  - the switching states set  $S_{switch}$ ,
  - the switching function  $f_{switch}$ .
2. Determining the current data  $\mathcal{D}_{now}$  and the current model  $ALM_{now}$ .
3. Determining the initial state of the system state of the current model  $(s_{now})_0$ .
4. Determining the next state of the system  $s'$  using the transition function of the current model  $s' = f(u, s)$  and the taken decision.
5. Checking if the state belongs to the set of switching states  $s \in S_{switch}$ . If not, then determine the next states according to 4. If yes, go to the next Step 6.
6. Selecting the switching rule  $R_{switch}$ .
7. Based on the transition function, determining new data  $\mathcal{D}_{next}$ , model  $ALM_{next}$  and the initial state of the new model  $(s_{next})_0$ . Then go to 4.

The algorithm flowchart is presented in the Figure 2.

### 3.3 Optimization algorithms

The trajectory generation is a process controlled by various general methods and/or specific optimization algorithms or algorithms for searching admissible solutions. Such general methods include the graph search algorithms and general methods (e.g., Branch & Bound). A special type of method is meta-heuristics and algorithms based on the ALMM methodology. The following methods using ALMM of multistage decision process have been proposed in earlier works:

- a method that uses a specially designed local optimization task and the idea of semi-metric — the special local optimization criterion  $q(u, x, t)$  consists the following three parts: components related to the value of the global index of quality for the generated trajectory, components related to additional limitations or requirements and components responsible for the preference of certain types of decisions resulting from problem analysis and the local opti-



**Fig. 2** The general algorithm of the switching method for scheduling problems of production with quality deficiencies. In contrast to the classic algorithm for generating trajectories using ALMM, this algorithm does not only check whether or not the calculated state belongs to the goal or to the not admissible states but also check belonging to the switching states. Also, it specifies the conditions for the ALMM model changes.

mization task may use the semi-metrics term in the state space (Dudek-Dyduch, 2000);

- learning-based methods — these use the above specially designed local optimization task and the idea of semi-metric and additionally gather the information about the quality of solution during a searching process. Learning is realized in such a way that

gathered information is used to change the coefficients of the local optimization task components (Dudek-Dyduch and Kucharska, 2011; Kucharska and Dudek-Dyduch, 2014);

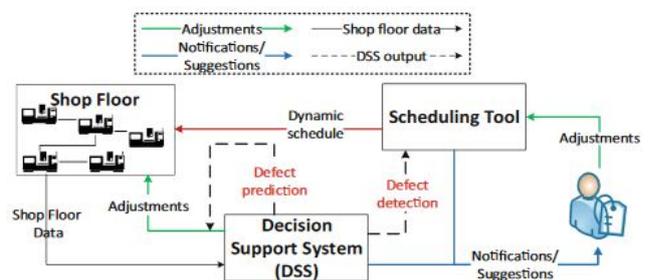
- a method based on the learning process connected with pruning non-perspective solutions — an extended learning-based method in which it is possible to use gathered knowledge to also change local optimization task form. Some components can be removed or new components related to new limitations can be added because new additional limitations may arise in the current state. Additionally, collected information is used to prune non-perspective trajectories. It should be noted that every pruning, partial trajectory may be analyzed because pruning trajectories are this kind of source of information that can be used to modify the coefficients of a local criterion form (Kucharska, 2017);
- substitution tasks method — a solution is generated using a sequence of dynamically created local optimization tasks, so-called substitution tasks. This task is a certain substitution multistage process with the substitution criterion. Substitution tasks are created to facilitate the decision making at a given state by substituting global optimization task with a simpler local task. A new or modified substitution task is defined based on information gained during an automatic analysis of the new process state. To construct the substitution task the concept of so-called intermediate goals is used, which determine a certain set of final states of the substitution process (Dudek-Dyduch and Dutkiewicz, 2013);

ALMM-based approaches allow solving non-deterministic discrete optimization problems. To optimize the scheduling problem with defect it should be used the combination of thus meta-heuristics and switching method. The searching algorithm determines the deterministic problem solution on the basis of discrete process simulation. ALMM switching method is used when events occur during the process. It allows to present a problem using simple models and switching function, which specifies the rules of using these models. Paper (Kucharska et al., 2017a) presents the combination of a hybrid algorithm with special local optimizing task and algebraic-logical models switching. In such case, a local criterion definition consists in determining particular components based on identifying limitations, requirements and preferences. The definition of component related to the estimation of the quality index for the final trajectory section is also important (it may have different forms and precision). Moreover, the modification of the local optimization task could be defined. It should be noted

that the local criterion form can be designed in different ways. Firstly, the components may be determined based on the properties of the problem without disturbance. Then the criterion in such a form can be used for a process with the disturbance. The second way is to design an additional component or components associated with the considered disturbance. When one has identified more different types of disturbances, suitable ingredients may be used appropriately. Therefore, in each state a set of decisions that can be taken  $U_p(s)$  is generated and the best decision related to the local criterion is selected.

#### 4 Proposed ZDM scheduling strategy

A scheduling tools for achieving Zero-Defect Manufacturing (ZDM) has been proposed by Psarommatis and Kiritsis (2018). This is a new approach. The authors have presented the conceptual framework of the proposed tool with the informational flow among the scheduling tool, the Decision Support System (DSS), and the shop-floor (Figure 3). The DSS utilizes real-time data from the shop-floor in order to predict or detect a defect. Then the outcome of the DSS is fed into the scheduling tool. The scheduling tool updates the schedule according to the suggestions produced by DSS. However, this tool is still a concept without the details of the proposed solution. Neither methods nor algorithms for specific solutions are given. Therefore, we propose ZDM scheduling strategy based on ALMM approach. It provides not only the possibility of formally defining the problem, but also we have already developed specific solutions. In particular, the product-oriented algorithms were tested (Kucharska et al., 2017a).



**Fig. 3** Scheduling tool for achieving ZDM proposed in the Psarommatis and Kiritsis (2018). The framework consists the integrated modules of Decision Support System and Scheduling Tool.

The proposed approach includes both decision support system and scheduling tool.

According to specification of the Decision Support System (DSS), given by Psarommatis and Kiritsis (2018), tool will consist of three individual components:

- (a) the defect detection module,
- (b) the defect prediction module,
- (c) the decision making algorithms for suggesting a solution based on user-defined rules.

In our proposition the DSS tool uses switching method and consist of:

- ALMM modeller which allows one to define the models of the problem in ALMM methodology. It includes: creating an algebraic-logical model of the process, setting the optimization criterion, and defining the additional knowledge about the problem (e.g., as auxiliary procedures or inference rules). The model can be built in different ways, e.g., by using pre-defined components or by defining it from the scratch. The modeller was presented in (Dudek-Dyduch et al., 2014; Raczka and Kucharska, 2017).
- ALMM switching modeller which allows one to define all elements of switching method. It includes: analysis of types of disturbances, determining the set of switching states, number of subproblems and their model, specifying types of switching, determination of the switching rules.
- Database that collects previously developed ALMM models, ALMM switching models with the obtained solutions, using the algorithms stored in the algorithm module.
- Predictive results interpreter module which identifies the solution of the given problem (the best one or an admissible one) on the basis of the data stored in the database.

The scheduling tool works in real time. The output of the DSS is fed into the dynamic scheduling tool along with the shop-floor status and specification. The new schedule is produced according a specific set of user defined criteria.

In our proposition the scheduling tool consists special algorithm module which provides a collection of already implemented methods and algorithms described in section 3.3. It also allows one to design new algorithms, especially meta-heuristics based on the ALMM methodology and ALMM switching method. The ALMM Solver which consist ALMM modeler and algorithm module was partially implemented and presented in (Raczka and Kucharska, 2017; Kucharska and Raczka, 2018).

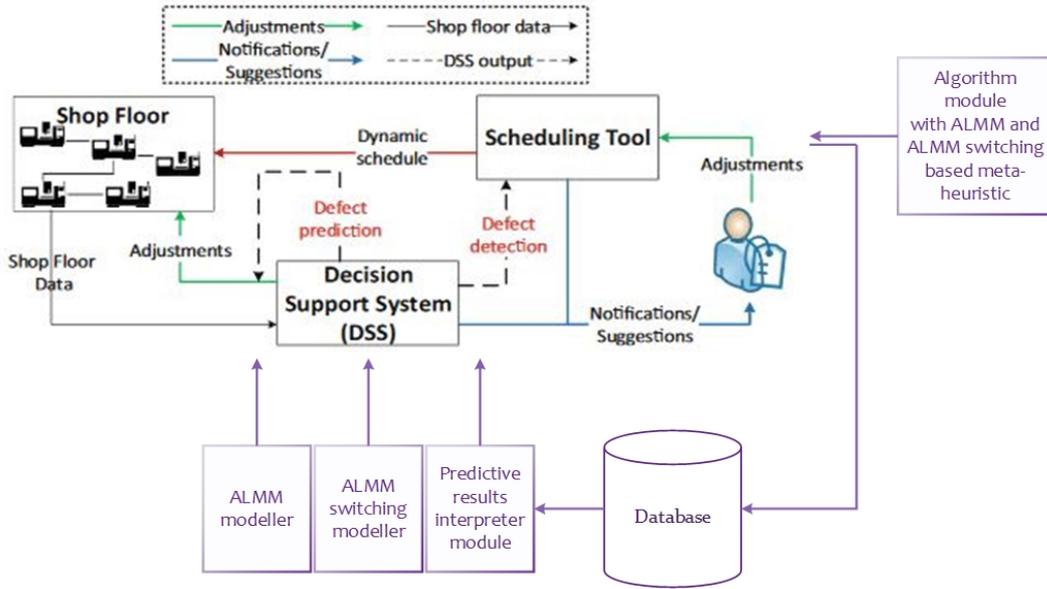
Thus implementation based on ALMM methods in the ZDM framework can give some benefits as:

- the ALMM models define multi-stage decision problems, so it can be used to defining decision making rules dynamically,
- the switching method can be used to the defect detection and initiate rescheduling in scheduling tool,
- the defect prediction function can be designed based on data from the switching method and some time-series prediction algorithm.

