



# A skill- and feature-based approach to planning process monitoring in assembly planning

Clemens Gonnermann<sup>1</sup> · S. Ehsan Hashemi-Petroodi<sup>2</sup> · Simon Thevenin<sup>2</sup> · Alexandre Dolgui<sup>2</sup> · Rüdiger Daub<sup>1</sup>

Received: 5 April 2022 / Accepted: 9 August 2022 / Published online: 7 September 2022  
© The Author(s) 2022

## Abstract

Frequent changes in customer needs and large product variety are forcing manufacturing companies to move from mass production to mass customization. Customized production can be achieved by introducing reconfigurable production systems (RMS). The customized flexibility and several characteristics of RMSs provide many opportunities in terms of process and production planning. However, those characteristics greatly increase the complexity of the design and planning of production systems. This paper presents a decision support system relying on a skill-based approach to design a reconfigurable assembly line considering the planning of assembly processes and monitoring. The proposed decision aid system is modular in design and is composed of four modules. The main input data is a CAD model of a new product variant for the identification of the assembly and monitoring requirements. Besides, a current assembly system layout with its resource descriptions exists. In the first developed module, assembly-by-disassembly and a skill-based approach are used to generate different assembly plans. In the second module, feature recognition and skill-based approaches generate process monitoring alternatives. The third module uses a linear program (LP) that aims to minimize the total cost of workstation activation and reconfiguration, as well as cycle time, and to maximize the process quality of the assembly tasks. A user-based generative model design approach is applied to optimize the values of three objective functions. In the fourth and final module, a simulation of the optimized assembly plan allows either the validation of the assembly plan and process monitoring plan or initiates a new iteration due to their infeasibility. To further demonstrate how the proposed methodology works, some computational experiments are provided for two use cases.

**Keywords** Decision support system · Reconfigurable manufacturing system · Assembly · Skill-based approach · CAD analysis · Feature recognition · Optimization

## 1 Introduction

Manufacturing companies are transforming from mass production to mass customization since they are facing increasing product individualization, uncertainty in the market demand, and product and technology variety. This situation causes those manufacturing companies to become more and more flexible and adaptable to quickly respond to any changes in the market and launch new products frequently. In this context, Koren et al. [1] proposed a new manufacturing system concept, called reconfigurable manufacturing system (RMS). This concept is based on the reconfigurability of machines, equipment (hardware), and their control systems (software). In such manufacturing systems, all manufacturing components (hardware/software) are designed taking into account the characteristics [2]: modularity, integrability, scalability, customization, convertibility, and diagnosability.

---

✉ Clemens Gonnermann  
clemens.gonnermann@iwb.tum.de

S. Ehsan Hashemi-Petroodi  
seyyed-ehsan.hashemi-petroodi@imt-atlantique.fr

Simon Thevenin  
simon.thevenin@imt-atlantique.fr

Alexandre Dolgui  
alexandre.dolgui@imt-atlantique.fr

Rüdiger Daub  
ruediger.daub@iwb.tum.de

<sup>1</sup> Institute for Machine Tools and Industrial Management (iwb), Technical University of Munich, Boltzmannstr. 15, Bavaria, Garching 85748, Germany

<sup>2</sup> IMT Atlantique, LS2N, Cedex 3, Rue Alfred Kastler 4 - La Chantrerie, Nantes 44307, France

An RMS is defined as a manufacturing system where physical manufacturing components and process capabilities can be added, removed, or modified easily and efficiently to change the production capacity. Thereby, an RMS can create the functionality and capacity that are needed, when they are needed [3]. RMSs offer higher flexibility in production and can therefore meet the challenges resulting from today's trends in production (e.g., individualized products, shorter product life cycles, higher quality demands) [4]. To be able to handle such flexibility a great amount of manual effort and expert knowledge is needed to plan the production processes. Especially in assembly, which causes most of the resulting costs of a product, the amount of time and expert knowledge required to consider the huge variety of possible assembly sequences and assembly plans is immense [5]. Therefore, automated planning approaches must be applied to handle this kind of flexibility efficiently. Besides planning assembly processes including the assignment of production devices and stations, the quality of the produced goods must be considered. This can be done by simultaneously planning process monitoring [6]. This paper addresses today's challenges in production planning by generating and analyzing various assembly and monitoring plans offered by an RMS. We identify the different alternatives for assembling a new product in an existing production system and simultaneously consider the possibility of introducing process monitoring. The developed automated planning methodology includes the optimization of the solution space (assembly plans) concerning production criteria (e.g., necessary reconfigurations, monitoring efficiency). The rest of this paper is organized as follows: Section 2 presents an overview of the state of the art on which the methodology of this paper is based. Furthermore, existing automated planning processes, necessary semantic taxonomies, and existing optimization approaches are presented. In Sect. 3, we discuss the challenges that are addressed in this paper. Additionally, the vision which motivates our research in this field is introduced. Section 4 first describes an overview of the methodology before the individual modules of the decision aid system are introduced and explained. Validation is given in Sect. 5 by presenting two case studies that show the applicability and functionality of the proposed methodology. Section 6 gives a conclusion based on the results and some perspectives for future research.

## 2 Literature review

Industrial companies are faced with a high level of uncertainty, changes, fluctuations, and a lot of challenges and constraints corresponding to the market, product variety, and manufacturing technologies. In this context, manufacturing systems need to increase their flexibility, adaptability, and reconfigurability. For this reason, within the evolution of manufacturing systems, they have switched from dedicated manufacturing systems

(DMS) to flexible manufacturing systems (FMS), and then to reconfigurable manufacturing systems (RMS). As mentioned, the concept of RMS is introduced by [1], which allows the company to manufacture a part family of products and react quickly and efficiently to any changes in the market. An RMS can be reconfigured by adding, removing, or changing the physical structure of the system or the manufacturing components. Thereby, an RMS can create the capacity and functionality that is needed when it is needed [3, 7–9]. Overall, an RMS combines the advantages of DMS and FMS in terms of higher throughput and customized flexibility [2]. In the following, the literature on RMS in general and automated planning in combination with skill-based approaches and optimization approaches for process planning to efficiently use the flexibility of RMS are reviewed. Last, approaches for feature-based identification of process requirements are presented. The combination of these approaches for the automated identification of resource capabilities and process requirements in an RMS and their matching represent an important cornerstone of the methodology presented here.

### 2.1 RMS literature review

The literature on RMS has considered several problems of operations management at different levels such as planning, design, and operation [10–13]. Since our study corresponds to the design of an RMS with process monitoring, this literature review focuses on this concept. With regard to the recent literature study of [11], little attention has been paid to diagnosability in the literature on the RMS. Diagnosability, which is an ability to quickly detect issues concerning the quality and reliability in a system, is one of the six characteristics of an RMS [3].

The development of sensor technology enables the monitoring of processes by measuring geometric, kinematic, and mechanical values, as well as fluid parameters and temperature [14]. Process monitoring and especially inline process monitoring have therefore become more relevant to address global trends such as production in small batch sizes while maintaining high production quality [15, 16]. Process monitoring is investigated as a feasible quality control method based on sensing and control technology [17]. Online signals, such as force and temperature, are collected to determine the states of manufacturing processes [18]. Planning process monitoring on the other hand is quite time and cost consuming and requires high expert knowledge so that in today's production it is often only used for mass production. To enhance the diagnosability of RMS, automated planning of process monitoring is an essential aspect to consider.

In 2018, [2] went into details of the characteristics, principles, and architecture of the RMS. A key aspect is the performance

improvement of RMS and the reduction of design effort by introducing cyber-physical manufacturing systems and new intelligent manufacturing technologies. This concept assists manufacturers and is fully applicable in practice. For this reason, the concept of a digital twin has been researched for the last several years. The term digital twin describes a digital image of a physical or immaterial object or process from the real world. Thus, the digital twin forms a simultaneous representation that exists throughout the entire life cycle of the depicted system [19–21]. The aim is to enable production to become an autonomous decision-making and control system by using simulations and sensor data in planning and operation [22]. To be able to benefit from RMS, the heterogeneous field device landscape must be handled efficiently. Therefore, a semantic description of devices and functionalities, as well as an automated analysis of the solution space, is necessary.

## 2.2 Automated process planning

Automated process planning represents a decisive research focus to use production systems efficiently in a flexible and targeted manner [23]. Skill-based approaches have become increasingly popular for automated process planning due to their flexibility. These approaches are also described as capability-based approaches [24, 25]. [26] present a capability-based approach for the evaluation of changes in the RMS. This approach has also been extended by integrating rules so that skills can be combined to determine the impact of changes in a production system by identifying different capability models and thus functionalities in a production system [27]. Due to different understanding and definition of capabilities, the focus of skill-based approaches varies. [28] focuses on planning and scheduling by using a skill-based approach to automatically generate production schedules. Furthermore, [29] combine taxonomies of different publications to build up a skill taxonomy that relies on norms and VDI guidelines (e.g., DIN 8580, VDI 2860). They focus on skills in assembly so that an existing production system is able to offer these skills for the assembly and thereby satisfy product requirements. [30] build upon this approach and defined skills that are relevant for monitoring processes in assembly. The results of these studies [29, 30] make it possible to generate a skill model of an existing assembly system by focusing on actorial and sensorial functionalities. The combination of skills has also been presented in these papers. Through this, functionalities of resources can be connected with each other in an existing assembly system to identify more skills than can be seen at first glance. This makes it possible to automatize the identification of skills in the existing assembly system by concerning actorial (e.g., assembly processes) and sensorial (e.g., monitoring processes) skills. The generation of assembly plans via skill-based approaches allows to use the flexibility of RMS. Hereby, multiple plans can be created automatically. Identifying which process plan to use hereby

leads to an assembly line balancing problem in automated process planning [31]. Using optimization approaches and algorithms can be a favorable condition to identify the most suitable assembly plan based on certain criteria (e.g., number of reconfigurations, reconfiguration cost and time) [32]. This will be discussed in the following subsection.

## 2.3 Optimization approaches for assembly line balancing

Assembly line balancing (task assignment) and design (resource allocation) are crucial steps for the production planning [33]. These topics have been studied by many researchers [34–37]. The lack of a complete planning approach that covers configuration and reconfiguration of a line taking into account the assembly and monitoring processes can be seen in the literature [33]. In the literature on assembly lines, several objective functions are optimized using different optimization methods. Such optimization problems for assembly line balancing and production planning are usually modeled using mathematical programming [38–41], and solved by different solution approaches, either exact or approximate methods like (meta-)heuristic algorithms [42–44]. For instance, [38] build a linear programming model and constructive heuristics to solve a workforce minimization problem in a paced assembly line, motivated by an automotive industry case. The goal was to find a workforce assignment, which minimizes the maximal number of workers used in all production cycles. [45] describe a bi-level optimization approach to combine the selection and positioning of production resources into one single optimization problem. Hereby, a nested genetic algorithm has been developed which is capable of solving an assembly line balancing problem with resource selection and layouting. Moreover, the optimization approaches can be integrated to simulation techniques which is called simulation-based optimization approaches. These approaches have been adopted to manufacturing systems' optimization, efficiently [46]. For example, [47] developed a simulation-based optimization approach using a discrete event simulation model and simulated annealing (SA) algorithm, to solve a production planning and resource allocation problem in reconfigurable manufacturing system.

## 2.4 Feature-based approaches for the identification of requirements

Products and processes place requirements on devices and skills. These requirements can be detected by identifying features in CAD models (computer-aided design) that store information about products, parts, or resources throughout

the entire product life cycle [48]. Functions and requirements can thereby be assigned to specific products and parts. These can either be geometric information (e.g., topology) or describe functional information (e.g., product manufacturing information (PMI) [49]). Depending on the application domain, features are defined differently for computer-aided design, manufacturing (CAM), and process planning (CAPP) models [48, 50]. Especially in manufacturing planning, feature-based approaches are used to define product and process requirements [51]. [52] describe an approach to identify critical manufacturing features on metallic aerospace components. Hereby, a rule-based feature recognition method is used to identify features on different representations of solid models (i.e., curve solid geometry (CSG), boundary-representation (B-rep)). In assembly, features have been considered just recently as relevant for process planning [53]. [54] summarizes several studies about assembly features and creates a structure of assembly features. The recognition of features varies among graph-based, hint-based, cell-based, rule-based, neuronal-network-based, and convex hull decomposition recognition approaches [52, 55]. These approaches are differently efficient depending on the required outcome and product complexity. For example, rule-based approaches have been shown to produce the best results for the identification of assembly features due to their scalability and transparency [52, 54].

## 2.5 Summary of the literature review

Up to now methodologies and decision support systems for process planning have not addressed the aspect of diagnosability in RMSs. As can be seen in Sect. 2.1, monitoring becomes more relevant in production and specifically in RMS. Unfortunately, there are few approaches to how to consider process monitoring in today's (semi-)automatic production or assembly planning. Existing methods in assembly planning have not yet been adapted to process monitoring planning (Sect. 2.2). CAD feature recognition is a promising way to identify relevant data in CAD models (assembly and/or parts) as can be seen in Sect. 2.4. These methods and tools can be transferred and modified for monitoring planning (e.g., identification of relevant parameters/data on individual products and/or parts). Therefore, this paper provides an approach on how to plan monitoring processes alongside assembly planning. This approach enables a reduction in manual effort, which is needed to plan monitoring processes especially for complex products and production systems.

## 3 Problem description

The problem addressed in this study concerns the design of an RMS, with humans and robots as resources in the assembly line. Several product variants are assembled on an existing line.

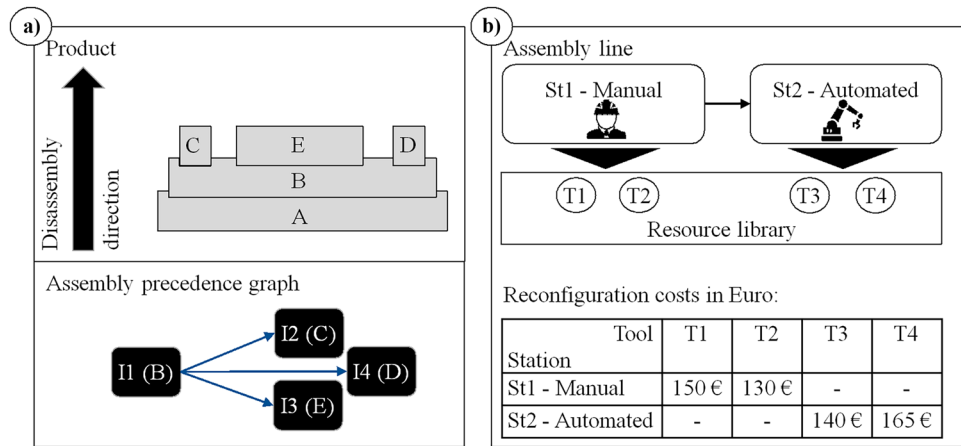
The CAD model of the new product variant and the simulation model of the assembly line, including their resources and tools, serve as inputs. These inputs are derived from the product design (new CAD model and process description due to a new product variant) and assembly line (digitized assembly system). In our problem description, only one robot or worker works at each station. Robots and workers are assigned to workstations. A set of automated and manual equipment pieces (i.e., tools) is considered in the resource library of the line (part of the digitized assembly system) which can be used to execute a task (i.e., process). Note that each piece of equipment may also be able to monitor an assembly process, which allows using it as a sensor for process monitoring at the station (e.g., torque sensor).

The CAD model ideally contains all information about the assembly sequence and, if possible, process information as well. Nevertheless, additional process data can be provided as input via supplementary documents. The assembly operations for each product must be performed according to the priority relationships between the tasks of the respective product. This information is contained in the CAD model and additional documents and must be obtained. Furthermore, additional information must be generated for each task to describe which combinations of resources and tools can be used for a process. Here, the processing time of each task depending on the combination of tool and resource, the reconfiguration costs of the tools in the line, and the efficiency of using each tool for process monitoring are needed to make a valid decision.

On the other hand, the skill model of the assembly line, which contains the current configuration of the resources and tools in the stations and additional tools from the resource library that can be used for any reconfigurations, is provided as input. The skill model contains semantic descriptions of the individual resources, tools, stations, and thereby assembly line. Hereby, algorithms are able to automatically analyze the assembly line configuration and match individual skills with product and process requirements (e.g., necessary torque). A structure is required to store these matches, which can be enriched using matchmaking algorithms to provide more detailed criteria (i.e., key performance indicators (KPI)). These are necessary to make critical decisions when selecting an assembly and process monitoring plan. Graph-based models have been proven to be a suitable structure for saving multiple matches or tasks (e.g., assembly precedence graph, and/or graph [56, 57]). Valuable information for such decisions in a production graph is the feasibility of the task, necessary reconfigurations (including the additional time and costs), process parameters (e.g., assembly paths, cycle times, efficiency of process monitoring), and task types (e.g., screwing — primary; handling — secondary).

As a result, there are several alternatives that require a multi-criteria selection approach or a user-centered optimization model. Based on any restriction from the user

**Fig. 1** The information related to a simple example, assembling a single new item and the assembly line. The left side of the figure **a**) displays the product domain with the individual parts (A, B, C, D, E) and tasks required (I). The right side of the figure **b**) shows the assembly line domain with the stations (St) and tools (T)



(the decision-maker) concerning the value of one or more criteria (e.g., cycle time, monitoring efficiency, number of reconfigurations, costs), the rest of the criteria are optimized. The feasibility of the optimal solution concerning collision freedom and reachability must be ensured through a simulation model of the individual assembly processes. This loop (optimization-simulation) continues until it finds a feasible and optimal process plan for the assembly line. The problem is further clarified with a small example given in Fig. 1.

The description of the assembly line and features of only one type of product are given in Fig. 1. We assume an assembly line arranged and equipped as it is shown in Fig. 1b. The line possesses two stations, one is manual and one is automated, which are equipped by a worker and a robot, respectively. Some manual and automated tools exist in the resource library. The current state of the line must be reconfigured for producing a new product with possible process plans. The new product requires a set of four tasks (I1, I2, I3, I4) as given by the precedence graph in Fig. 1a and Table 5 in Appendix 2 (e.g., task processing times, precedence graph, possible matches of resources, and tools to perform the tasks).

According to the data provided by the product and production system (e.g., CAD models, process parameters, resource capabilities), a set of feasible processes can be generated. Table 5 in Appendix 2 describes an approach how

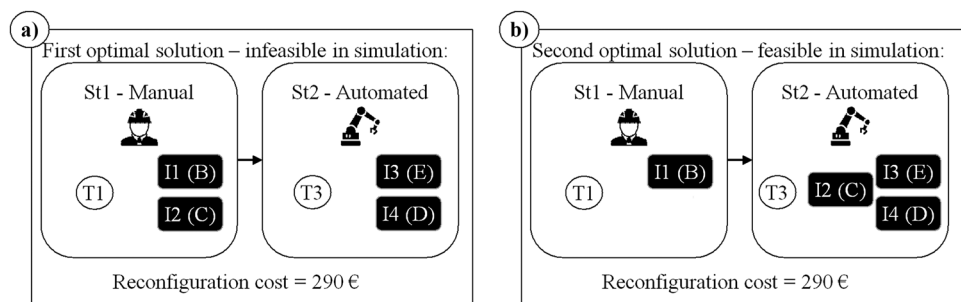
to display the matches and which criteria are relevant. Each valid match between a tool (T) and a task (I) is described with a process time (Pt in seconds) and a monitoring efficiency (Mon. Eff. in percent) based on suitability of the resource and tool. The objective of the optimization model is to determine the optimal solution with regard to individual criteria (e.g., cycle time and monitoring efficiency) for an assembly plan including monitoring tasks in order to enable high process quality. Fig. 2a shows, as an example, the optimal process plan with reconfiguration costs of 290 € (reconfiguration costs for T1 and T2).

This solution (shown in Fig. 2a) is not feasible according to the simulation model, since the second task (I2) cannot be performed by the first manual tool (T1) in the first station 1 due to a collision. Thereby, by implementing the simulation-optimization loop, the final feasible and optimal process plan is achieved with the same objective function value of 290 € for the reconfiguration cost (see Fig. 2b).

### 4 System overview

The system proposed in this paper analyzes digital products and production system data to generate multiple feasible assembly plans including the capability to monitor processes. Fig. 3 describes the 4 modules that result in an optimized

**Fig. 2** **a**) The optimal solution from the optimization model, but an infeasible solution in the simulation model. **b**) The optimal solution from the optimization model, and a feasible solution in the simulation model





assembly and inspection plan by analyzing a new product variant (i.e., CAD model) and existing digitized production system (i.e., layout, resource descriptions). Multiple alternative assembly and process monitoring plans are generated that increase the flexibility of an RMS. Continuous user interaction enables the demand-driven selection of an optimized plan depending on the defined criteria (i.e., number of necessary reconfigurations, the relevance of process monitoring in each individual process step). Each of the four modules of the system can be viewed individually and is connected to the next one via a text-based data format (i.e., XML and JSON). The following subsections will describe each module as well as its inputs and outputs.

Module (1) focuses on the generation of assembly plans through an assembly-by-disassembly approach as described in [58]. Assembly plans contain information about the assembly sequences, process types (e.g., joining), assembly path and process specific parameters (e.g., torque).

Module (2) displays a new approach for the planning and generation of alternatives to monitor assembly processes inline. This means that no additional and unnecessary monitoring processes are generated that increase the cycle time, but rather parallel monitoring processes are

identified. A rule-based CAD feature recognition approach in combination with a general process requirement database identifies the monitoring requirements. Existing assembly plans are then used in module (2) to generate a variety of process monitoring alternatives that result in new assembly plans that include process monitoring.

Module (3) focuses on the optimization of these alternatives depending on user criteria (e.g., cycle time, production costs, number of reconfigurations). This results in one optimized assembly plan with a process monitoring plan.

The last module (4) validates the optimized assembly plan via a simulation in which the optimized assembly plan, including the monitoring processes, is checked for collision-free and visual reachability. If an assembly plan including its process monitoring alternative is not feasible, the optimization module identifies another optimal assembly plan taking into account process monitoring for the validation module. To keep the simulation effort low, the validation by means of simulation of the process plan has been placed after the optimization module. For complex assembly modules and a large number of possible process plans, the simulation effort can quickly become very high. Selecting the assembly plan with optimization before simulation reduces the number of simulation runs.

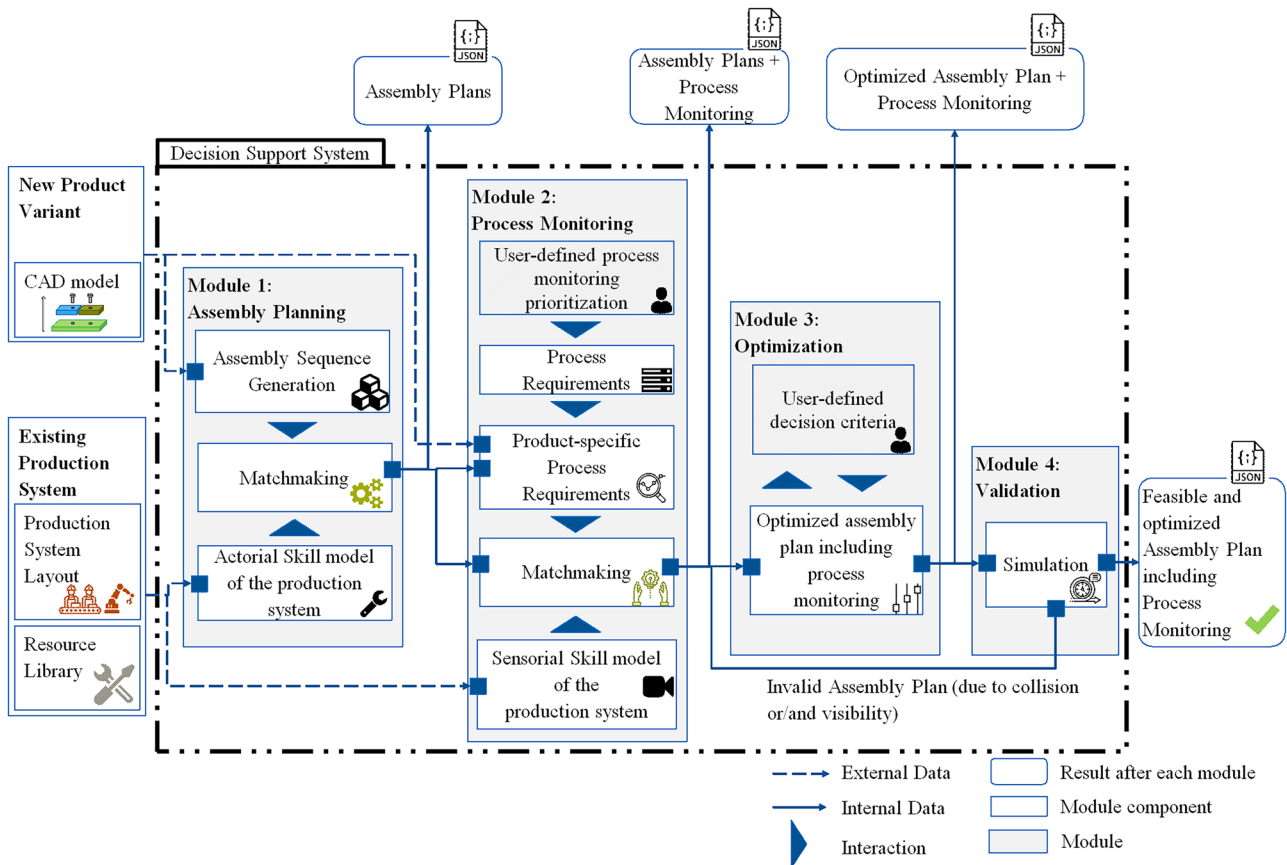


Fig. 3 The overall methodology with all decision aid modules

### 4.1 Generation of assembly plans

To identify valid and collision-free assembly sequences, an assembly-by-disassembly approach is used to analyze CAD models of new product variants [57–59]. Each part of the CAD assembly group is virtually disassembled to generate assembly sequences and define assembly paths for each part of the product as described in Fig. 4. An assembly process graph, independent of the production system and its resources, is generated. Here, each vertex (node) in the process graph contains information about the necessary assembly process and its requirements (not yet of the production system and resources, i.e., devices). The requirements describe the necessary tasks semantically to allow broad applicability for a variety of products (e.g., screwing, joining).