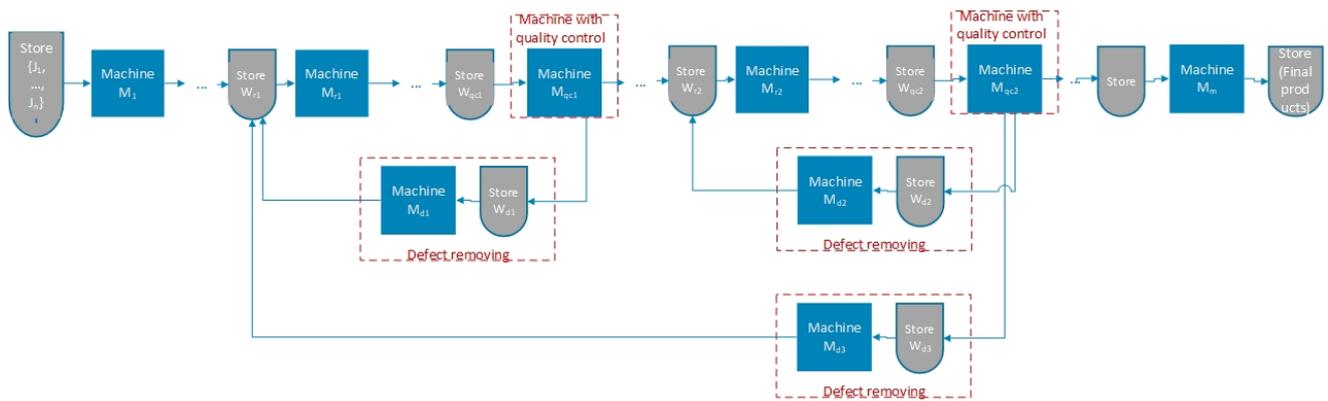
## 5 Application of proposed approach in flow-shop system with different types of product defect

Let us consider flow-shop manufacturing problem with quality control for the defects detection, removal of the manufacturing quality defects on additional repairing machines, and job re-treatment in part or all technological route. We consider this problem to illustrate the effect of ALMM switching methods for solving scheduling problems with supporting the ZDM strategy. There is more than one machine with a quality control in the technological route ( $1 < |\mathcal{M}_Q| \leq |\mathcal{M}|$ ) and more than one additional machine for repairing defective components ( $|\mathcal{M}_D| > 1$  and  $\mathcal{M}_D = \{M_{m+1}, \dots, M_{md}\}$ ). The choice of machine for repair depends on the degree or type of the defective components and it is the result of quality control. As a result of each quality control we receive information to which specific machine the task should be transferred.

We use the following notation to describe problem under consideration. There is a set of dedicated machines  $\mathcal{M} = \{M_1, M_2, \dots, M_m\}$  in the production route. The set of jobs  $\mathcal{J} = \{J_1, J_2, \dots, J_n\}$  must be processed by these machines. One job is equivalent to a batch of a few elements. A number of elements in the batch may be different for each task. Therefore, we assume that the jobs can be divided into smaller jobs. Each job  $j$  consists on  $m$  operations. The processing time of each operation  $i$  of job  $j$  is denote  $p_{i,j}$ . The processing time on the machine with quality control includes quality control time. All jobs follow the same route from the first machine to the last one  $(M_1, M_2, \dots, M_m)$ . Also, there is a store in front of each machine, where the job must wait when the machine is busy. There is also a store at the end of production line for finished jobs. The store before the first machine is the initial store. The final store contains only correctly (of adequate quality) completed jobs. Let  $\mathcal{W} = \{W_0, W_1, \dots, W_m\}$  mean a set of stores, where the store  $W_0$  is the final store and  $W_i$  is the store before machine  $M_i$ . We assume unlimited storage capacity. Let us denote by  $\mathcal{M}_D = \{M_{m+1}, \dots, M_{md}\}$  such that  $\mathcal{M}_D \cap \mathcal{M} = \emptyset$  set of additional machines used to repair faulty elements. There is



**Fig. 4** ALMM scheduling tool for achieving ZDM: with additional modules ALMM models modeller, ALLM switching method modeller, Database with previously used models and solutions, Predictive results interpreter module and Algorithm module.



**Fig. 5** Example of flow-shop technology route with stores, quality control and repairing machines. The production line includes two machines on which quality control is carried out and three additional machines for repairing defects. The first quality control machine detects one type of defects. This type of defects is repair on one additional repairing machine. The second quality control machine detects two different types of defect. This defects are fixed on two different additional repairing machines. The system provides for the detection of three different types of product defect and three different ways to eliminate them.

also a store in front of each repair machine. Let's denote the set of these stores by  $\mathcal{W}_D = \{W_{m+1}, \dots, W_{md}\}$ . After job repairing on the repair machine, the job returns to reprocessing to the specifically selected machine in the process line.

Figure 5 shows an example production line for such defined problem.

### 5.1 Models for subproblems

The first subproblem is the flow-shop problem with more than one quality control machine but without any

repair machine. The number of machines with quality control determines the number of types of quality defects, on one machine with quality control one type of quality defects is detected.

Let  $U_A$ ,  $S_A$ ,  $(s_0)_A$ ,  $f_A$ ,  $(S_N)_A$ ,  $(S_G)_A$  denote an adequate set of decisions, the set of proper states, the initial state, the transition function, the set of non-admissible states and the state of goal states in the algebraic-logical model of the flow-shop system with time limits  $ALM_A$ . A set of jobs denotes as  $\mathcal{J}_A$  and a set of machines denotes as  $\mathcal{M}$ .

**State of system** The generalized state of a process  $s_A = (x_A, t_A)$  at any instant  $t_A$  can be described by the state of machines in the technology route, state of the initial store and state of the work-in-progress store. Intentionally in the system state, it is not taken into account the information of the final store including completed jobs because it is redundant information (a set of completed jobs is calculated slightly lower).

Significant elements of the  $ALM_A$  for the switching method are the machines with quality control  $M_{q_\gamma}$ , where  $\gamma > 1$ , and additionally distinguishing many return machine.

Therefore, the generalized state of the process  $s_A = (x_A, t)$  at a given moment  $t$  is described as the state of machines from the route, the state of machines with quality control, the state of the initial warehouse and the state of inter-operational warehouses.

Thus, the proper state of the process is defined as a  $n$ -tuples:

$$x_A = \left( \underbrace{x_A^1, \dots, x_A^m}_{\text{warehouses state}}, \underbrace{x_A^{m+1}, \dots, x_A^{m+q_1}, \dots, x_A^{m+i}, \dots, x_A^{m+q_d}, \dots, x_A^{2m}}_{\substack{\text{machines states} \\ \text{with states of quality control machines}}} \right) \quad (3)$$

where:  $x_A^1$  - a state of the initial store,  $x_A^i$  - a state of the  $i$ -th work-in-progress store (between  $i-1$ -th and  $i$ -th machine) for  $i = 2, \dots, |\mathcal{M}|$ ,  $x_A^{|\mathcal{M}|+i}$  - a state of the  $i$ -th machine for  $i = 1, \dots, |\mathcal{M}|$ ,  $x_B^{m+q_\gamma}$  - a state of the  $i$ -th machine with quality control  $M_{q_i}$  where  $i \in \{1, \dots, d\}$ .

A coordinate  $x_A^1$  is a set of jobs to process by first machine, e.i. jobs which performing has not started yet. Particular coordinates  $x_A^i$ , where  $i = 2, \dots, |\mathcal{M}|$ , is a set of jobs which has been finished on the  $i-1$ -th machine and in the current state are available to process by  $i$ -th machine and are not assigned to this machine.

The structure of the machine state is as follows:  $x_A^i = (\beta_A, \tau_A)$  where:  $\beta_A \in \{0, 1, \dots, |\mathcal{J}_A|\}$  - job number, which is currently performing on the machine (0 denotes that no job is assigned to process by the machine),  $\tau_A \in [0, \tau_{idle}]$  - time remaining to complete current job ( $\tau_{idle}$  denotes, that the machine doesn't work).

The structure of the machine with quality control state is as follows:

$$x^{m+q} = (\beta, \tau, QC_1, \dots, QC_{md}, \hat{\beta})$$

where particular coordinates take the following:  $\beta_A \in \{0, 1, \dots, n\}$  - job number, which is currently performing on the machine (0 denotes that no job is assigned to process by the machine),  $\tau > 0$  -time remaining

to complete currant job ( $\tau_{idle}$  denotes, that the machine doesn't work),  $QC_\gamma \in [0, 100]$  - is the percentage of defect in the lot, where  $\gamma = 1, \dots, md$ , and  $md$  is the number of additional repairing machines,  $|\mathcal{M}_D| = md$ , (0 denotes positive result of quality control),  $\hat{\beta} \in \{0, 1, \dots, n\}$  - job number, in which defect were detected.

We can say that each  $i$ -th machine is not working (is free) in state  $s$  when machine state is  $x_A^{|\mathcal{M}|+i}(s) = (0, \tau_{idle})$ . Each  $i$ -th machine is working (is busy) in state when  $x_A^{|\mathcal{M}|+i}(s) = (\beta_A \neq 0, \tau_A > 0)$ . In fact, there are no states when the machine is working and job is not allocated to machine ( $\beta = 0$  and  $\tau > 0$ ) as well as machine is not working and has assigned the job the same time ( $\beta > 0$  and  $\tau = 0$ ). Thus, other combinations of the coordinate values of the machine state are not reflected in the real manufacturing process.

Let  $\beta_A^i$  and  $\tau_A^i$  denote the appropriate value for a particular  $i$ -th machine and  $x_A^i(s)$  denote the value of  $i$ -th coordinate in a particular state  $s_A$ .

In an initial state ( $(t_0)_A = 0$ ), the jobs are not performed and no job has been assigned to perform by any machine. Thus, the initial system state is as follows:

$$(x_0)_A = ((x_0)_A^1, \dots, (x_0)_A^{|\mathcal{M}|}, (x_0)_A^{|\mathcal{M}|+1}, \dots, (x_0)_A^{2|\mathcal{M}|})$$

where:

$$(x_0)_A^1 = \mathcal{J}_A$$

$$(x_0)_A^i = \emptyset \text{ for } i = 2, \dots, |\mathcal{M}|$$

$$(x_0)_A^i = (0, \tau_{idle}) \text{ for } i = |\mathcal{M}| + 1, \dots, 2|\mathcal{M}|.$$

We define a set of completed jobs  $(\mathcal{J}_F)_A$  in state  $s$ , which is equal to a difference of set of all jobs to perform and sets of jobs in stores with jobs processing by machines:  $(\mathcal{J}_F)_A(s) = \mathcal{J}_A \setminus \bigcup_{i=1}^{|\mathcal{V}|} (x_A^i(s) \cup \{\beta_A^{i+|\mathcal{M}|}(s)\})$ .

The set of not admissible generalized states  $S_N$  includes states in which job has not been finished and its due date has passed:

$$(S_N)_A = \{s_A : \exists_{j \in J} j \notin (\mathcal{J}_F)_A(s_A) \wedge d(j) < t_A\}.$$

A state  $s_A$  is a goal state where all the jobs have been performed and all job's deadlines are not overdue. Set of goal generalized states  $(S_G)_A$  is as follows:

$$(S_G)_A = \{s_A : (\mathcal{J}_F)_A(s_A) = \mathcal{J}_A \wedge \forall_{j \in J_A} d(j) \geq t_A\}.$$

**Decisions** In the problem, we assume that the decision consists of assigning jobs to individual machines at the same time. Particular co-ordinate  $u_A^i \in U_A^i$  represents separate decisions and refers to the  $i$ -th machine ( $i = 1, \dots, |\mathcal{M}|$ ).

The following assumptions have been proposed regarding the decision:

- the decision to allocate the next job for the machine can be taken if and only if the machine finished the previously assigned job and the job was performed by the previous machine in the technological route,
- the taken decision can not be changed.

The busy machine (realizing decision taken earlier) can only continue performing the previously assigned job. To the free machine in the current state, it can be assigned the job which satisfies the condition that it has been done by the previous machine in the technological route. When there will not be assigned any job to the free machine, the machine is still free.

Thus, the decision is a vector,  $u_A = (u_A^1, \dots, u_A^{|\mathcal{M}|})$  and value of particular co-ordinate  $u_A^i \in \mathcal{J}_A \cup \{0\}$  is as follows:

- if  $j$ -th job is assigned to the  $i$ -th machine, then the only possible decision is processing continuation  $u_A^i = 0$ ,
- if the  $i$ -th machine is free and there is no job that could be assigned i.e.  $x_A^i(s) = \emptyset$ , then the only possible decision is not assigning any task  $u_A^i = 0$ ,
- if the  $i$ -th machine is free and the  $i$ -th store is not empty (e.i. any  $j$ -th job from the store can be assigned to the machine), then the possible decision is assigned job  $u_A^i = j$  or no any job (still inactivity)  $u_A^i = 0$ .

The decision  $(u_p)_A$  must belong to the set of possible decision  $(U_p)_A$  in a given state  $s_A$ . Possible decision depends on the state of the particular machine. Possible decisions  $(u_p^i)_A(s)$  for the  $i$ -th machine in state  $s = (x, t)$  is defined as follows  $\forall_{i=1,2,\dots,|\mathcal{M}|}$ :

$$(u_p^i)_A(s) = \begin{cases} 0 & \text{for } x_A^{i+|\mathcal{M}|}(s) \neq (0, \tau_{idle}) \\ 0 & \text{for } x_A^{i+|\mathcal{M}|}(s) = (0, \tau_{idle}) \text{ if } x_A^i(s) = \emptyset \\ 0 \vee j & \text{for } x_A^{i+|\mathcal{M}|}(s) = (0, \tau_{idle}) \text{ if } j \in x_A^i(s) \end{cases} \quad (4)$$

We should note that it is not possible to assign one job to two different machines. Each set  $(U_p^i)_A$  is equal to the set of jobs in work-in-progress store plus the element 0, i.e.  $(U_p^i)_A = x_A^i(s) \cup \{0\}$ . When  $i$ -th machine is free and the  $i$ -th store is not empty it is possible to decide  $u_A^i = 0$ . That means the machine hasn't assigned any job to perform and waits. It gives us possibility for the machine to wait for the appearance of a new more perspective job. The profitability of that decision can be calculated by dedicated heuristic algorithm or predicted.

The complete definition of the set of the possible decision is as follows:  $(U_p)_A(s) = (U_p^1)_A(s) \times \dots \times (U_p^{|\mathcal{M}|})_A(s) \setminus H_A(s)$ , where  $H_A(s)$  is a set of decision not

assigning to any job  $u_A = (0, \dots, 0)$  when all the machines in a given state are free and not all jobs are completed.  $H_A(s) = \{u_A : (\forall_{i \in \mathcal{M}} : u_A^i(s) = 0 \wedge x_A^{i+|\mathcal{M}|}(s) = (0, \tau_{idle})) \wedge (\mathcal{J}_F)_A \neq \mathcal{J}_A\}$  Consecutively taken decisions form a decision sequence

$$\tilde{u}_A = ((u_1)_A, (u_2)_A, \dots, (u_c)_A),$$

where  $c$  is the number of taken decisions. This sequence uniquely determines the trajectory of process.

**Transition function  $f_A$ .** The transition function is defined by means of two functions  $f_A = ((f_x)_A, (f_t)_A)$ , where  $(f_x)_A$  determines the next proper state and  $(f_t)_A$  determines the next time instant. Firstly, it is necessary to determine the moment when the subsequent state occurs  $t'_A = t_A + \Delta t_A$ .  $\Delta t_A$  equals the lowest value of time needed to complete job processing by  $i$ -th machine.

Once the moment  $t'_A$  is known, it is possible to determine the proper state of the process at that time (determining  $(f_x)_A$ ). The coordinate  $x_A^1$  (state of initial store) is reduced by job, which is assigned to the first machine:  $(x_A^1(s))' = x_A^1(s) \setminus \{j : u_A^1(s) = j\}$ . Coordinate  $x_A^i$  for  $i = 2, \dots, |\mathcal{M}|$  (state of store between  $i-1$ -th and  $i$ -th machine), is increased by the job performed on  $(i-1)$ -th machine and is reduced by the job, which is assigned to  $i$ -th machine in the technological route:

$$(x_A^i(s))' = x^i(s) \cup (\{j : x_A^{i-1+|\mathcal{M}|}(s) = (j, \Delta t_A)\} \setminus \{j : u_A^i(s) = j\}). \quad (5)$$

Value of coordinates related to machine state

$$(x_A^{i+|\mathcal{M}|})' = (\beta'_A, \tau'_A)$$

for  $i = 1, \dots, |\mathcal{M}|$ , depends on the taken decision and is as follows:

1. if  $i$ -th machine is free in given state machine and decision is  $u_A^i(s) = j$  (assigning  $j$ -th job to perform by this machine):

$$\beta'_A = \begin{cases} 0 & \text{for } \tau_A^{i+|\mathcal{M}|} = \Delta t_A \\ j & \text{for } \tau_A^{i+|\mathcal{M}|} > \Delta t_A \end{cases} \quad (6)$$

$$\tau'_A = \begin{cases} \tau_{idle} & \text{for } \tau_A^{i+|\mathcal{M}|} = \Delta t_A \\ p^{ij} - \Delta t_A & \text{for } \tau_A^{i+|\mathcal{M}|} > \Delta t_A \end{cases}$$

2. if  $i$ -th machine is free in given state and decision is  $u_A^i(s) = 0$  (not assigning job to perform by this machine):  $\beta'_A = 0, \tau'_A = \tau_{idle}$ ,
3. if  $i$ -th machine is busy in given state and decision is  $u_A^i(s) = 0$  (continuation previously assigned job  $j$ ):

$$\beta'_A = \begin{cases} 0 & \text{for } \tau_A^{i+|\mathcal{M}|} = \Delta t_A \\ j & \text{for } \tau_A^{i+|\mathcal{M}|} > \Delta t_A \end{cases} \quad (7)$$

$$\tau'_A = \begin{cases} \tau_{idle} & \text{for } \tau_A^{i+|\mathcal{M}|} = \Delta t_A \\ \tau_A^{i+|\mathcal{M}|} - \Delta t_A & \text{for } \tau_A^{i+|\mathcal{M}|} > \Delta t_A \end{cases}$$

The  $x^{q+m} = (\beta, \tau, QC, \hat{\beta})$  coordinate representing the state of the machine with the quality control is also carried out to the new state  $(x^{q+m})' = (\beta', \tau', QC', \hat{\beta}')$ . The new state depends on the decision made and the quality control result. The quality control value is provided at the time when the job is completed. Let  $QC(j)$  and  $\hat{\beta}(j)$  represent the quality control result for the job  $J_j$ , respectively. Therefore, the transition function for the machine state with quality control is as follows:

1. if the machine is free and the quality control is positive ( $x^{q+m} = (0, \tau_{idle}, 0, \hat{\beta})$ ) in a given state and the decision  $u^q(s) = j$  (assigning  $j$ -th job to perform by this machine), then:

- i for  $\tau^{q+m} = \Delta t$ , in the next state, the machine with QC will be free and the new quality control result will be read:

$$(x^{q+m})' = (0, \tau_{idle}, QC(j), \hat{\beta}(j)), \quad (8)$$

- ii otherwise when  $\tau^{q+m} > \Delta t$ , in the next state the machine still processes the  $j$  job and the time to complete the job is  $p^{qj} - \Delta t$  and the quality control result is the same as in the previous state:

$$(x^{q+m})' = (j, p^{qj} - \Delta t, 0, \hat{\beta}). \quad (9)$$

2. if  $q$ -th machine is free and quality control result is positive  $x^{q+m} = (0, \tau_{idle}, 0, \hat{\beta})$  in given state and decision is  $u^q(s) = 0$  (not assigning job to perform by this machine), then:  $(x^{q+m})' = x^{q+m}$

3. if  $q$ -th machine is busy  $x^{q+m} = (j, \tau, QC, \hat{\beta})$  in a given state and the only possible decision is  $u^q(s) = 0$  to continue processing the previously assigned job  $j$ , then:

- i for  $\tau^{q+m} = \Delta t$ , similarly to the case of 1 in the next state, the machine with quality control is free and a new quality control result will be read, i.e. the next state is given by the formula (8),

- ii otherwise, when  $\tau^{q+m} > \Delta t$ , in the next state the machine continues job  $j$  processing, the time to complete the job is  $\tau - \Delta t$  and the control result is the same as the previous:

$$(x^{q+m})' = (j, \tau - \Delta t, QC, \hat{\beta}). \quad (10)$$

That should be noted that if machine with quality control is free and the result of the control is negative, the next state is not calculated using the transition function but using a switching function. Then we switch models from  $ALM_A$  to  $ALM_B$ .

The second subproblem is represented by  $ALM_B$  and it consists not only states of machines in the technological route, states of machines with quality control but also states of additional repairing machines. Let us

consider a special case in which there are quality deficiencies of all types. Then all the additional repairing machines  $M_{m+1}, \dots, M_{md}$  from outside the technological route are in the second subproblem. The model of this problem is a modification of the model  $ALM_A$  by adding elements related to repair of defective jobs, i.e. repair machines. The proper state  $x_B$  in such system is as follows:

$$x_B = \underbrace{(x_B^1, \dots, x_B^m)}_{\text{store states}}, \underbrace{(x_B^{m+1}, \dots, x_B^{m+q_1}, \dots, x_B^{m+q_d}, \dots, x_B^{2m})}_{\text{machine states with quality control machines}}, \underbrace{(x_B^{2m+1}, x_B^{2m+2}, \dots)}_{\text{store state and repairing machine type 1}}, \dots, \underbrace{(x_B^{2m+2md-1}, x_B^{2m+2md})}_{\text{store state and repairing machine type md}} \quad (11)$$

where:

$x_B^{m+q_\gamma}$  - state of machine with quality control  $M_{q_\gamma}$  for  $\gamma$  type of defects, where  $\gamma \in \{1, \dots, d\}$

$x_B^{2m+2\gamma-1}$  - state of additional store  $W_{m+\gamma}$  for jobs with  $\gamma$  type of defects, where  $\gamma = 1, \dots, md$ ,

$x_B^{2m+2\gamma}$  - state of additional repairing machine  $M_{m+\gamma}$  for jobs with  $\gamma$  type of defects, where  $\gamma = 1, \dots, md$ .