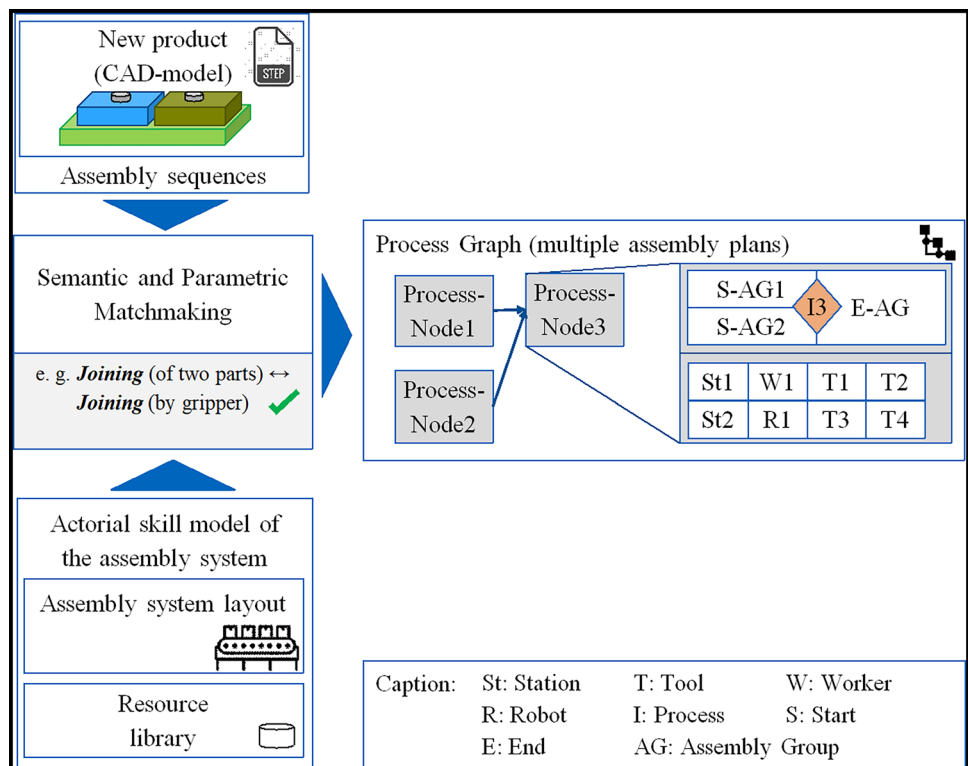
The allocation of each requirement to each assembly process and part is executed automatically. Additional or missing information (e.g., torque, force) can be inserted afterward by the user as this module serves as an assistant for the identification of valid assembly sequences and processes. Standard parts and constraints between parts are identified automatically to define assembly processes such as *screwing* and *joining*. To identify the specific requirements of each assembly process, a semantic skill-based domain is used that can be applied to the product as well as resources and the assembly system. [29] describe a skill domain for actorial skills (e.g., joining, screwing). Regardless of the

sensorial requirements, an assembly process graph can be created that includes the skills of each resource and possible combinations (e.g., robot + gripper).

The matchmaking between product requirements and assembly system skills generates different assembly plans through resource allocation. As a prerequisite, the existing assembly system, its resources (i.e., tools), and additional resources available for production (resource storage) must be digitized with their specific information models, including a semantical description of functionalities (e.g., joining). The digitized production system is described in Fig. 4 as the actorial skill model of the production system. The resource library describes relevant technical and economic device functionalities (e.g., velocity, investment cost) as well as skills.

The generation of the process graph, containing the assembly processes and possibly allocated resources to stations, is divided into two steps. The first step deals with the identification of tools and stations that are suitable for executing the production system-independent assembly plans. Each necessary product and process requirement (e.g., joining, screwing) is semantically matched with the corresponding skill (e.g., gripping, moving, screwing). Then, the alignment parameters are verified for their suitability in the second step (e.g., gripping-width, joining-force, torque). If existing resources at a station do not fit due to missing skills or unfitting parameters, a reconfiguration is taken into account. Resources in the library (i.e., storage) are

**Fig. 4** Module (1) — generation of assembly plans through an assembly-by-disassembly and skill-based approach



considered for their suitability (station restrictions need to match, e.g., interfaces — manual station-worker, automatic station-robot, pneumatic, electrical interfaces). The process graph at the end contains several assembly processes, which in combination represent different assembly plans.

### 4.2 Generation of process monitoring alternatives for existing assembly plans

To date, there is no framework or system that addresses CAD models and assembly information and relates to the automatic determination of parameters for process monitoring. Therefore, the following section presents methods to determine process monitoring alternatives for existing assembly plans. As illustrated in Fig. 5, the product-specific identification of process monitoring requirements can be divided into three steps: templates for monitoring requirements, feature recognition, and parameter determination. These steps consist of importing data (e.g., CAD-model of the assembly, assembly plans), selection of monitoring requirement templates (i.e., solution neutral monitoring requirements for assembly processes), data extraction (i.e., rule-based feature recognition), and determination of monitoring requirements (i.e., filling the templates according to analyses). The output consists of several parameters. Each parameter has five characteristics: variables, values, priorities, units, and error ranges. The

variables and units come from the templates stored in a parameter template subsystem (i.e., a database). The values, priorities, and error ranges of different parameters must be determined or entered by the user (e.g., engineer).

The monitoring requirement templates contain product-neutral information about monitoring parameters relevant for individual assembly processes and are stored in a parameter template subsystem. These templates need to be filled through product-specific information. In the feature recognition step, features are automatically recognized using a hybrid geometric reasoning and rule-based approach (Fig. 5a). The CAD model including its parts is imported automatically. Afterward, the geometric and topological information can be extracted from the CAD models using an Open Cascade library (pythonocc). A unique ID is assigned to each feature. Hence, different features can be associated with each other through their IDs. Different topological entities are found, for example, solids and faces. Skill parameters [30], standard components (fasteners, such as DIN 912 for screws and DIN 934 for nuts), and standards related to connection (such as DIN 13-1 for ISO threads) are stored in the parameter template subsystem, describing requirements of processes (e.g., templates for assembly processes). According to the classification of joining processes from DIN 8593-0, each parameter template stores the specifications for a specific assembly process. Concrete assembly processes (such as screwing or point welding) are specified as tables in the databases.

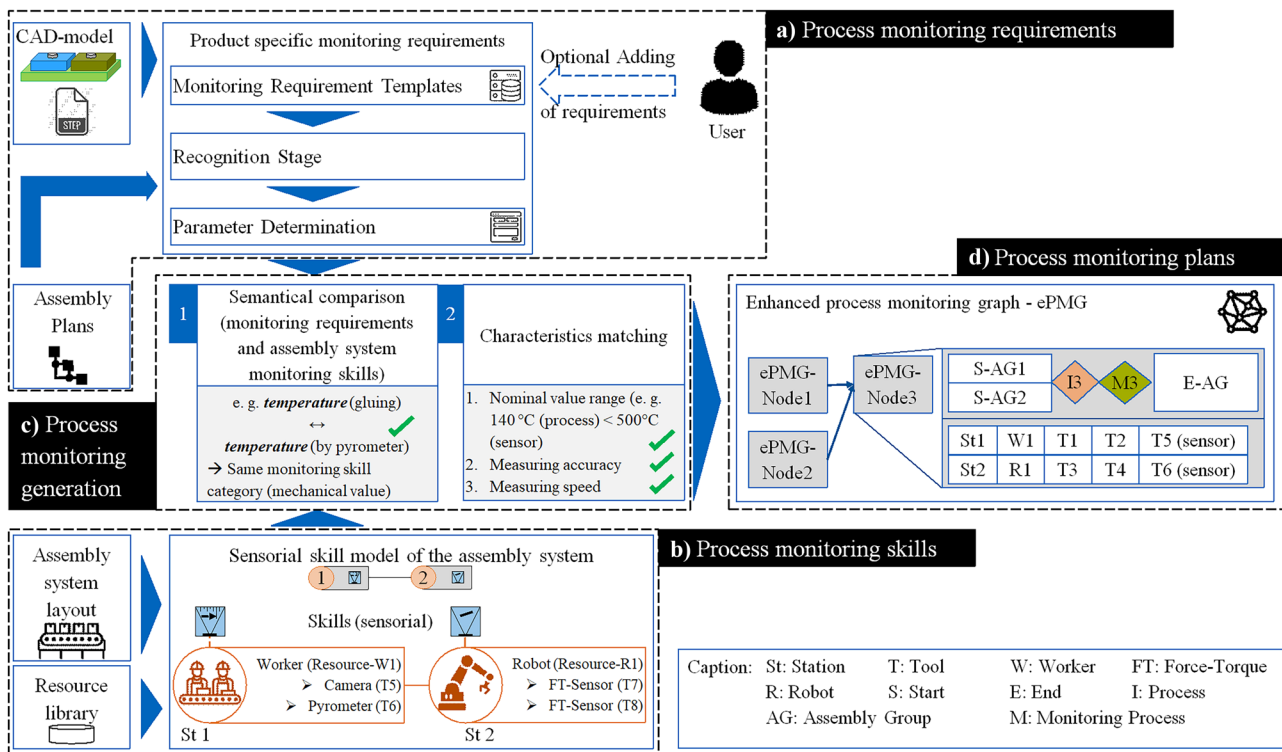


Fig. 5 Module (2) — decision support module for the automated generation of process monitoring alternatives for existing assembly plans. The steps a) to d) display the individual domains of the submodules (e.g., a) monitoring requirements, b) monitoring skills)



While creating or managing a database, engineers need to extract and construct data from standards and specifications (i.e., DIN, VDI). These tables can be supplemented or changed manually by an expert if necessary when processes or relevant process requirements have changed (e.g., due to new technologies or quality requirements of processes).

Based on the features and the filled process monitoring templates, a set of parameters is selected and collected in the parameter determination step (Fig. 5a). During this step, the user can either confirm or deny parameters and values, priorities, units, and error ranges for monitoring. The sensorial skill model of the production system is set up automatically according to the assembly system layout (Fig. 5b). Therefore, a skill-based approach is used as described in [30] and [56].

The automated generation of process monitoring alternatives is executed by a comparison of product-specific monitoring features and assembly system-specific (i.e., resource-specific) monitoring skills [6, 56]. The semantical analysis focuses on the presence or absence of monitoring skills, i.e., skills related to measuring and checking physical process properties. After the semantical analysis, parameters are checked to ensure that the monitoring feature can be monitored efficiently by the aligned resource (i.e., characteristics matching in Fig. 5c). Different parameters, depending on the specific skill and monitoring feature, are analyzed and matches are determined (e.g., joining force, temperature). Once a matching pair of a monitoring feature and a monitoring skill of a resource is identified, the comparison of the feature and resource-specific parameters is initialized. If the criteria are not fulfilled for any monitoring resource in the production system (i.e., no reconfiguration possible), the feature is designated as invalid for monitoring and the matchmaking continuous for the next feature. Matching efficiency (i.e., monitoring efficiency) depends on the alignment of resources with monitoring requirements (e.g., sensorial skills) according to [56]. The weighting of the pairing depends on the relevance being defined by the user for each monitoring requirement in the monitoring requirement step (Fig. 5a).