## 5.2 The switching function

The switching function between the models includes three types of switching. It is defined as follows:

- i Switching from the  $ALM_A$  to the  $ALM_B$  in the case of detecting at least one quality defect of any type (negative result of quality control of at least one machine with quality control), when there is no job with previously detected quality deficiencies,
- ii Switching from the  $ALM_B$  model to the model with the same structure only with changed parameters, i.e. the  $ALM'_B$  structure in cases where:
  - at least one machine with quality control has detected another job with quality defect,
  - some previously detected job has been repaired and others previously detected jobs are still waiting to repair.
- iii Switching the  $ALM_B$  to the  $ALM_A$  structure with changed parameters, that is  $ALM'_A$ , when the last of the defected job is repaired by any repair machine and has just been fixed.

### 5.3 The switching states

The set of switching states is defined as the set of states in which (a) any machine with quality control  $M_{qk}$  has just finished job and the value of quality control is negative (defect has been just detected) or (b) any repairing machine has just finished job  $M_\gamma$ , where  $\gamma = m + 1, \dots, m + md$  and it is necessarily to put the job which has been just repaired to the machine in the technology rough:

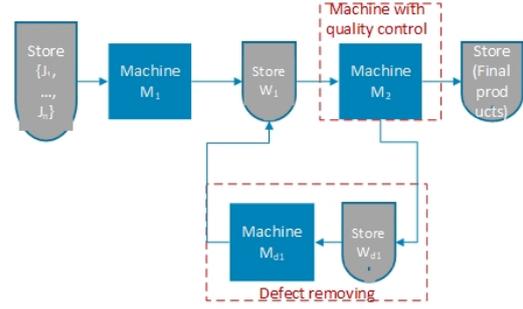
$$\begin{aligned} \mathcal{S}_{switch} = \{ & (x, t) : \\ & \exists_{k \in \{1, \dots, d\}} x^{m+q_k} = (0, \tau_{idle}, QC \neq 0, \hat{\beta} \neq 0) \\ & \vee \\ & \exists_{\gamma \in \{1, \dots, md\}} x^{2m+2\gamma-1} = (0, \tau_{idle}) \} \end{aligned} \quad (12)$$

### 5.4 The switching rules

Switching rules  $\mathcal{R}_{switch}$  determine how to calculate the model parameters during switching. Additionally, if any machine with quality control has just finished any job processing with negative quality control then we should indicate the returning place of this job in the technological route, taking into account the appropriate repairing machine and type of quality defect. Therefore, when dedicated machine  $M_{q_\gamma}$  detects a new  $\gamma$  defect, the rule is to determine the changes of parameters for the repair machine dedicated to a given type of defect. The appearance of a quality deficiency of the  $\gamma$  type does not change the state of other repairing machines dedicated to other types of quality deficiencies. Completion of a quality defect repairing of  $\gamma$  type by the repair machine  $M_{m+\gamma}$  requires recalculation of the store state  $W_{r\gamma}$  before the machine  $M_{r\gamma}$ . The states of return store dedicated to other defect types do not change.

## 6 Discussion

In the paper we consider the approach combining several process disturbances which are “product-oriented” and aligned with the ZDM concept. In particular, we consider a detection of a quality defects in manufactured parts as “product-oriented” disturbances. However, it is debatable whether or when damaged product should be repair. When to produce additional parts instead of repair? Many factors may impact on this decision, such as the cost of repair or the availability of additional raw materials. Undoubtedly, one of such factors is the repairing time, especially when contracts include the required completion dates.



**Fig. 6** Example of flow-shop technology route with stores, quality control, one type of product defect and one repairing machine.

Let consider the special case of a flow-shop system with two machines in a technological route  $M_1$  and  $M_2$ , including one machine with quality control (the second machine in the technological route), and one additional repairing machine  $M_{d1}$ . When a fault is detected, the damaged parts are placed in the store  $W_{d1}$ , and then repaired on an additional machine  $M_{d1}$ . After repairing on additional repairing machine, the job is placed in store  $W_1$  to reprocess on the second machine again. Figure 6 presents an example of this production system. In the switching method, two models should be developed. The first is  $ALM_A$  for two machines in the technological route and second is  $ALM_B$  with additional repair machine. Switching occurs when the quality control machine detects a quality deficiency, or the repair machine has just finished repairing the deficient job and there are no more deficient jobs in the system.

### 6.1 Example 1

Let consider three jobs  $J_1$ ,  $J_2$  and  $J_3$  in the above flow-shop system. The processing time on the machines is given in the table 2:

**Table 2** First problem example, with two machines  $M_1$ ,  $M_2$ , one additional repairing machine  $M_{d1}$  and three jobs  $J_1$ ,  $J_2$  and  $J_3$  and respectable processing times

	$J_1$	$J_2$	$J_3$
$M_1$	2	10	15
$M_2$	5	2	2
$M_{d1}$	5	2	2

Figure 7 presents optimal schedule without deficient jobs for this problem. Total processing time is equal 29. According to the approach described in section FFS, the problem is modeled using the  $ALM_A$  with

two machines in a technological route  $M_1$  and  $M_2$  and three stores.



**Fig. 7** Example of optimal schedule in flow-shop system without disturbances for two machine and three jobs.

Let us consider the case in which the quality control machine gives a negative quality result for the first job  $J_1$ . That is, quality deficiencies have been detected only in the first job. In this case, the quality control takes place after the completion of the job by the second machine  $M_2$ , and the repair job requires the use of additional treatment on the repairing machine  $M_{d1}$  and re-treatment on the second machine  $M_2$ . At this point, a switch between models occurs and the problem is modeled using the  $ALM_B$ , which includes an additional repairing machine  $M_{d1}$ . Due to the need to repair the first job  $J_1$  the schedule is changing. When the time is 7, the second machine  $M_2$  has finished the first job  $M_1$  with negative quality control result. For this reason, an additional repairing machine  $M_{d1}$  has to be considered in the system and the first job  $M_1$  must be assigned to the additional repairing machine  $M_{d1}$  to be processed. At this time the first machine  $M_1$  is in the process of the second job  $J_2$  and there are 5 units of time left to complete this job (which is presented in a lighter color in the figure 8). When the time is 12, the repairing machine  $M_{d1}$  has finished processing of the first job  $J_1$  and no new quality defects have been detected. Hence the system does not need to consider the repairing machine  $M_{d1}$  any longer. The repaired first job  $J_1$  has to be assigned to re-processing on the second machine  $M_2$ . In 12 unit of time in the store  $W_2$  are both  $J_1$  and  $J_2$  jobs. The optimization algorithm chooses the order of their processing on second machine. The figure 8 shows the situation, when the job  $J_1$  is executed first on the second machine, and then job  $J_2$ . In this example, we do not consider any more quality defects. At this point, there is a switch between the models and the problem is modeled using the  $ALM_A$  without an additional repairing machine  $M_{d1}$ . In 29 unit of time the all jobs are finished with adequate quality. It should be note that despite the need of repairing first job  $J_1$ , the total processing time  $C_{max}$  does not change in relation to the total processing time in the case of no quality deficiencies.

## 6.2 Example 2

Let us consider the same production system with another example of processing time (table 3).

**Table 3** Second problem example, with two machines  $M_1$ ,  $M_2$  and three jobs  $J_1$ ,  $J_2$  and  $J_3$  and respectable processing times. The processing time of jobs  $J_1$ , and  $J_2$  on second machine is significantly longer

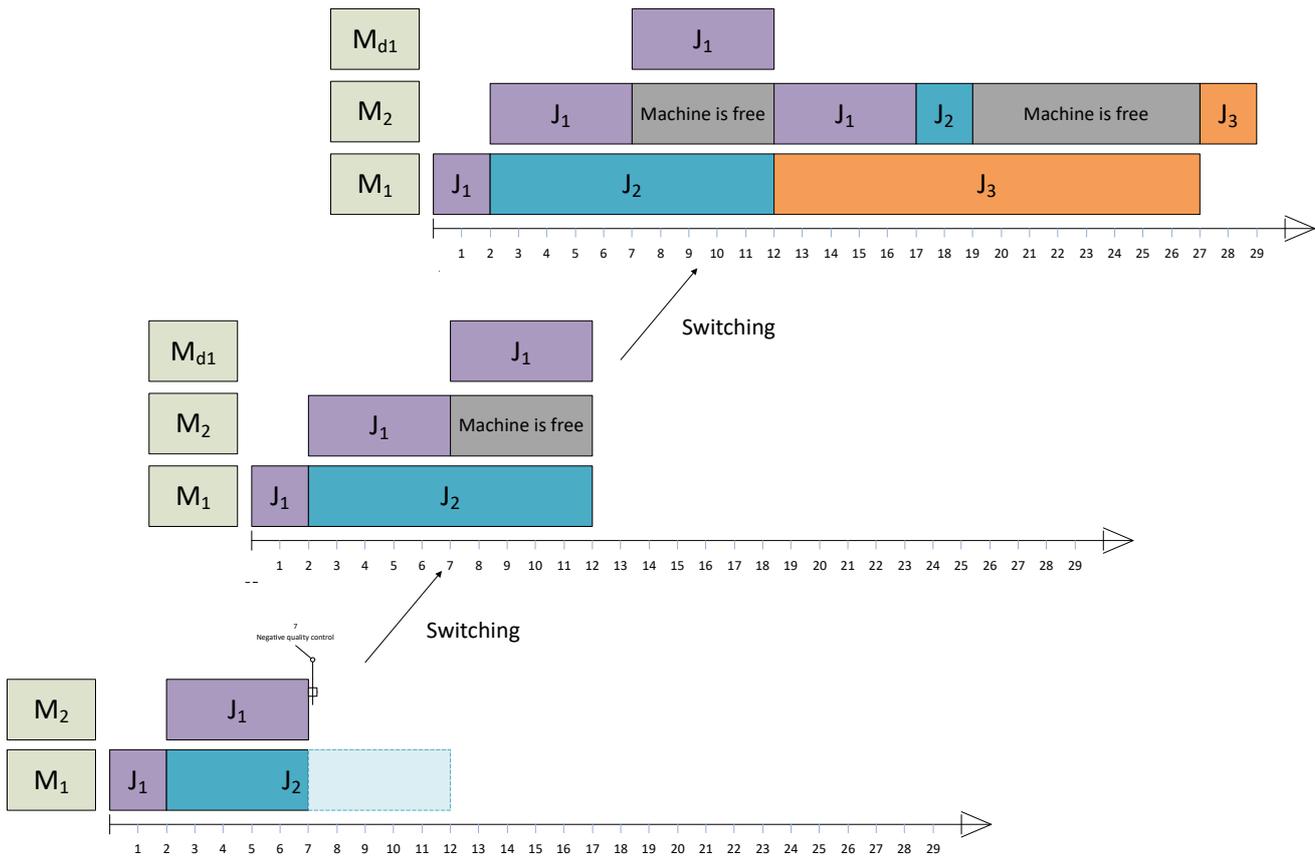
	$J_1$	$J_2$	$J_3$
$M_1$	2	10	15
$M_2$	9	7	2
$M_{d1}$	9	7	2

Let us consider again the situation when quality control is on  $M_2$  machine and gives negative quality control for  $J_1$ . Then this job has to be repaired on an additional machine and reprocessed on machine  $M_2$ . An example of this schedule is presented in figure 8.