The nominal value range of the monitoring feature of the assembly process defines the parameter comparison. To ensure suitability, the resource must be able to measure values across the range. As an example, a pyrometer that cannot detect temperatures below 300 °C is not suitable for monitoring a hot gluing process performed at temperatures between 120 and 180 °C. Likewise, a force-torque sensor must be able to monitor the exact force range in which the process takes place. The minimum and maximum values of the requirement ( $R_{min}$  and  $R_{max}$ ) and the range in which the skill can be used by the resource ( $S_{min}$  and  $S_{max}$ ) define the match and its suitability. If the range  $R_{min}$ – $R_{max}$  is not contained entirely within the interval  $S_{min}$ – $S_{max}$ , the resource is considered unsuitable, even though it has the appropriate skill.

The assessment of the measuring accuracy of the resource is a further criterion for the parameter comparison. This includes quality-critical or high-precision processes where a fraction of a deviation can affect the outcome of a process. The logic for the evaluation is based on a comparison of the required accuracy for the process and the accuracy of the resource for the skill (e.g., 0.1 N is represented with the accuracy value 10). If there is no resource at a station with sufficient accuracy to monitor a feature, it is assumed that reconfiguration is necessary for that feature.

Measuring speed expressed as the frequency with the unit Hertz is a parameter to be checked continuously. This ensures that the monitoring process does not add any additional time to the cycle time. The parameter comparison is used to identify unsuitable sensors that need to be replaced or adjusted. If a resource fulfills the qualitative and quantitative matchmaking to monitor a feature, the information is stored in a node (vertex) of the enhanced process monitoring graph (ePMG) (see Fig. 5d — enhanced process monitoring graph). If monitoring requirements cannot be matched with sensors (i.e., resources) a reconfiguration of the production system is necessary. Determining which resources can be used to cover the missing skills is done through further iterations via matchmaking. Matchmaking between monitoring characteristics and the resource library is integrated so that whenever a process is classified as *not monitorable*, a query for the relevant characteristics is started and a potential resource is searched in the resource library. The process monitoring efficiency (Mon. Eff. in percentage) depends on the resource inheriting the sensorial skill. This value is calculated from the match (best fit) of the sensorial skills with the individual monitoring requirement (parameter match and number of individual matches) and the accuracy of the sensor. Any monitoring efficiency between 0 and 100 % represents a match, whereas 100 % indicates the best possible match.

The result of the module (2) is a process graph and process monitoring graph containing valid assembly plans and process monitoring alternatives (i.e., allocated actorial and sensorial resources and stations). These alternatives are further analyzed in module (3) (optimization). Process times are approximated depending on manual and automatic assembly processes.

### 4.3 Optimization model

The next module aims to find the optimal process plan for the models taking into account all theoretically feasible process plans generated in the production graph. A generative design model for a user-optimization system is built. The goal is to propose an optimization model which incorporates users' (decision-makers) opinions relating to

the three criteria: minimization of the reconfiguration cost and the cycle time, and maximization of the minimum monitoring efficiency of all tasks. To be precise, a user-optimization framework is designed where the user can enter the value that he/she wants for each objective function, and get the optimal values for the rest. The rest of this section covers importing data from the production graph to the optimization model, the mathematical model, and the user-optimization loop.

The production graph contains the inputs of the optimization model, namely, the set of precedence  $A$  including pairs of tasks  $(i, i')$  where task  $i$  must be performed before task  $i'$ , the compatibility among tools, tasks, and resources through a binary matrix  $K_{it}^r$  that shows the capabilities of tools to perform a set of tasks on the resources, the processing time  $Pt_{irt}$  of task  $i$  if executed on resource  $r$  with tool  $t$ , and cycle time  $C$ . Note that the compatibility matrix  $K_{it}^r$  provides two sets  $N_{tr}$  and  $N_{ir}$ , which respectively contain the tasks that can be performed by tool  $t$  on resource  $r$  and the tools that can execute task  $i$  on resource  $r$ . Moreover, several tools are used for process monitoring, so-called sensors, which can monitor the quality of tasks executions.  $\alpha_{irt}$  refers to the efficiency of using tool  $t$  (as a sensor) to monitor task  $i$  on resource  $r$ , and takes the value in range  $[0, 1]$ .  $c_{tr}$  denotes the set-up cost (per time unit) of tool  $t$  on resource  $r$ . Besides,  $a_r$  refers to the set-up cost of resource  $r$  for use.

Based on this data, the following mixed integer linear programming (MILP) is solved to find the assembly plan with minimum reconfiguration cost, minimum cycle time, and maximum level of monitoring efficiency. The model has three binary decision variables, namely,  $x_{irt}$  is equal to 1 if task  $i$  is performed by tool  $t$  on resource  $r$  (0 otherwise),  $z_r$  is equal to 1 if resource  $r$  is opened (0 otherwise), and  $y_{rt}$  equals to 1 if tool  $t$  is assigned to resource  $r$  in the configuration associated to process task  $i$  (0 otherwise). Besides, two continuous variables  $q_i$  and  $Q$  are defined which respectively represent the monitoring efficiency of executing task  $i$  and the minimum monitoring efficiency of all tasks. For these decision variables, a quantification of the monitoring efficiency is required. This efficiency can be calculated from the ratio of the match between process monitoring requirements and the capabilities of the matched resource 4.2. The nomenclature with further details on the quantities, parameters, and variables considered is given in Appendix 1. The proposed MILP is given in Eq. (1)–(16), where  $\mathcal{I}$ ,  $\mathcal{T}$ , and  $\mathcal{R}$  denote the sets of tasks, tools, and resources, respectively.

$$\min \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} c_{tr} y_{rt} + \sum_{r \in \mathcal{R}} a_r z_r \quad (1)$$

$$\min C \quad (2)$$

$$\max Q \quad (3)$$

s.t

$$Q \leq q_i \quad i \in \mathcal{I} \quad (4)$$

$$q_i = \sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} \alpha_{irt} x_{irt} \quad i \in \mathcal{I} \quad (5)$$

$$\sum_{r \in \mathcal{R}} \sum_{t \in \mathcal{T}} x_{irt} = 1 \quad i \in \mathcal{I} \quad (6)$$

$$\sum_{i \in \mathcal{I}} x_{irt} \leq y_{rt} \quad t \in \mathcal{T}, r \in \mathcal{R} \quad (7)$$

$$\sum_{r \in \mathcal{R}} y_{rt} \leq 1 \quad t \in \mathcal{T} \quad (8)$$

$$\sum_{t \in \mathcal{T}} y_{rt} \leq z_r \quad r \in \mathcal{R} \quad (9)$$

$$\sum_{t \in \mathcal{T}} \sum_{i \in N_{tr}} Pt_{irt} x_{irt} \leq C \quad r \in \mathcal{R} \quad (10)$$

$$\sum_{t \in \mathcal{T}} \sum_{r \in \mathcal{R}} r x_{irt} \leq \sum_{t' \in \mathcal{T}} \sum_{r' \in \mathcal{R}} r' x_{i'r't'} \quad (i, i') \in A \quad (11)$$

$$x_{irt} \in \{0, 1\} \quad i \in \mathcal{I}, t \in \mathcal{T}, r \in \mathcal{R} \quad (12)$$

$$y_{rt} \in \{0, 1\} \quad t \in \mathcal{T}, r \in \mathcal{R} \quad (13)$$

$$z_r \in \{0, 1\} \quad r \in \mathcal{R} \quad (14)$$

$$q_i \in [0, 1] \quad i \in \mathcal{I} \quad (15)$$

$$Q \in [0, 1] \quad (16)$$

The objective function Eq. (1) is to minimize the total cost, including the used resources activation cost and reconfiguration cost of the tools that have been used in the line, where sensors are counted as tools. The objective function (2) aims to minimize cycle time. The objective function Eq. (3) maximizes the minimum monitoring efficiency of all task execution monitored by sensors. Constraint Eq. (4) ensures that objective function Eq. (3) maximizes the minimum monitoring efficiency of all tasks, which is computed by constraints Eqs. (4) and (5). Constraint Eq. (6) states that each task must be processed by exactly one resource. Constraint (7) ensures that a tool

is available on the resource to process each assigned task. Constraint (8) prevents assigning a tool to more than one resource. Constraint Eq. (9) opens a resource if at least one tool is located on that resource. Constraint Eq. (10) states that the total processing time on each resource cannot exceed the cycle time. Constraint Eq. (11) ensures the precedence relationships between the tasks. Constraints Eqs. (12), (13), (14), (15) and (16) give the domains of the variables.

In real case studies as well as in future researches, different variants of this model can be considered. For example, constraints Eqs. (7) and (9) could be written for each task and tool. As another example, the uncertain processing time for tasks can be considered in constraint Eq. (10). In addition, several constraints can be considered, such as area constraints that prevent or penalize the assignment of a set of tasks to the same station.

For the user-optimization loop, the user can give his/her opinion for the value of any objective functions Eqs. (2) and (3). Next, this value is taken into account by adding any constraint Eqs. (17), (18) and (19) to the model and removing the corresponding objective function from the model. In this context, the upper limit for the reconfiguration cost of each model and cycle time, and the lower limit for the monitoring efficiency of each model, proposed by the user, are shown as  $Cost_{user}$ ,  $Cycletime_{user}$ , and  $Quality_{user}$ , respectively. For example, if the user aims at cycle time  $Cycletime_{user}(second)$  and  $Quality_{user}(\%)$  quality for the process in the whole line, two constraints Eqs. (18) and (19) are added to the mathematical model, and it is tackled with only one objective function Eq. (2) to minimize the reconfiguration cost.

$$\sum_{r \in R} \sum_{t \in T} c_{tr} y_{rt} + \sum_{r \in R} \alpha_r z_r \leq Cost_{user} \tag{17}$$

$$C \leq Cycletime_{user} \tag{18}$$

$$Q \geq Quality_{user} \tag{19}$$

To guide the user in the selection of relevant values for the target, the tool provides upper and lower bounds for each objective. Let  $o$ ,  $o_1$ , and  $o_2$  be three objective functions like the ones existing in the proposed MILP. Solving the model for objective  $o$  with no restriction on the other objectives yields a lower bound for  $o$ . To compute an upper bound for objective  $o$ , we solve the multi-objective model with the lexicographic approach. We consider any orders where  $o$  is the least important objective, and the maximum value for  $o$  is an upper bound. If the user sets infeasible targets, the tool may suggest a correction for the two objectives  $o_1$  and  $o_2$  where a target is set. To suggest a new value for  $o_1$  (resp.  $o_2$ ), the model is optimized for  $o_1$  (resp.  $o_2$ ) with the target on  $o_2$  (resp.  $o_1$ ).

#### 4.4 Feasibility testing through the simulation model

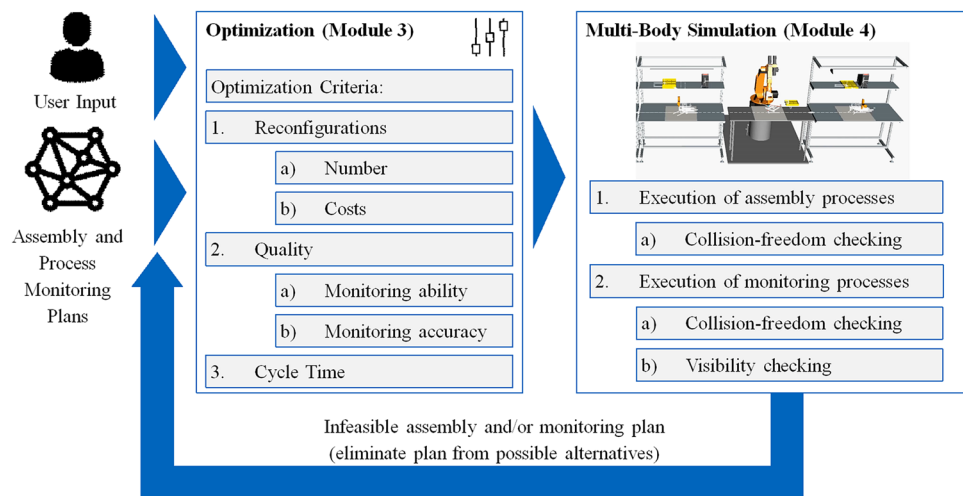
Module (3) generates an optimized assembly plan including resources that are available for monitoring. The last module of the methodology serves as a validation (Fig. 6). A global 3D simulation of the entire assembly line is executed to analyze the collision freedom and feasibility of the assembly plan. Hereby, each device involved in the individual assembly, feeding, and monitoring processes are checked for collision freedom and accessibility in a multi-body simulation. The multi-body simulation consists of rigid components such as assembly parts, stations, and resources, which are loaded as step files with the additional information (assembly and monitoring plan) via a standalone solution through an application programming interface (API). Kinematics (e.g., robots) are set in advance whereas the movements of assembly parts and resources (e.g., gripper and screwdriver) can either be calculated using the information of module (1) simulation and assembly-by-disassembly or set manually.