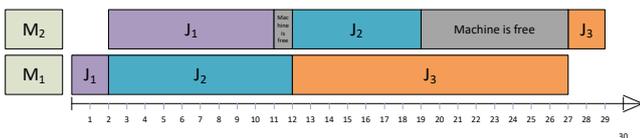
When the time is 10, the second machine  $M_2$  has finished the first job  $J_1$  with negative quality control result. For this reason, an additional repairing machine  $M_{d1}$  has to be considered in the system and the first job  $J_1$  has to be assigned to the additional repairing machine  $M_{d1}$  to be processed. At this point, a switch between models occurs and the problem is modeled using  $ALM_B$ , which includes an additional repairing machine  $M_{d1}$ . At this time, the first machine  $M_1$  processes the second job  $J_2$  and there are 2 units of time left to complete this job (which is presented in a lighter color in figure 8). When the time is 20, the repairing machine  $M_{d1}$  has just finished processing of the first job  $J_1$  and no new quality defects have been detected. Hence, the system does not need to consider repairing machine  $M_{d1}$  any longer and the repaired first job  $J_1$  has to be assigned to re-processing on the second machine  $M_2$ . At this point, there is a switch between the models and the problem is modeled using  $ALM_A$  without an additional repairing machine  $M_{d1}$ . In this example, we do not consider any more quality defects. In 31 unit of time, all jobs are finished with adequate quality.

In this case, the total processing time  $C_{max}$  is longer than 29. The schedule would be shorter if we dropped the first task and pick up new materials to start producing the job once again. The algorithm should calculate a part of the solution related to the repair of defects or reproduction and provide a more admissible solution depending on the optimization criterion.

It is also possible that then after reprocessing on the second machine,  $M_2$  quality control gives negative quality control for  $J_1$  one again and again. In such cases, in the proposed framework in the database, all previous



**Fig. 8** Example of the schedule of the flow-shop system with defect detection after processing on the second machine, defect repairing on one additional machine, and then re-processing on the second machine. The schedule presents defect detection for the first job, then the first job is repairing to eliminating defects and then re-processing in the second machine. Switching function enables processing jobs with defects differently than it would result from the technological route.



**Fig. 9** Example of optimal schedule in flow-shop system without disturbances for two machine and three jobs with parameters from the table 2.

calculated solutions and models are stored so the predictive results interpreter module can give the information to the Decision Support System. It is not profitable to repair the defects in this case and a better solution is to take new material (if available) from the warehouse and perform the job again from the beginning.

**7 Conclusions**

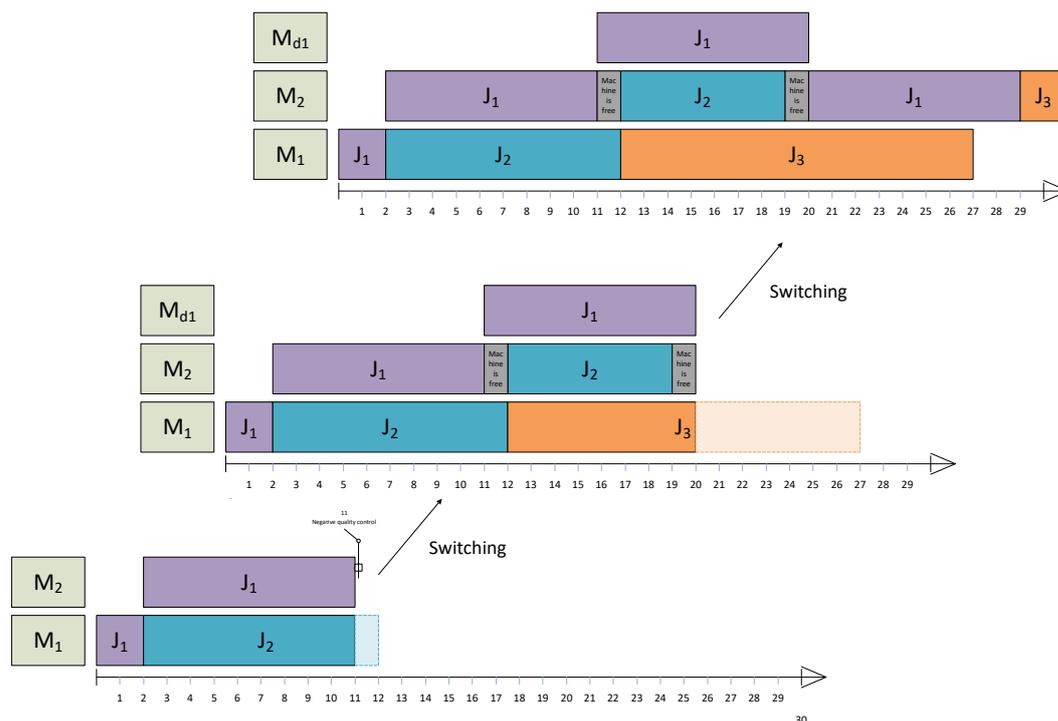
In this paper we have considered the problem of ensuring Zero Defect Manufacturing requirements through formal methods. We have proposed an adaptation of Algebraic-Logical Meta Model for an uncertain produc-

tion environment through appropriate model switching. The advantage of our approach comes from filling the gap in typical ZDM strategies by handling “product-oriented” disturbances. Additional benefit is a possibility of integrating the meta heuristic algorithms through predictive-reactive scheduling.

The main drawback of our approach is the fact that current form is not supported by any computational framework. While methodology is formally complete and allows representing large variety of problems, there is no tool allowing for efficient model creation or optimization. This will be a topic of further research. Moreover the investigation of large scale extensions are still far from complete. In our future works we want to investigate the aspect of scaling of switching ALMMs in terms of computational complexity.

**Declarations**

**Funding**



**Fig. 10** Example of a schedule of the flow-shop system with defect detection after processing on the second machine, defect repairing on one additional machine, and re-processing on the second machine. The schedule presents defect detection for the first job, then the first job is repairing to eliminating defects and re-process on the second machine. Switching function enables processing jobs with defects differently than it would result from technological route.

Work funded by AGH University of Science and Technology subvention for scientific research

#### Conflicts of interest/Competing interests

Not applicable

#### Availability of data and material (data transparency)

Not applicable

#### Code availability (software application or custom code)

Not applicable

#### References

- Abedinnia H, Glock CH, Schneider MD (2017) Machine scheduling in production: A content analysis. *Applied Mathematical Modelling* 50:279 – 299
- Barbosa J, Leitão P, Ferreira A, Queiroz J, Gerales CA, Coelho JP (2018) Implementation of a multi-agent system to support zdm strategies in multi-stage environments. In: 2018 IEEE 16th International Conference on Industrial Informatics (INDIN), IEEE, pp 822–827
- Bufardi A, Akten O, Arif M, Xirouchakis P, Perez R (2017) Towards zero-defect manufacturing with a

combined online - offline fuzzy-nets approach in wire electrical discharge machining. *WSEAS Transactions on Environment and Development* 13:401–409

- Cardin O, Trentesaux D, Thomas A, Castagna P, Berger T, El-Haouzi HB (2017) Coupling predictive scheduling and reactive control in manufacturing hybrid control architectures: state of the art and future challenges. *Journal of Intelligent Manufacturing* 28(7):1503–1517

- Chien CF, Hsu SC, Chen YJ (2013) A system for on-line detection and classification of wafer bin map defect patterns for manufacturing intelligence. *International Journal of Production Research* 51(8):2324–2338, DOI 10.1080/00207543.2012.737943

- Dias LS, Ierapetritou MG (2016) Integration of scheduling and control under uncertainties: Review and challenges. *Chemical Engineering Research and Design* 116:98 – 113, process Systems Engineering - A Celebration in Professor Roger Sargent's 90th Year

- Dreyfus PA, Kyritsis D (2018) A framework based on predictive maintenance, zero-defect manufacturing and scheduling under uncertainty tools, to optimize production capacities of high-end quality products. In: Moon I, Lee GM, Park J, Kiritsis D, von Cieminski G (eds) *Advances in Production Management Systems. Smart Manufacturing for Indus-*

- try 4.0, Springer International Publishing, Cham, pp 296–303
- Dudek-Dyduch E (1989) Formal bases of classification of discrete production processes control problems, *zeszyty naukowe agh nr 1313*. *Automatyka* 49(8):189–200
- Dudek-Dyduch E (1990) Formalizacja i analiza problematyki dyskretnych procesów produkcyjnych. *Zeszyty Naukowe AGH, Automatyka* 54
- Dudek-Dyduch E (2000) Learning-based algorithms in scheduling. *Journal of Intelligent Manufacturing* 11(2):135–143
- Dudek-Dyduch E, Dutkiewicz L (2013) Substitution tasks method for discrete optimization. In: *International Conference on Artificial Intelligence and Soft Computing*, Springer, pp 419–430
- Dudek-Dyduch E, Kucharska E (2011) Learning method for co-operation. In: *International Conference on Computational Collective Intelligence*, Springer, pp 290–300
- Dudek-Dyduch E, Kucharska E, Dutkiewicz L, Rączka K (2014) Almm solver - a tool for optimization problems. In: Rutkowski L, Korytkowski M, Scherer R, Tadeusiewicz R, Zadeh LA, Zurada JM (eds) *Artificial Intelligence and Soft Computing*, Springer International Publishing, Cham, pp 328–338
- Grobler-Dębska K, Kucharska E, Dudek-Dyduch E (2013) Idea of switching algebraic-logical models in flow-shop scheduling problem with defects. In: *The 18th International Conference on Methods and Models in Automation and Robotics MMAR proceedings*, pp 532–537
- Grobler-Dębska K (2019) Methodology of switching algebraic-logical models for discrete manufacturing with defects. PhD thesis, AGH University of Science and Technology, Department of Automatics and Robotics
- Ju F, Li J, Xiao G, Arinez J (2013) Quality flow model in automotive paint shops. *International Journal of Production Research* 51(21):6470–6483
- Konrad B, Lieber D, Deuse J (2013) Striving for zero defect production: Intelligent manufacturing control through data mining in continuous rolling mill processes. In: Windt K (ed) *Robust Manufacturing Control*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 215–229
- Kucharska E (2017) Heuristic method for decision-making in common scheduling problems. *Applied Sciences* 7(10):1073
- Kucharska E (2019) Dynamic vehicle routing problem—predictive and unexpected customer availability. *Symmetry* 11(4):546
- Kucharska E, Dudek-Dyduch E (2014) Extended learning method for designation of co-operation. In: *Transactions on computational collective intelligence XIV*, Springer, pp 136–157
- Kucharska E, Rączka K (2018) Almm solver - idea of algorithm module. In: Kozielski S, Mrozek D, Kasprowski P, Małysiak-Mrozek B, Kostrzewa D (eds) *Beyond Databases, Architectures and Structures. Facing the Challenges of Data Proliferation and Growing Variety*, Springer International Publishing, Cham, pp 467–479
- Kucharska E, Grobler-Dębska K, Rączka K (2017a) Algebraic-logical meta-model based approach for scheduling manufacturing problem with defects removal. *Advances in Mechanical Engineering* 9(4):1–18
- Kucharska E, Grobler-Dębska K, Rączka K (2017b) Almm-based methods for optimization makespan flow-shop problem with defects. In: *Information Systems Architecture and Technology: Proceedings of 37th International Conference on Information Systems Architecture and Technology - ISAT 2016 - Part I*, pp 41–53
- Kuo CF, Hsu CTM, Fang CH, Chao SM, Lin YD (2013) Automatic defect inspection system of colour filters using taguchi-based neural network. *International Journal of Production Research* 51(5):1464–1476
- Lindström J, Lejon E, Kyösti P, Mecella M, Heutelbeck D, Hemmje M, Sjö Dahl M, Birk W, Gunnarsson B (2019) Towards intelligent and sustainable production systems with a zero-defect manufacturing approach in an industry4.0 context. *Procedia CIRP* 81:880 – 885, 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019
- Liu L (2019) Outsourcing and rescheduling for a two-machine flow shop with the disruption of new arriving jobs: A hybrid variable neighborhood search algorithm. *Computers & Industrial Engineering* 130:198 – 221
- Myklebust O (2013) Zero defect manufacturing: A product and plant oriented lifecycle approach. *Procedia CIRP* 12:246 – 251, eighth CIRP Conference on Intelligent Computation in Manufacturing Engineering
- Psarommatis F, Kiritsis D (2018) A scheduling tool for achieving zero defect manufacturing (zdm): A conceptual framework. In: Moon I, Lee GM, Park J, Kiritsis D, von Cieminski G (eds) *Advances in Production Management Systems. Smart Manufacturing for Industry 4.0*, Springer International Publishing, Cham, pp 271–278

- Psarommatis F, May G, Dreyfus PA, Kiritsis D (2020) Zero defect manufacturing: state-of-the-art review, shortcomings and future directions in research. *International Journal of Production Research* 58(1):1–17
- Raczka K, Kucharska E (2017) Almm solver - database structure and data access layer architecture. In: Kozielski S, Mrozek D, Kasprowski P, Małysiak-Mrozek B, Kostrzewa D (eds) *Beyond Databases, Architectures and Structures. Towards Efficient Solutions for Data Analysis and Knowledge Representation*, Springer International Publishing, Cham, pp 551–563
- Shen J, Zhu Y (2017) Uncertain flexible flow shop scheduling problem subject to breakdowns. *Journal of Intelligent & Fuzzy Systems* 32(1):207–214
- Teti R (2015) Advanced it methods of signal processing and decision making for zero defect manufacturing in machining. *Procedia CIRP* 28:3 – 15, 3rd CIRP Global Web Conference - Production Engineering Research
- Wang KS (2013) Towards zero-defect manufacturing (zdm)—a data mining approach. *Advances in Manufacturing* 1(1):62–74

# Figures

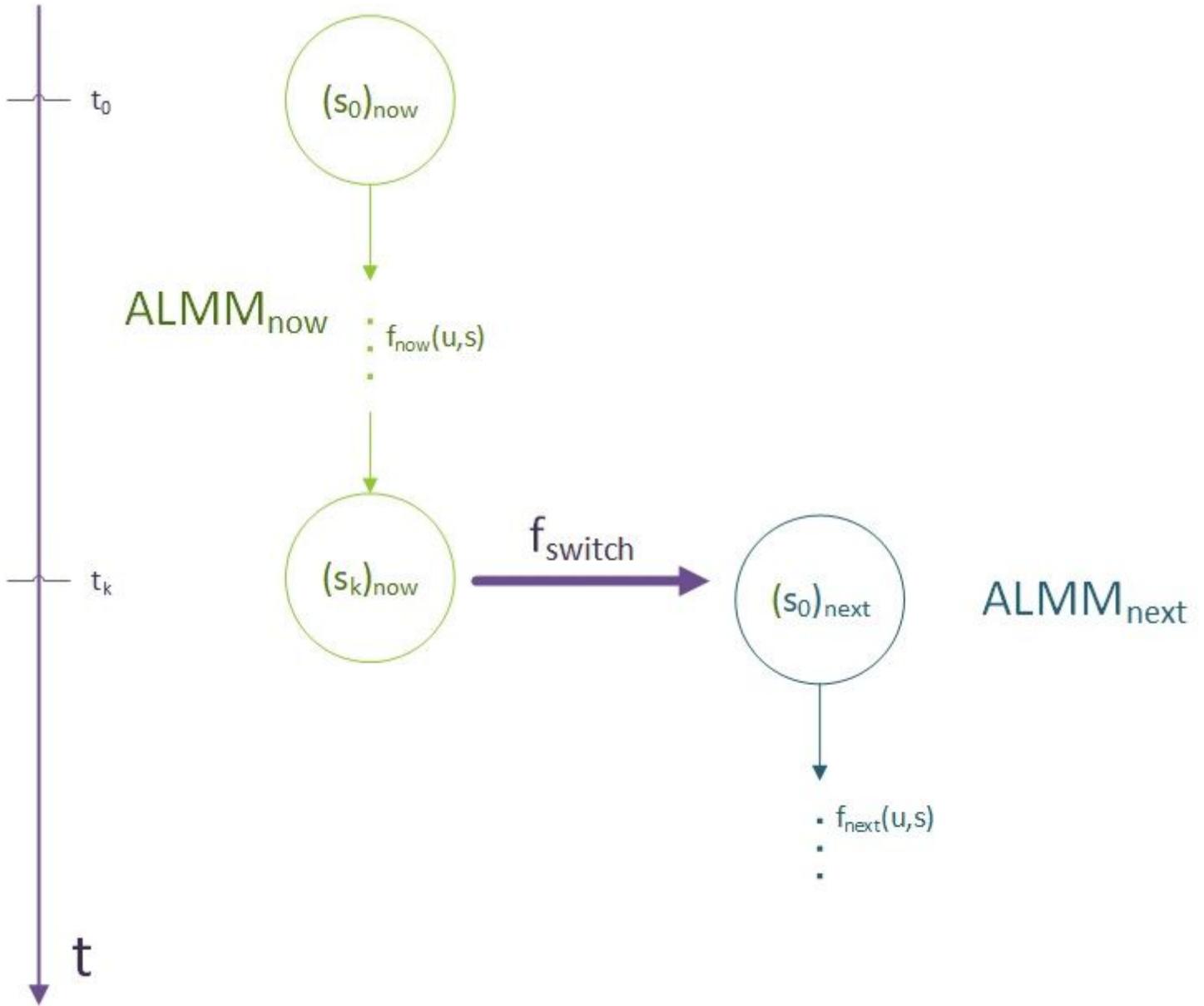
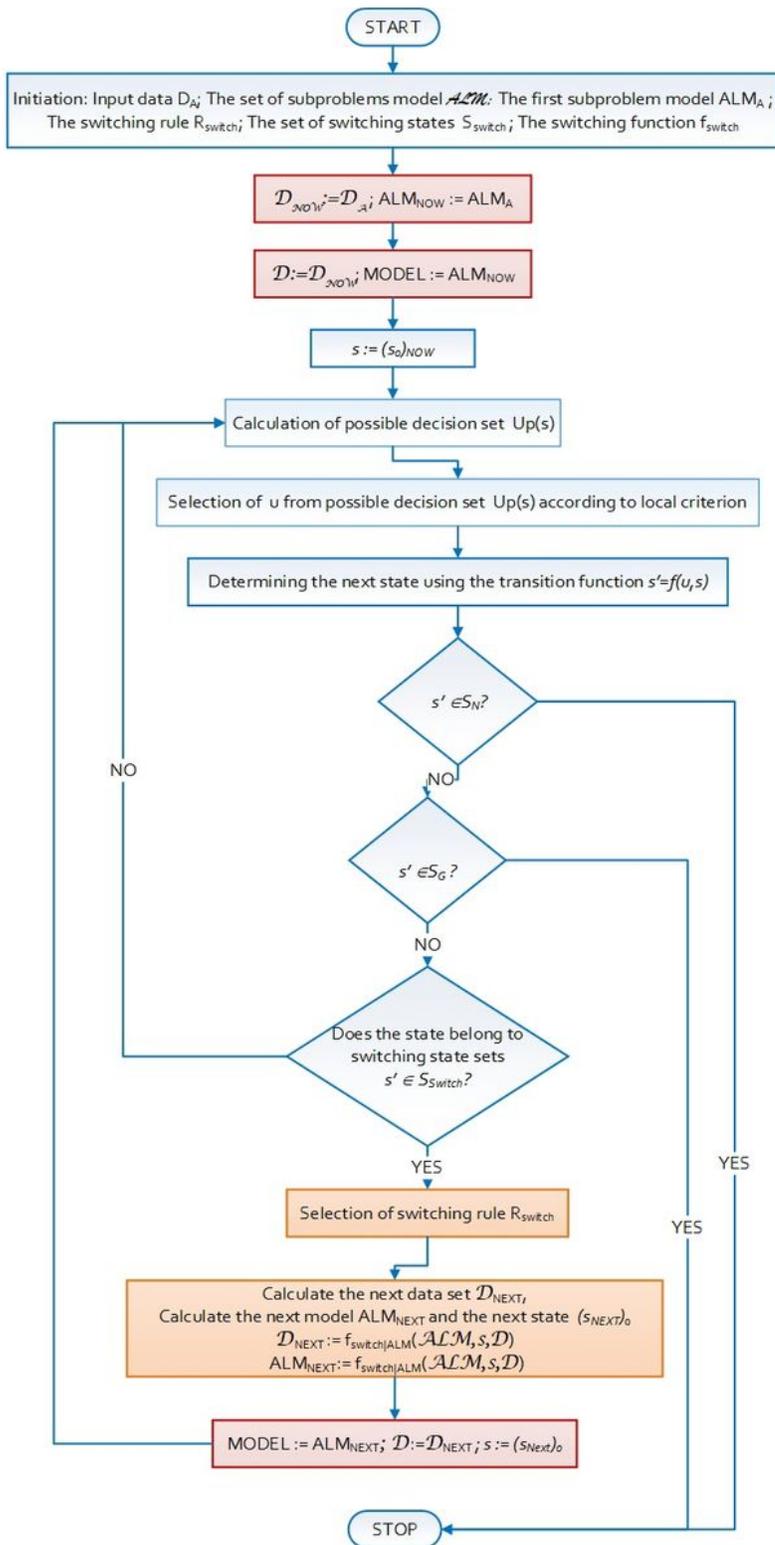