Process Simulate (PS) has been chosen as a simulation software in this paper. During assembly and feeding execution, the device and assembly path are checked for any collisions. In the meantime, devices responsible for monitoring are checked for collision freedom and visual accessibility. Information about the individual resources and processes is provided by the assembly and monitoring plan identified by the optimizer. The simulation model to which the assembly and monitoring plan data is applied is an existing digitized production system. During the simulation of the assembly and monitoring processes, continuous collision and visibility checking evaluates if the entire plan is feasible or infeasible and needs to be tagged as “infeasible.” This information is played back to the optimization module as a set of constraints. A new iteration is executed to identify the optimal assembly and monitoring plan.

### 5 Computational experiments and results

This section serves to validate the methodology proposed in this study. First, two case studies, at two different levels of complexity, are presented with all associated data. The first case study shows a relatively simple product to be assembled, illustrating the flexibility available in creating an optimized assembly plan including a process monitoring alternative. The second case study considers a somewhat more complex product to be assembled with different types of assembly processes (i.e., “joining,” “screwing”). For case study (2), the same assembly system is used as in case study (1), which is an RMS. The computational results are presented for each case study in the following. The

**Fig. 6** Modules (3) and (4) — optimization and simulation module for the identification of an optimal and feasible assembly and monitoring plan



methodology is implemented on different computer systems and the individual modules are implemented in different programming and software environments (see Table 1).

The individual module numbers and descriptions correspond to the nomenclature from the methodology in this article. Table 1 also shows the programming and software environments used to implement the developed methods and the computer system on which they were carried out for both case studies. Software programs often interact with self-developed solutions, e.g., for CAD feature recognition (FreeCAD, PythonOCC, and C# implementation) or multi-body simulation (Process Simulate and C# implementation).

### 5.1 Computational results of case study 1 — LEGO® product

The first case study displays a simple product that needs to be assembled on an existing assembly line (Fig. 7). The assembly and monitoring plan are generated automatically by using the methodology described in Sect. 4. The product, consisting of 4 parts (1 LEGO® plate (base part) and 3 LEGO® bricks), already displays the complexity which assembly and inspection planners are facing. The inherent flexibility resulting from the number of possible assembly sequences, stations, and resource allocation cannot be efficiently accounted for in manual processing. Various criteria have to be taken into account during the generation of assembly plans (e.g., fulfillment of process requirements, number of reconfigurations). Additionally, monitoring aspects have to be taken into account (e.g., monitoring alternatives and their monitoring accuracy).

Therefore, the methodology presented in this paper serves as an assistant by automatically providing decision support during the planning phase. In the first module of the methodology, the LEGO® product is analyzed for its valid and collision-free assembly sequences (Fig. 7). Three

sequences generated by the assembly-by-disassembly approach are displayed in Fig. 7. Figure 8 shows a hybrid assembly line of 5 stations. The first and the last three stations (1, 3, 4, 5) are manual stations equipped with supply tools, screwdrivers, screwdriver controllers, and a worker. The second station is an automatic station where a robot automatically executes processes with end effectors, in this case, grippers. Each resource (e.g., a robot, a worker) accompanied by a tool (e.g., screwdriver or gripper) has skills such as “screwing” or “joining.” Therefore, each combination of tools is designated for specific processes. The combination of the assembly line layout with each resource and the resource library with the skill taxonomy allows the automated generation of a skill model. Each station inherits its specific skills depending on the resources.

The skill model of the production system can be displayed as a graph or matrix. In Fig. 8, the default configuration as well as resource storage of the assembly system are shown. Once the product requirements and assembly line skills are defined, the assignment of production resources to processes automatically creates assembly plans. This procedure has been described in Sect. 4.1 and partly in [58]. The comparison generates the assembly plans using semantical matchmaking and parameter matchmaking (quantitatively). Due to the fact that this methodology considers the planning of process monitoring alongside the assembly planning, the optimization and simulation-based validation are done after the allocation of resources/tools for monitoring. When assembly plans have been generated considering the necessary assembly processes, the module (2) identifies product requirements that are relevant for monitoring the assembly processes. The input is provided by the assembly plans, the CAD model of the product, the process monitoring database with tables and logical rules for the identification of geometries. The CAD model is analyzed automatically by using predefined geometrical rules in Python in combination

**Table 1** Software and hardware used in the case studies — implementation of the methodology

Nr.	Module description	Analysis	Software/programming environment	Hardware
1	Assembly planning	Assembly-by-disassembly	Siemens NX (1904), NXOpen, C# Implementation (.NET Framework 4.7.2)	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz and 16,0 GB of RAM under MS Windows 10 Edu (64 bit)
		Process generation	C# Implementation (.NET Framework 4.7.2)	
		Resource allocation — regarding assembly layout	C# Implementation (.NET Framework 4.7.2)	
2	Process monitoring planning	CAD feature recognition	FreeCAD, PythonOCC, C# Implementation (.NET Framework 4.7.2)	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz and 16,0 GB of RAM under MS Windows 10 Edu (64 bit)
		Process generation for monitoring	C# Implementation (.NET Framework 4.7.2), MySQL	
		Skill modeling of assembly system	C# Implementation (.NET Framework 4.7.2), Protegé, Semantic Web Rule Language (SWRL)	
		Generation of monitoring processes regarding existing assembly processes and assembly system layout	C# Implementation (.NET Framework 4.7.2), Libraries for Levenshtein Distancing	
3	Optimization	Optimization	IBM ILOG CPLEX Optimization Studio V12.10	Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz 2.11 GHz and 32 GB of RAM under MS Windows 10 Pro (64 bit)
4	Validation	Multi-body simulation (assembly processes and monitoring processes)	Process Simulate (Tecnomatix 15.1), C# Implementation (.NET Framework 4.7.2)	Intel(R) Core(TM) i7-7700HQ CPU @ 2.80GHz 2.81 GHz and 16,0 GB of RAM under MS Windows 10 Edu (64 bit)



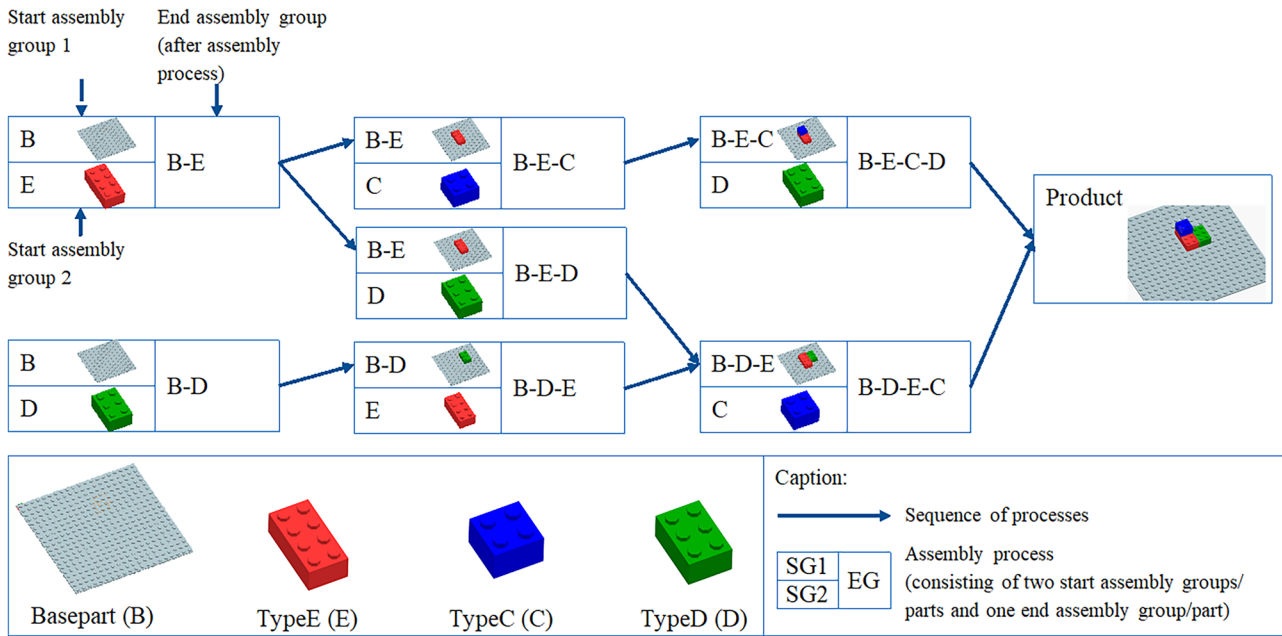


Fig. 7 Assembly process graph of the first case study (LEGO® product) automatically generated using assembly-by-disassembly

Layout	Station 1 → Station 2 → Station 3 → Station 4 → Station 5				
Resources	R1: Worker	R2: Robot	R3: Worker	R4: Worker	R5: Worker
Tools	T1: Screwdriver T2: Control T3: Supply device	T4: Gripper T5: FT-Sensor T6: Supply device	T7: Screwdriver T8: Control T9: Supply device	T10: Screwdriver T11: Control T12: Supply device	T13: Screwdriver T14: Control T15: Supply device
Skills (sensorial/actorial)	Screwing Handling Joining	Handling Joining	Screwing Handling Joining	Screwing Handling Joining	Screwing Handling Joining
Resource - storage	Manual resources (e. g. for worker stations) T17: Control T18: Screwdriver		Automatic resources (e. g. for robot stations) T20: FT-Sensor T19: Gripper		T16: Supply device

Fig. 8 Existing assembly line with five stations to which resources (R) and tools (T) are assigned

with an Open Cascade library in Python (pythonocc). The recognition stage allows the identification of geometrical shapes and constraints. Therefore, contacts, chamfers, and their positions that are relevant for the success of the assembly process can be identified and fill the requirement table (template) for process monitoring. The table is derived through the fuzzy search algorithm (e.g., joining) from the database of the monitoring requirement stage (see Sect. 4.2). In this use case, the table already contains the relevant parameter templates (e.g., joining force in [Nm]) necessary to assemble the LEGO® bricks, and therefore the user does not have to create the template manually.

After the feature recognition and parameter determination define the monitoring requirements, the comparison of monitoring skills and monitoring requirements starts. Semantical matching allows the identification of a variety of different resources for monitoring. The comparison of sensorial skills and monitoring requirements, taking into account the ability to assemble the product at each station (from assembly plans), identifies different alternatives for process monitoring. These alternatives have not yet been validated due to their parameters. The second step allows a parameter comparison where efficiency, speed, and accuracy are taken into account. Multiple assembly plans and alternatives for process monitoring are automatically generated by combining individual assembly/monitoring matches or processes. The individual assembly and monitoring processes are displayed in Appendix 3 in Table 6. Possible combinations of resources are shown for each station (e.g., second station (automatic robot station 2) R2 (robot) + T4 (gripper) + T6 (supply device)). Hereby, resources (R) define the station type (e.g., manual or automatic station) and inherit the essential skill “move.” Tools (T) must always be connected to a resource. Depending on the process, process times (Pt in seconds) vary due to the assembly path of each LEGO® brick. As described in Sect. 4.2, the process monitoring efficiency (Mon. Eff. in percentage) depends on the resource inheriting the sensorial skill. This quantified value of monitoring efficiency can take values between 0 % and 100 % and represents a match, while 100 % indicates the best possible match. Therefore, at station 2 the combination (Station 2: R2; T4, T5, T6) with a force-torque sensor (T5) has a higher monitoring efficiency than the combination without a force-torque sensor (Station 2: R2; T4, T6). Both combinations require a reconfiguration (Rec.) for the second process due to the required gripping width (here T19).

Each assembly plan contains different alternatives for process monitoring as a result of module 2 and the allocation of resources with sensorial skills. Due to the similar component geometry and process parameters of the individual LEGO® bricks, the requirements for process monitoring do not differ between the tasks (see tasks 1, 2,

and 3 in Table 6 in Appendix 3). This leads to similar match results between monitoring requirements and resource skills. Therefore, the individual process monitoring alternatives do not differ between tasks at the same station with the same resources and combination of tools (e.g., monitoring efficiency of 20% for tasks 1, 2, and 3 on station 1; Table 6). In a further step, assembly plans are optimized with various alternatives for process monitoring. Criteria for the optimization were explained in Sect. 4.3.