Figure 1

The idea of generating a single process trajectory with switching between models. Switching occurs as a result of an event in the current state of the production process (represented by  $s_k$  of the current model  $\text{ALM}_{\text{now}}$ ) and goes to the next state of the process (represented as the initial state  $s_0$  usually in another model  $\text{ALM}_{\text{next}}$ .)



**Figure 2**

The general algorithm of the switching method for scheduling problems of production with quality deficiencies. In contrast to the classic algorithm for generating trajectories using ALMM, this algorithm does not only check whether or not the calculated state belongs to the goal or to the not admissible states but also check belonging to the switching states. Also, it specifies the conditions for the ALMM model changes.

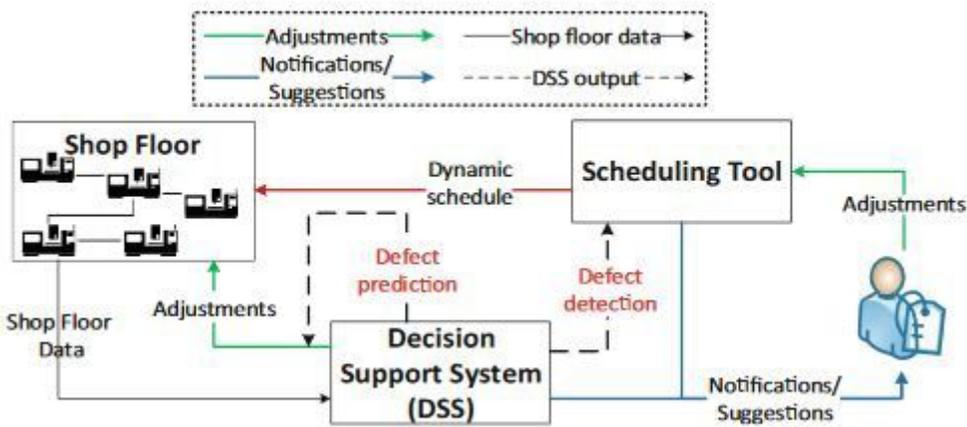


Figure 3

Scheduling tool for achieving ZDM proposed in the Psarommatis and Kiritsis (2018). The framework consists the integrated modules of Decision Support System and Scheduling Tool.

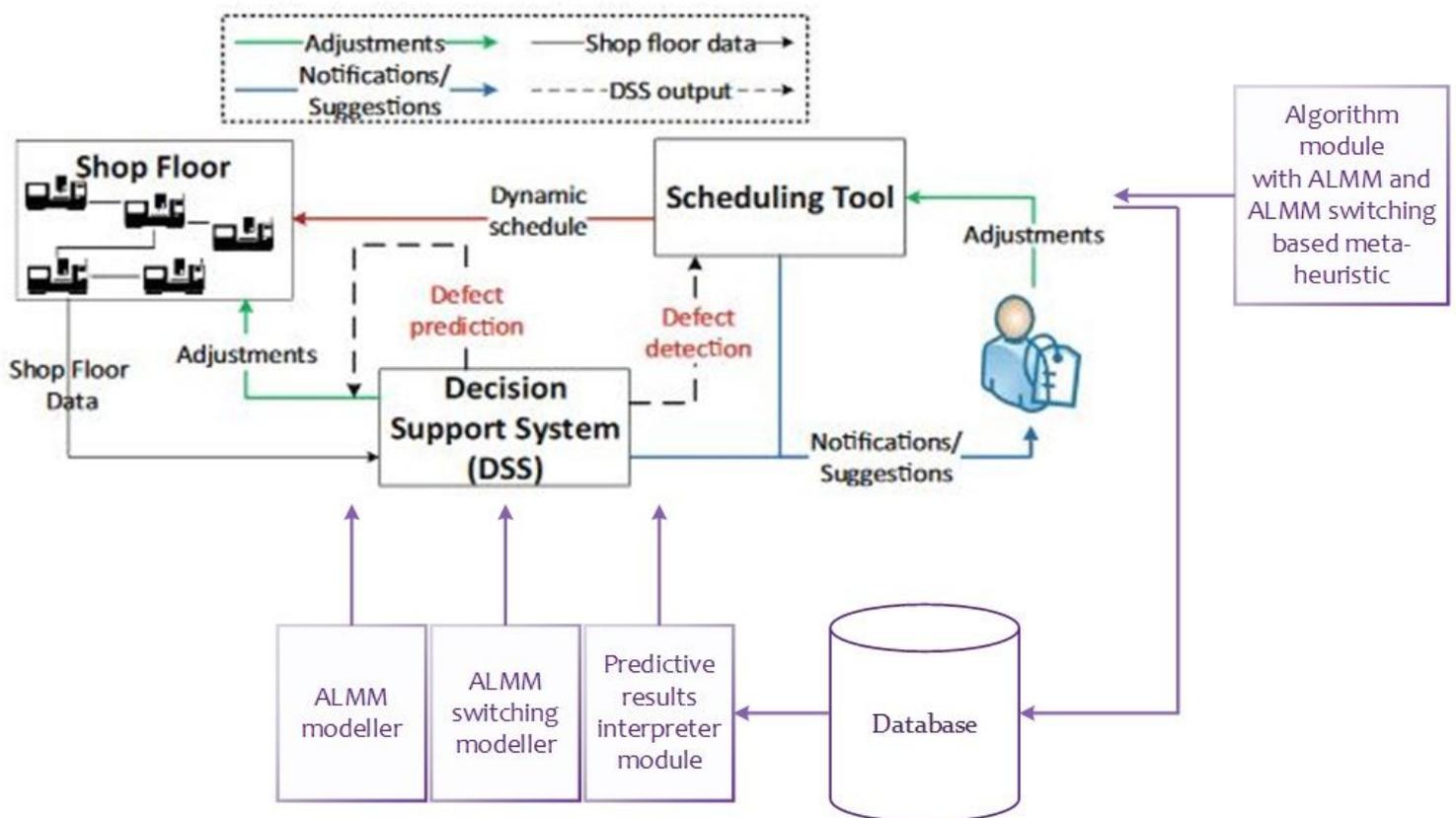
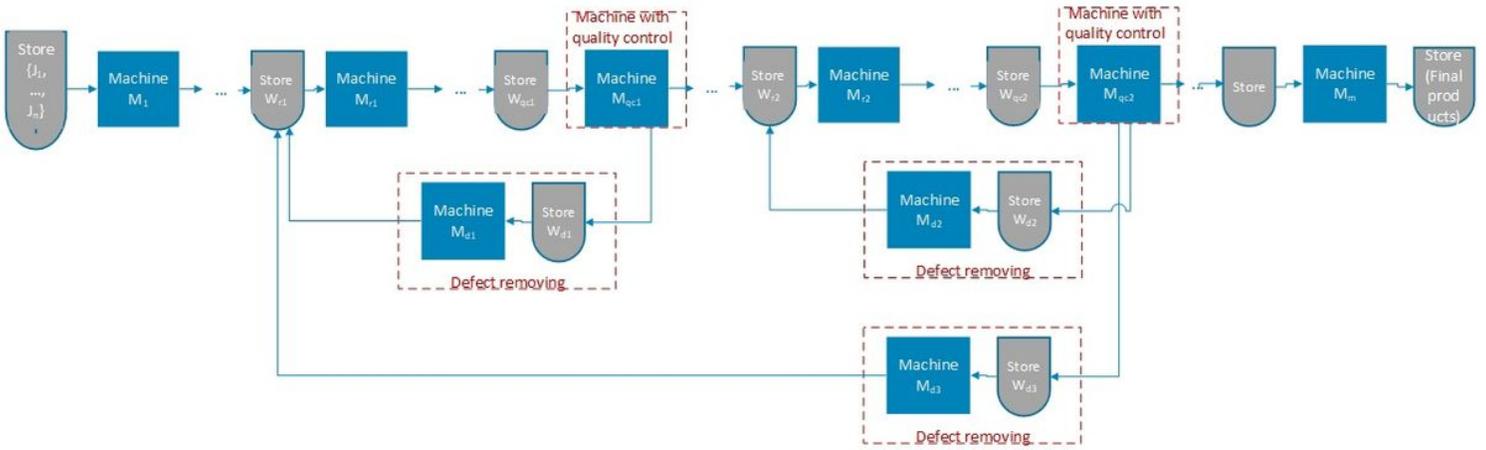


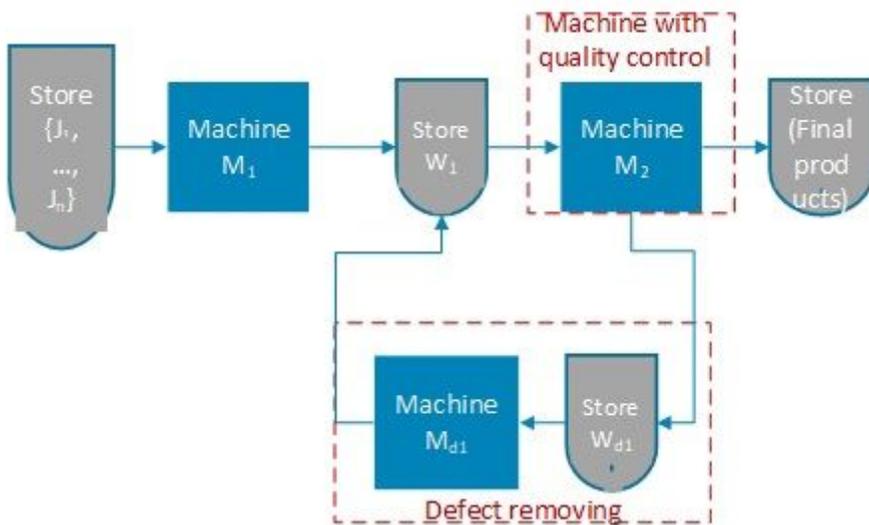
Figure 4

ALMM scheduling tool for achieving ZDM: with additional modules ALMM models modeller, ALLM switching method modeler, Database with previously used models and solutions, Predictive results interpreter module and Algorithm module.



**Figure 5**

Example of flow-shop technology route with stores, quality control and repairing machines. The production line includes two machines on which quality control is carried out and three additional machines for repairing defects. The first quality control machine detects one type of defects. This type of defects is repair on one additional repairing machine. The second quality control machine detects two different types of defect. This defects are fixed on two different additional repairing machines. The system provides for the detection of three different types of product defect and three different ways to eliminate them.



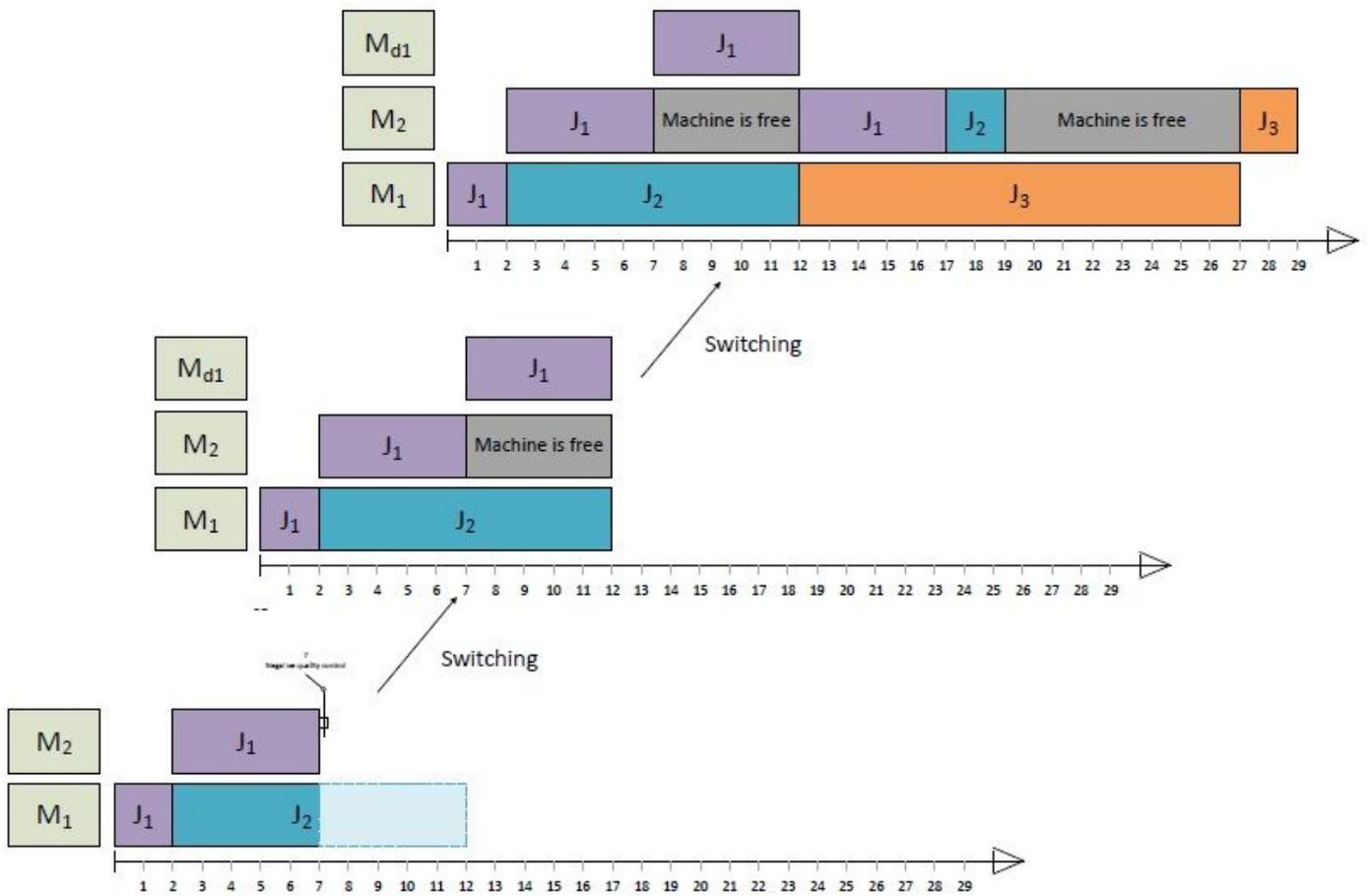
**Figure 6**

Example of flow-shop technology route with stores, quality control, one type of product defect and one repairing machine.



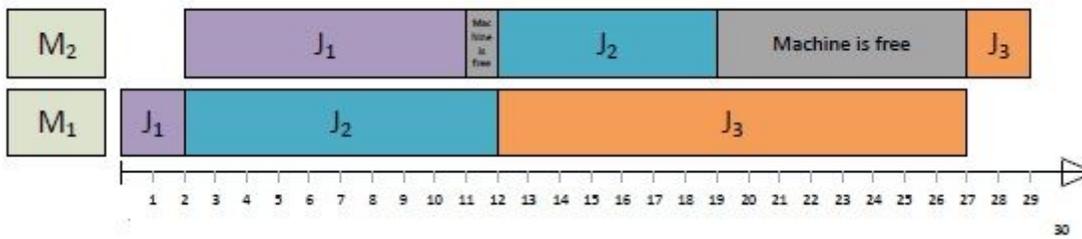
**Figure 7**

Example of optimal schedule in flow-shop system without disturbances for two machine and three jobs.



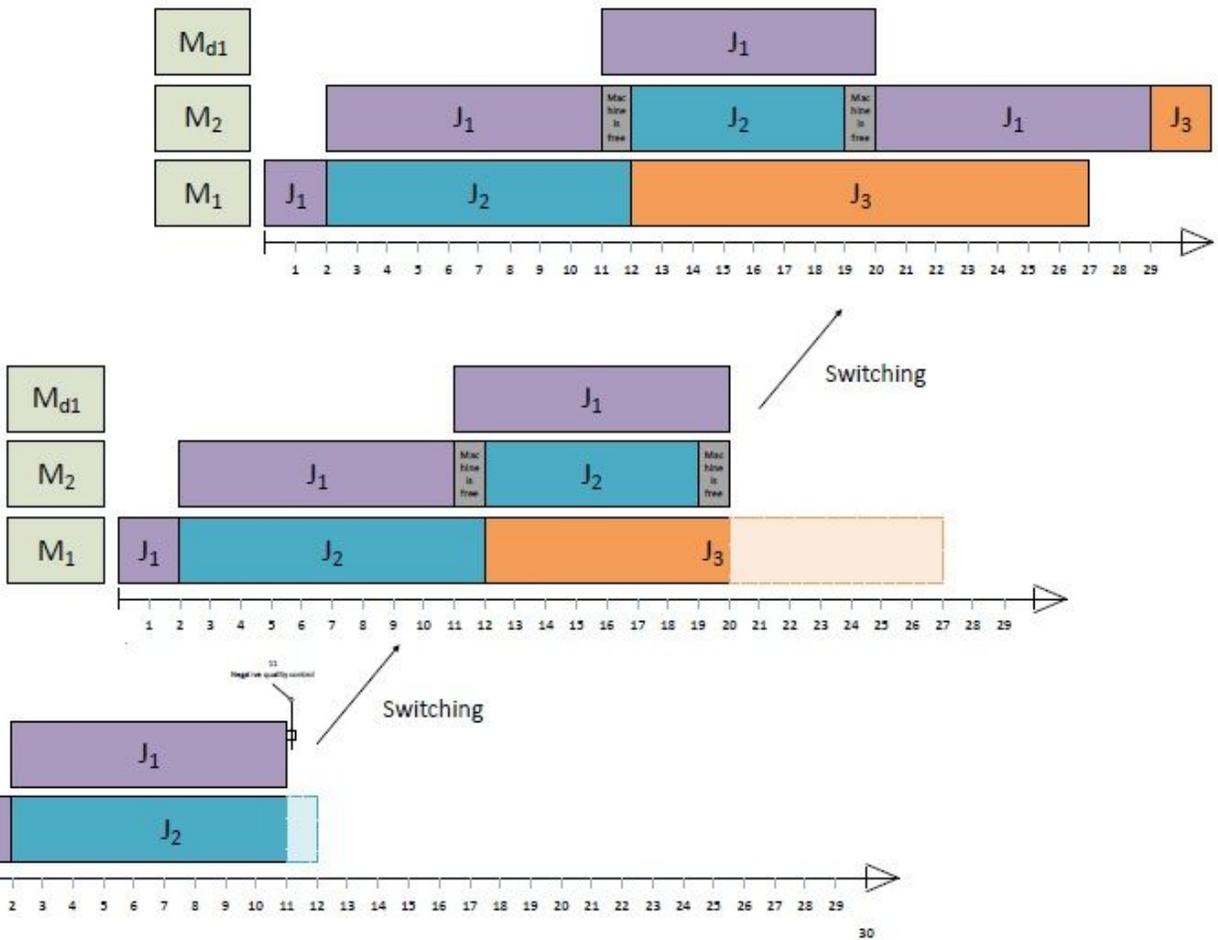
**Figure 8**

Example of the schedule of the flow-shop system with defect detection after processing on the second machine, defect repairing on one additional machine, and then re-processing on the second machine. The schedule presents defect detection for the first job, then the first job is repairing to eliminating defects and then re-processing in the second machine. Switching function enables processing jobs with defects differently than it would result from the technological route.



**Figure 9**

Example of optimal schedule in flow-shop system without disturbances for two machine and three jobs with parameters from the table 2.



**Figure 10**

Example of a schedule of the flow-shop system with defect detection after processing on the second machine, defect repairing on one additional machine, and re-processing on the second machine. The schedule presents defect detection for the first job, then the first job is repairing to eliminating defects and re-process on the second machine. Switching function enables processing jobs with defects differently than it would result from technological route.