The obtained production information is used to solve the mathematical model for the first case study. As a result of the optimization, regarding the defined ranges for each pair of two criteria (see Sect. 4.3) by the user, we obtain several assembly plans with an alternative for process monitoring. Since the user defines several ranges for only two criteria, the mathematical model aims to optimize the third one at each run. The user must select only one of the obtained assembly plans (the best solution). Note that the assembly plan must be validated by a 3D multi-body simulation (Sect. 4.4). If collisions occur due to an infeasible assembly sequence or resource option and collisions or lack of visibility are detected, the assembly plan is considered infeasible. This leads to an iteration where the optimization model must take into account a set of new constraints on assignments of tasks and tools to resources (stations). The optimization model will determine another optimal assembly plan with process monitoring and provide another assembly plan. This plan also must be verified by the simulation model. The process ends if the solution of the optimization model is valid through the simulation model. Table 7 in Appendix 3 shows all optimal solutions concerning the optimization of each selected criterion from the user side. Precisely, at each level, the user defines a range for two criteria and optimizes the third one. The optimal objective function value is obtained and it is shown with the corresponding tool and tasks assignments to the stations. These solutions are given to the user, then he/she selects the proper one among others. The user is satisfied by forcing the total costs to less than 350 €, the cycle time to less than 20 (s), and obtaining 94 % as the minimum monitoring efficiency level of the tasks. Table 2 shows that the feasibility of the final selected assembly plan cannot pass through the simulation model since T19 cannot be used to perform task 2 on the robot existing at the second station. Due to the gripper geometry of T19, the LEGO® brick “E” cannot be assembled after LEGO® brick “D” (see Fig. 7 and Table 2). A collision occurs in the multi-body simulation between the gripper and the already assembled LEGO® brick “D.” The constraints concerning the infeasibility of these assignments are added to the mathematical model which prevents the model from taking into account such assignments. The optimization model is solved considering new constraints generated by the simulation model and provides the new optimal solution. As

**Table 2** Final optimization results after validation via simulation (case study 1)

Generative intervals			Objective value	Task/tool assignment				
Cost (€)	Cycle time (s)	Mon. Eff.	(type)	R1 (worker)	R2 (robot)	R3 (worker)	R4 (worker)	R5 (worker)
350	20	-	0.94 (quality)	-	Tasks 1, 2, 3 T4, T6, T19	-	-	-
Validation via simulation (constraint: tool “T19” cannot be used to perform task 2 on robot “R2”)								
390	17.5	-	0.25 (quality)	-	Tasks 1, 3 T4, T6	-	-	Task 2 T15

it can be seen in Table 2, the new assembly plan is valid in the simulation model. Therefore, the final plan corresponds to performing tasks 1 and 3 using T4 and T6 on the robot at the second station (R2), and task 2 using T15 by the worker at the last station (R5). This solution respects the cost of 390 € and the cycle time of 17.5 (s) and provides 25 % monitoring efficiency, which all satisfy the user.

## 5.2 Computational results of case study 2 — toy car product

The second case study deals with a more complex product where two assembly types are considered *joining* and *screwing*. This product (toy car) contains 11 parts (1 base-part, 2 screws, and 8 parts for joining). One can easily see that the complexity of assembly plan generation increases due to multiple valid assembly sequences (generated via an assembly-by-disassembly analysis: Fig. 9 in Appendix 4). In combination with the existing production system in Fig. 8 the matchmaking module (Fig. 4) generates multiple assembly plans. Also, the identification of process monitoring features increases in this case study exponentially due to the number and geometries of assembly parts (e.g., geometrical: joining surface, alignment; process: force) and feature types (e.g., torque-screwing, force-joining). The matchmaking module in Sect. 4.2 identifies multiple process monitoring alternatives to each assembly plan as can be seen in Table 8 in Appendix 4. For the screwing processes, different screwdrivers and controllers necessary for the screwdrivers can be identified for the execution of the processes and also their monitoring. This enables the generation of multiple assembly plans including process monitoring alternatives. Consideration of process monitoring allows the optimization module to use various criteria for selecting the best assembly plan, taking into account quality aspects when planning production. Higher monitoring efficiency values indicate that the resources' sensorial skills are more applicable for the monitoring features of the toy car in case study (2) than the LEGO® product of use case (1) (Table 8 in Appendix 4). Similar assembly parts and processes (e.g., axle front (AF) and axle back (AB)) lead to similar monitoring efficiencies. This is caused by the same input criteria for the CAD feature recognition and process type parameters for monitoring. The

geometries of the wheels are the same, as well as the features (i.e., surface, chamfer) and the assembly process *joining*, which must be done with a certain force and coaxial to the axles (the same applies to the rear wheel (WB) and the chassis). Equivalent requirements for process monitoring lead to similar resource allocations (see monitoring efficiencies in Table 8 in Appendix 4). The consideration of different tools from other stations (e.g., T2) or the equipment and tool store (T18, T17) expands the solution space by making greater use of the flexibility of the assembly line. Multiple alternatives for process monitoring can be generated by identifying diverse combinations (e.g., tasks 8 and 9). These alternatives have been saved and transferred to the optimization module (Sect. 4.3) to identify the optimal assembly plan, including the process monitoring alternative. As with the first case study, the production graph is considered in the optimization model for the second case study (the toy car). Table 9 in Appendix 4 shows the best assembly plans obtained by the optimization module. Among all these plans, the user selects the best one which corresponds to a cycle time of 141 (s) and monitoring efficiency of 70 %, and the optimal cost of 640 €, where we perform tasks 1, 3, and 4 at the second station (R2) using tools T4 and T6, and the rest of tasks at the fourth station (R4) using corresponding proper tools T12, T10, and T11. Table 3 shows the results of the simulation model for the performance of the selected assembly plan. Because of a collision, simulation prevents us from performing tasks 9 and 10 using tools T10 and T11. Similar to the first case study, new constraints for preventing such assignments are generated and imported to the optimization model. Therefore, a reconfiguration is needed to remove the tool T10 and use tools T11 and T12 (which are already located at the station). However, this plan results in a bit higher cost (690 € compare to 640 €). This plan with a certain configuration of the route can be realized with the simulation model.

## 5.3 Discussion

The results of the two case studies show how assembly plans, including alternatives for process monitoring, can be generated automatically. Even with a minimal increase in complexity, one can see how the number of possible assembly plans increases. In addition, planning

**Table 3** Final optimization results after validation via simulation (case study 2)

Generative intervals			Objective value	Task/tool assignment				
Cost (€)	Cycle time (s)	Mon. Eff.	(type)	R1 (worker)	R2 (robot)	R3 (worker)	R4 (worker)	R5 (worker)
-	< 141	> 0.7	640 (cost)	-	Tasks 1, 3, 4 textitT4, T6	-	Tasks 2, 5, 6, 7, 8/tasks T12/T10, T11	-
Validation via simulation (constraint: tool “T10” cannot be used to perform tasks 9–10 by worker “R4”)								
-	< 141	> 0.7	690 (cost)	-	Task 1, 3, 4 T4, T6	-	Tasks 2, 5, 6, 7, 8/tasks 9–10 T12/T12, T11	-

process monitoring for an assembly plan further enhances complexity. Multiple alternatives for process monitoring can be identified for every single assembly plan. This methodology enables efficient and demand-driven planning of process monitoring alongside assembly planning. Due to multiple criteria (number of reconfigurations for assembly processes, number of reconfigurations for monitoring processes, monitoring accuracy and speed as monitoring efficiency, assembly and monitoring process costs, etc.) users can prioritize individual assembly and process monitoring plans. As shown in the results this can be achieved by connecting the information to an optimization module. Two observations can be derived from the case studies: First, the importance of the individual parameters defining the assembly system (e.g., resource and reconfiguration costs), and second, the consideration of investing in new resources and/or tools. The first observation concerns both case studies and the high activation costs of each station. This leads to more appropriate assembly on fewer stations to reduce assembly costs. When the activation costs decrease, other criteria such as monitoring efficiency come into play more in the optimization module. This allows the use of more stations in the line. A second remark is that both case studies consider occupied or not reconfigurable resources and/or tools, but investing in new resources/tools is not taken into account. The simulation makes it possible to give feedback if a resource/tool is available for reconfiguration (i.e., not occupied on other station). As an example of case study (2), if T2 is used in station 1, it cannot be used for reconfiguration in station 3 for the same assembly plan.

Furthermore, scalability has not yet been tested. Increasing the complexity of the product to be assembled as well as the assembly system with its resources and tools has shown a direct relation to the data being generated (assembly plans and monitoring alternatives). It can be argued that individual modules can work more efficiently on a higher scale than others. For example, the generation of assembly plans due to assembly-by-disassembly is quite time-consuming for complex products due to the simulation effort. This can be seen in Table 4. The calculation times are approximate values determined from screen recordings. The time required for user interactions was not taken into account. The individual times assigned to the modules are therefore pure computing times (in seconds).

### 6 Conclusion

The proposed methodology represents an approach to integrating process monitoring planning into assembly planning in combination with optimization. Considering monitoring processes as a relevant aspect in process planning can result in higher quality in process execution and final products. An early consideration of monitoring processes enables a more targeted generation of suitable assembly plans based on individual decision criteria (e.g., number of reconfigurations vs. higher accuracy in process monitoring and thus process control). To handle the complexity of various product variants, assembly sequences, assembly and monitoring resources, and assembly and monitoring plans, the proposed methodology has been implemented with proper feasibility and efficiency. We showed that the developed support system is very useful in practice to link the product designer and the process planner. Automated analysis of product CAD files to identify assembly and monitoring processes, and resource matching through a skill-based approach in an existing production system, enables more efficient process planning and use of RMS. In particular the aspect of diagnosability through planning variable process monitoring alternatives has been addressed in this paper. The optimization module shows that additional

**Table 4** Performance of individual modules

Nr.	Module description	Case study (1)	Case study (2)
		Computing time	Computing time
1	Assembly planning	Approx. 370 s	Approx. 840 s
2	Process monitoring planning	Approx. 0.74 s	Approx. 0.83 s
3	Optimization	Approx. 0.01 s	Approx. 0.08 s
4	Validation	Approx. 1160 s	Approx. 2350 s

criteria, by integrating process monitoring alternatives via a quality factor, can extend and improve the generation of assembly plans for RMS. This is the first conceptual attempt to show the performance of the proposed support system which has the potential to be extended considering following future research directions. In further research, the scalability of this methodology has to be tested by taking into account more complex products and production systems. Considering a large number of tasks leads to a combinatorial explosion in the number of possible assembly plans. Therefore, developing an appropriate optimization approach (e.g., a (meta-)heuristic) can be a useful future research contribution on the application of the proposed support system in large cases. Several extensions can be considered in the optimization model, as mentioned in this paper, such as adding area constraints for task assignment. Moreover, considering mobile robots and moving workers, as well as studying human-robot collaborative systems, opens interesting research avenues.

## Appendix 1 The nomenclature and definitions of sets, parameters, and variables

All the sets, indices, parameters, and decision variables are defined in detail as follows:

Sets and indices	
$\mathcal{R}$	The set of resources (worker/robot) $r, r' \in \mathcal{R}$ .
$\mathcal{T}$	The set of combinations of tools $t, t' \in \mathcal{T}$ .
$\mathcal{I}$	The set of tasks $i, i' \in \mathcal{I}$ .
Parameters	
$c_{ir}$	The set-up cost of tool $t$ on resource $r$ .
$a_r$	The activation cost of resource $r$ .
$\alpha_{irt}$	The efficiency (in percentage %) of using monitoring tool (e.g. a sensor) $t$ on resource $r$ to monitor the quality of task $i$ .
$pt_{irt}$	The process time of task $i$ performing by tool $t$ on resource $r$ .
$C$	The cycle time.

Sets and indices	
$A$	The set of pairs of tasks $i$ and $i'$ where task $i$ precedes task $i'$ .
$K_{it}^r$	The compatibility matrix which shows the possible combinations of resource $r$ and tool $t$ that are capable to perform task $i$ .
$N_{tr}$	The set of tasks that can be performed by tool $t$ on resource $r$ .
$N'_{ir}$	The set of tools that can be used to perform task $i$ on resource $r$ .
Decision variables	
$x_{irt}$	Binary variables that is equal 1 if task $i$ is assigned to resource $r$ to be performed using tool $t$ on this resource.
$y_{rt}$	Binary variables that is equal 1 if tool $t$ is assigned/installed on resource $r$ .
$z_r$	Binary variables that is equal 1 if resource $r$ needs to be activated/used.
$q_i$	Continuous variables that show the monitoring efficiency of task $i$ .
$Q$	Continuous variables that show the minimum monitoring efficiency among all tasks.

## Appendix 2 Problem description

Table 5 shows the possible task resource assignments. Tasks I1, I2, I3, and I4 are assigned to tools T1, T2, T3, and T4, which are located in station 1 and station 2 and their respective resources (i.e., worker and robot). For each allocation, a process time (in seconds) and monitoring efficiency (percentage of match of alignment) value has been generated depending on the alignment of the task requirements and tool/resource skills. These values have been generated with the matchmaking of module (1) and module (2) of the decision aid system (Fig. 3). The layout information provided by the production system makes it possible to generate various process plans from these matches.



**Table 5** Compatibility among tasks (I), tools (T), and resources (robot/worker), the processing time of tasks (Pt), and efficiency values (Mon. Eff.) for each set of task-tool-resource

Resource	Task	T1		T2		T3		T4	
		Pt (s)	Mon. Eff. (%)	Pt (s)	Mon. Eff. (%)	Pt (s)	Mon. Eff. (%)	Pt (s)	Mon. Eff. (%)
Station 1 (worker)	I1	30	95	-	-	-	-	-	-
Station 1 (worker)	I2	15	90	20	80	-	-	-	-
Station 1 (worker)	I3	-	-	-	-	-	-	-	-
Station 1 (worker)	I4	-	-	-	-	-	-	-	-
Station 2 (robot)	I1	-	-	-	-	-	-	-	-
Station 2 (robot)	I2	-	-	-	-	20	85	-	-
Station 2 (robot)	I3	-	-	-	-	25	92	10	95
Station 2 (robot)	I4	-	-	-	-	20	90	15	80

### Appendix 3 Case study 1

Table 6 shows the alternative processes generated after matchmaking module (1) and module (2) for the LEGO® case study. As an example, task 2 at station 2 can be executed with robot R2 when tool T4 (i.e., gripper) is reconfigured with T19 (i.e., gripper with different gripping width). For this alignment, the process time is calculated as 7.2 s and the monitoring efficiency due to the sensory accuracy of the gripper is 40 (%). If the adjustment with the T5 (force-torque sensor) is selected, the monitoring efficiency increases to 94 (%) because the sensor is capable of better monitoring (higher accuracy).

In Table 7, module (3) can be used to generate different optimal process plans depending on the selected criteria. Depending on the selection of user criteria (e.g., (1) cycle time and monitoring efficiency, (2) cost and monitoring efficiency, (3) cost and cycle time), the optimal process plan varies. If the user defines low cycle time and high monitoring efficiency as his priority criteria and is satisfied with the cost of about 350 €, the result is shown in row 3 (all tasks are performed at station 2 with R2).

**Table 6** Multiple assembly processes with monitoring alternatives (Pt, process time; Mon. Eff., monitoring efficiency; and Rec., necessary reconfiguration of the individual assembly process) (case study 1 — LEGO®)

Station	Resource	Tool	Part Task	D 1	E 2	C 3
1	R1—worker	T3	Pt (s)	23.94	17.65	23.94
			Mon. Eff. (%)	20	20	20
			Rec.	-	-	-
2	R2—robot	T4+T6	Pt (s)	6.3	7.2	5.9
			Mon. Eff. (%)	40	40	40
		Rec.	-	T4->T19	-	
		T4+T5+T6	Pt (s)	6.3	7.2	5.9
			Mon. Eff. (%)	94	94	94
Rec.	-	T4->T19	-			
3	R3—worker	T9	Pt (s)	19.4	26.45	19.4
			Mon. Eff. (%)	23	23	23
			Rec.	-	-	-
4	R4—worker	T12	Pt (s)	17.44	23.87	17.44
			Mon. Eff. (%)	34	34	34
			Rec.	-	-	-
5	R5—worker	T15	Pt (s)	33.23	16.29	33.23
			Mon. Eff. (%)	25	25	25
			Rec.	-	-	-

**Table 7** The results of the optimization for each pair of criteria (e.g., cycle time and monitoring efficiency) before validation via simulation (case study 1)

Generative intervals			Objective value (type)	Task/tool assignment				
Cost (€)	Cycle time (s)	Mon. Eff.		R1 (worker)	R2 (robot)	R3 (worker)	R4 (worker)	R5 (worker)
-	< 80	> 0.20	170 (cost)	Tasks 1, 2, 3 T3	-	-	-	-
-	< 40	> 0.20	340 (cost)	Task 2 T3	-	Tasks 1, 3 T9	-	-
-	< 20	> 0.94	350 (cost)	-	Tasks 1, 2, 3 T4, T6, T19	-	-	-
-	< 19	> 0.20	390 (cost)	Task 2 T3	Task 1, 3 T6	-	-	-
-	< 17.5 > 17.43	> 0.34	520 (cost)	-	Tasks 1, 2 T4, T6, T19	-	Task 3 T12	-
< 600	-	> 0.34	17.43 (time)	-	Tasks 2, 3 T4, T6, T19	-	Task 1 T12	-
< 520	-	> 0.20	17.65 (time)	Task 2 T3	Tasks 1, 3 T4, T6	-	-	-
< 390	-	> 0.94	19.4 (time)	-	Tasks 1, 2, 3 T4, T6, T19	-	-	-
< 350	-	> 0.20	34.88 (time)	Task 2 T3	-	-	Tasks 1, 3 T12	-
< 340 > 170	-	> 0.34	58.75 (time)	-	-	-	Tasks 1, 2, 3 T12	-
< 360	< 20	-	0.94 (quality)	-	Tasks 1, 2, 3 T4, T5, T6, T19	-	-	-
< 350	< 20	-	0.94 (quality)	-	Tasks 1, 2, 3 T4, T6, T19	-	-	-
< 390	< 17.5	-	0.25 (quality)	-	Tasks 1, 3 T4, T6	-	-	Task 2 T15

## Appendix 4 Case study 2

All possible assembly sequences according to the constraints imposed by the assembled product (i.e., the CAD file) can be seen in Fig. 9. These sequences were determined using the assembly-by-disassembly approach, where the entire assembled product is virtually disassembled and checked for collision-freeness. Each node in this graph represents two start assemblies and one end assembly that is created after the assembly process (e.g., joining or screwing).

Table 8 shows all process results after modules (1) and (2) are executed. Each process resource allocation is assigned a process time and monitoring efficiency that

shows the match between the monitoring requirements from the process and the resource and tool capabilities for monitoring that specific process.

Table 9 shows the results after the optimization module (module (3)). The user defines criteria (i.e., a multi-criteria problem) so that different process plans are identified as optimal depending on these criteria. A pair of two criteria is defined by the user (e.g., (1) cost and cycle time, (2) cycle time and monitoring efficiency, (3) cost and monitoring efficiency), while on the right side of the table the individual tasks/tool assignments to the predefined criteria are shown.

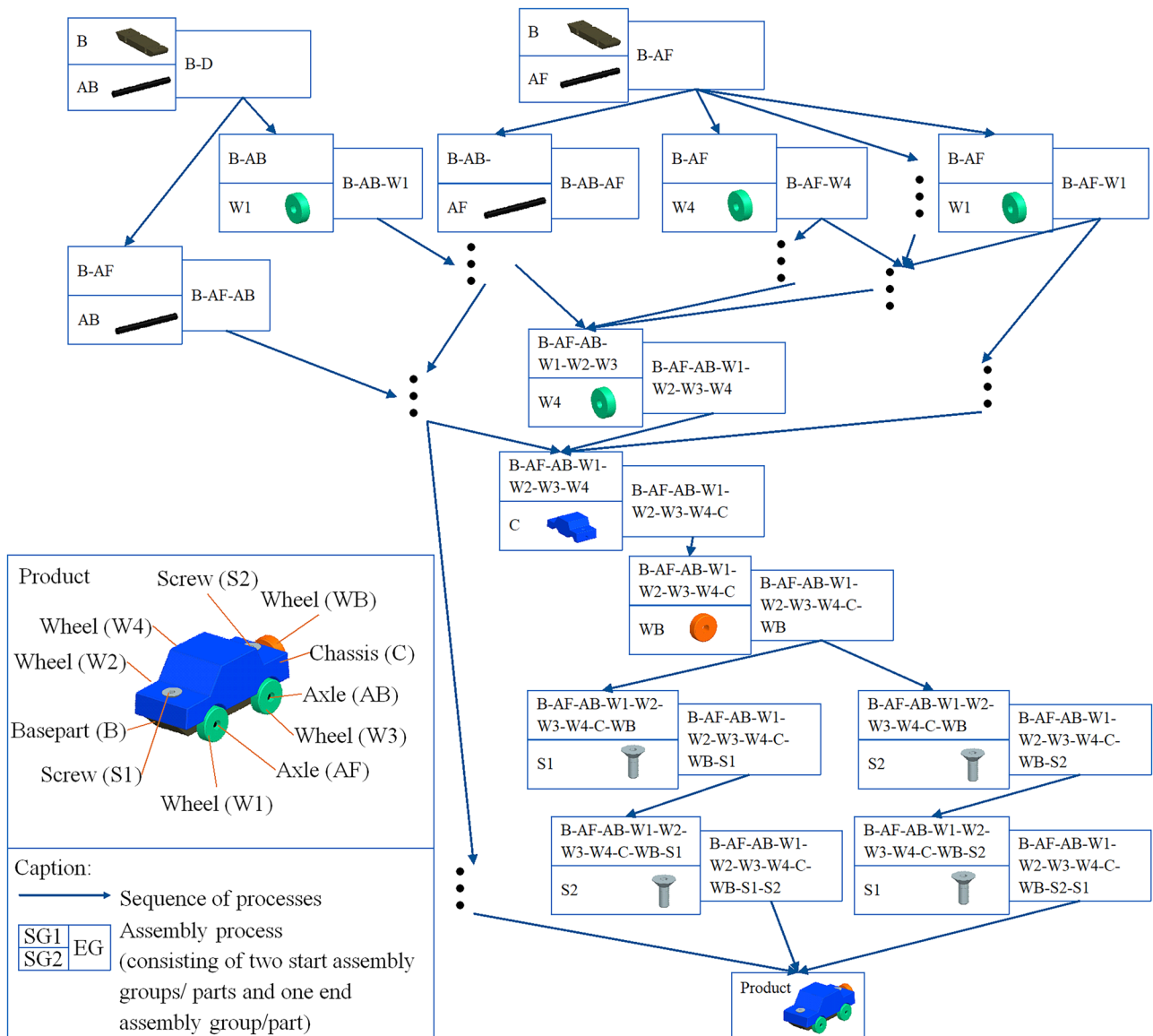


Fig. 9 Assembly precedence graph of the second case study (toy car) generated by the assembly-by-disassembly approach

**Table 8** Multiple matches (assembly and monitoring processes) according to assembly and process monitoring planning, which in sequential combination result in assembly and monitoring plans (case study 2 — toy car)

Station	Resource	Tool	Part	AF	AB	W1	W2	W3	W4	C	WB	S1	S2		
			Task	1	2	3	4	5	6	7	8	9	10		
1	<i>R1—worker</i>	<i>T1+T2</i>	Pt (s)	-	-	-	-	-	-	-	-	23.67	23.67		
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	80	80	
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
			Pt (s)	26.73	26.73	11.4	11.4	11.4	11.4	28.35	13.63	-	-	-	
			Mon. Eff. (%)	76	76	54	54	54	54	53	54	-	-	-	
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
2	<i>R2—robot</i>	<i>T4+T6</i>	Pt (s)	7.56	7.56	3.54	3.54	3.54	3.54	8.94	5.25	-	-		
			Mon. Eff. (%)	64	64	61	61	61	61	59	61	-	-		
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
		<i>T4+T5+T6</i>	Pt (s)	7.56	7.56	3.54	3.54	3.54	3.54	8.94	5.25	-	-		
			Mon. Eff. (%)	96	96	94	94	94	94	89	94	-	-		
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
3	<i>R3—worker</i>	<i>T7+T8</i>	Pt (s)	-	-	-	-	-	-	-	-	25.38	25.38		
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	85	85	
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
		<i>T7+T8</i>	Pt(s)	-	-	-	-	-	-	-	-	-	18.45	18.45	
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	-	95	95
			Rec.	-	-	-	-	-	-	-	-	-	-	T8->T2	T8->T2
	<i>T9</i>	Pt (s)	19.8	19.8	15.76	15.76	15.76	15.76	23.65	18.54	-	-			
		Mon. Eff. (%)	73	73	56	56	56	56	58	56	-	-			
		Rec.	-	-	-	-	-	-	-	-	-	-	-		
4	<i>R4—worker</i>	<i>T10+T11</i>	Pt (s)	-	-	-	-	-	-	-	-	24.51	24.51		
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	95	95	
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
		<i>T10+T12</i>	Pt (s)	-	-	-	-	-	-	-	-	-	21.8	21.8	
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	-	95	95
			Rec.	-	-	-	-	-	-	-	-	-	-	T10->T18	T10->T18
	<i>T12</i>	Pt (s)	18.83	18.83	16.8	16.8	16.8	16.8	24.54	17.92	-	-			
		Mon. Eff. (%)	63	63	74	74	74	74	77	74	-	-			
		Rec.	-	-	-	-	-	-	-	-	-	-	-		
5	<i>R5—worker</i>	<i>T13+T14</i>	Pt (s)	-	-	-	-	-	-	-	-	19.74	19.74		
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	88	88	
			Rec.	-	-	-	-	-	-	-	-	-	-	-	
		<i>T13+T14</i>	Pt (s)	-	-	-	-	-	-	-	-	-	19.74	61	
			Mon. Eff. (%)	-	-	-	-	-	-	-	-	-	-	95	95
			Rec.	-	-	-	-	-	-	-	-	-	-	T14->T17	T14->T17
	<i>T15</i>	Pt (s)	28.34	28.34	22.76	22.76	22.76	22.76	22.4	18.1	-	-			
		Mon. Eff. (%)	81	81	63	63	63	63	76	63	-	-			
		Rec.	-	-	-	-	-	-	-	-	-	-	-		

**Table 9** The results of the optimization for each pair of criteria (e.g., cycle time and monitoring efficiency) before validation via simulation (case study 2)

Generative intervals			Objective value (type)	Task/tool assignment				
Cost (€)	Cycle time (s)	Mon. Eff.		R1 (worker)	R2 (robot)	R3 (worker)	R4 (worker)	R5 (worker)
-	< 200	> 0.50	370 (cost)	-	-	Tasks 1–8/tasks 9–10	-	-
-	< 200	> 0.70	420 (cost)	-	-	T9/T7, T8	Tasks 1–8/tasks 9–10	-
-	< 181	> 0.50	420 (cost)	Tasks 1–8	-	T12/T10, T11	-	-
-	< 181	> 0.70	445 (cost)	-	T3	- Tasks 9–10	-	-
-	< 141	> 0.50	540 (cost)	Task 1 - 6	-	T7, T8	Tasks 1–8	Tasks 9–10
-	< 141	> 0.70	640 (cost)	-	Tasks 1, 3, 4	-	T12	T13, T14
-	< 70	> 0.50	730 (cost)	Tasks 1, 2, 5	-	Task 7–8/tasks 9–10	-	-
-	< 70	> 0.70	790 (cost)	-	Tasks 1, 2, 3, 7	T3	T9/T7, T8	Tasks 9–10
-	40	> 0.50	815 (cost)	Task 2,5	Tasks 1, 3, 4, 7	T4, T6	Tasks 2, 5, 6, 7, 8/tasks 9–10	-
-	40	> 0.60	960 (cost)	T3	T4, T6	Task 7/tasks 9–10	T12/T10, T11	Tasks 9–10
< 370	-	> 0.5	181.7 (time)	960 (cost)	Tasks 1, 3, 4, 7/ task 2	Tasks 9–10	Tasks 3, 4, 6, 8	-
< 420	-	> 0.5	141 (time)	Tasks 1–8	T4, T6/T19, T5, T6	T7, T8	T12	-
< 450	-	> 0.7	147.3 (time)	T3	-	Tasks 1–8/tasks 9–10	Tasks 4, 5, 6, 8	-
< 600	-	> 0.5	76.2 (time)	Tasks 1, 2, 3, 6	-	Tasks 9–10	T12	-
< 800	-	> 0.5	50 (time)	-	Tasks 1, 2, 3, 4, 7	Tasks 5, 6, 8	Tasks 9–10	-
< 800	-	> 0.7	51.5 (time)	-	T4, T6	T9	Tasks 9–10	T13, T14
< 900	-	> 0.5	38.1 (time)	Task 2,6	Tasks 1, 2, 3, 4, 7	Tasks 5, 6, 8	Tasks 9–10	-
< 900	-	> 0.7	51.5 (time)	T3	T4, T5, T6	Tasks 9–10	Tasks 5, 8	-
< 900	< 40	-	0.54 (quality)	Tasks 2, 5	Tasks 1, 2, 3, 4, 7	T7, T8	Tasks 5, 6, 8	Tasks 9–10
				T3	T4, T6	Tasks 9–10	T12	T13, T14
						T7, T8	Tasks 6, 8	-
						T12		



**Table 9** (continued)

Generative intervals			Objective value (type)	Task/tool assignment				
Cost (€)	Cycle time (s)	Mon. Eff.		R1 (worker)	R2 (robot)	R3 (worker)	R4 (worker)	R5 (worker)
< 900	< 181	-	0.74 (quality)	-	Tasks 1, 2, 3, 4, 7 T4, T6	-	Tasks 5, 6, 8 T12	Tasks 9–10 T13, T14
< 450	< 181	-	0.74 (quality)	-	-	-	Tasks 1–8 T12	Tasks 9–10 T13, T14
< 440	< 181	-	0.56 (quality)	-	-	Tasks 1–8 T9	-	Tasks 9–10 T13, T14
< 800	< 70	-	0.74 (quality)	-	Tasks 1, 4, 7/ task 2 T4, T6/T19, T6	Tasks 9–10 T7, T8	Tasks 3, 5, 6, 8 T12	-
< 750	< 70	-	0.56 (quality)	-	-	Tasks 1, 2, 7 T9	Tasks 4, 5, 6, 8 T12	Task 3/tasks 9–10 T15/T13, T14

**Acknowledgements** The authors of this paper would like to thank the Region Pays de la Loire, France ([www.paysdelaloire.fr](http://www.paysdelaloire.fr)), the Franco-German Alliance for Factory of the Future ([www.future-industry.org](http://www.future-industry.org)), and the EU commission. The research in this paper was possible due to the funding of the EU project “ASSISTANT” (grant number: 101000165).

**Author contribution** All authors contributed to the study conception and design. Material preparation, data collection, and analysis were performed by Clemens Gonnermann and S. Ehsan Hashemi-Petroodi. The first draft of the manuscript was written by Clemens Gonnermann and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

**Funding** Open Access funding enabled and organized by Projekt DEAL. This work was supported by Franco-German Alliance for Factory of the Future ([www.future-industry.org](http://www.future-industry.org)), and the EU commission. The research in this paper was possible due to the funding of the EU project “ASSISTANT” (grant number: 101000165).

## Declarations

**Conflict of interest** The authors declare no competing interests.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

- Koren Y, Heisel U, Jovane F, Moriwaki T, Pritschow G, Ulsoy G, Van Brussel H (1999) Reconfigurable manufacturing systems. *CIRP Ann* 48:527–540. [https://doi.org/10.1016/S0007-8506\(07\)63232-6](https://doi.org/10.1016/S0007-8506(07)63232-6)
- Koren Y, Gu X, Guo W (2018) Reconfigurable manufacturing systems: principles, design, and future trends. *Front Mech Eng* 13:121–136. <https://doi.org/10.1007/s11465-018-0483-0>
- Mehrabi MG, Ulsoy AG, Koren Y (2000) Reconfigurable manufacturing systems: key to future manufacturing. *J Intell Manuf* 11:403–419. <https://doi.org/10.1023/A:1008930403506>
- Chryssolouris G (2006) *Manufacturing systems: theory and practice*. Springer, New York, NY. <https://doi.org/10.1007/0-387-28431-1>
- Lotter B, Wiendahl HP (2013) *Montage in der industriellen Produktion: Ein Handbuch für die Praxis*. Springer-Verlag
- Gonnermann C, Reinhart G (2019) Automatized setup of process monitoring in cyber-physical systems. *Procedia CIRP*, 81, 636–640. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12–14, 2019. <https://doi.org/10.1016/j.procir.2019.03.168>
- Elmaraghy H (2005) Flexible and reconfigurable manufacturing systems paradigms. *Int J Flex Manuf Syst* 17:261–276. <https://doi.org/10.1016/j.jmsy.2011.01.001>
- Koren Y, Shpitalni M (2010) Design of reconfigurable manufacturing systems. *J Manuf Syst* 29. <https://doi.org/10.1016/j.jmsy.2011.01.001>
- Mehrabi M, Ulsoy A, Koren Y, Heytler P (2002) Trends and perspectives in flexible and reconfigurable manufacturing systems. *J Intell Manuf* 13. <https://doi.org/10.1023/A:1014536330551>
- Andersen A-L, Brunoe TD, Nielsen K (2015) Reconfigurable manufacturing on multiple levels: literature review and research directions. In: Umeda S, Nakano M, Mizuyama H, Hibino N, Kiritsis D, von Cieminski G (eds) *Advances in production management systems: innovative production management towards sustainable growth*. Springer International Publishing, Cham, pp 266–273. [https://doi.org/10.1007/978-3-319-22756-6\\_33](https://doi.org/10.1007/978-3-319-22756-6_33)

11. Galizia FG, Bortolini M, Mora C (2018) Reconfigurable manufacturing systems: literature review and research trend. *J Manuf Syst* 49:93–106. <https://doi.org/10.1016/j.jmsy.2018.09.005>
12. Hashemi-Petroodi SE, Dolgui A, Kovalev S, Kovalyov MY, Thevenin S (2021) Workforce reconfiguration strategies in manufacturing systems: a state of the art. *Int J Prod Res* 59:6721–6744. <https://doi.org/10.1080/00207543.2020.1823028>
13. Yelles-Chaouche AR, Gurevsky E, Brahimi N, Dolgui A (2021) Reconfigurable manufacturing systems from an optimisation perspective: a focused review of literature. *Int J Prod Res* 59:6400–6418. <https://doi.org/10.1080/00207543.2020.1813913>,
14. Tränkler H-R, Fischerauer G (2014) *Das Ingenieurwissen: Messtechnik*. Springer-Verlag. <https://doi.org/10.1007/978-3-662-44030-8>
15. Chen C, Zhang C, Wang T, Li D, Guo Y, Zhao Z, Hong J (2020) Monitoring of assembly process using deep learning technology. *Sensors* 20:4208
16. Foidl H, Felderer M (2016) Research challenges of industry 4.0 for quality management. In M. Felderer F, Piazzolo W, Ortner L, Brehm H-J Hof (Eds.). *Innovations in Enterprise Information Systems Management and Engineering* (pp. 121–137). Springer International Publishing. [https://doi.org/10.1007/978-3-319-32799-0\\_10](https://doi.org/10.1007/978-3-319-32799-0_10)
17. Doukas C, Chantzis D, Stavropoulos P, Papacharalampopoulos A, Chryssolouris G (2013) Monitoring and control of manufacturing processes: a review. *Procedia CIRP* 8. <https://doi.org/10.1016/j.procir.2013.06.127>
18. Shao C, Paynabar K, Kim TH, Jin JJ, Hu SJ, Spicer JP, Wang H, Abell JA (2013) Feature selection for manufacturing process monitoring using cross-validation. *J Manuf Syst* 32:550–555. <https://doi.org/10.1016/j.jmsy.2013.05.006>
19. Glaessgen E, Stargel D (2012) The digital twin paradigm for future nasa and us air force vehicles. In 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA (p. 1818). <https://doi.org/10.2514/6.2012-1818>
20. Halenar I, Juhas M, Juhasova B, Borkin D (2019) Virtualization of production using digital twin technology. In 2019 20th International Carpathian Control Conference (ICCC) (pp. 1–5). Organization IEEE. <https://doi.org/10.1109/CarpathianCC.2019.8765940>
21. Shafto M, Conroy M, Doyle R, Glaessgen E, Kemp C, LeMoigne J, Wang L (2012) Modeling, simulation, information technology & processing roadmap. National Aeronautics and Space Administration (NASA) 32:1–38
22. Boschert S, Rosen R (2016) Digital twin-the simulation aspect. In: Hehenberger P, Bradley D (eds) *Mechatronic futures: challenges and solutions for mechatronic systems and their designers*. Springer International Publishing, Cham, pp 59–74. [https://doi.org/10.1007/978-3-319-32156-1\\_5](https://doi.org/10.1007/978-3-319-32156-1_5)
23. Cohen Y, Naseraldin H, Chaudhuri A, Pilati F (2019) Assembly systems in industry 4.0 era: a road map to understand assembly 4.0. *Int J Adv Manuf Technol* 105:4037–4054. <https://doi.org/10.1007/s00170-019-04203-1>
24. Kombaya Touckia J, Hamani N, Kermad L (2022) Digital twin framework for reconfigurable manufacturing systems (RMSs): design and simulation. *Int J Adv Manuf Technol* 120:5431–5450. <https://doi.org/10.1007/s00170-022-09118-y>
25. Perzylo A, Grothoff J, Lucio L, Weser M, Malakuti S, Venet P, Aravantinos V, Miny T (2019) Capability-based semantic interoperability of manufacturing resources: a BaSys 4.0 perspective. *International Federation of Automatic Control - IFAC* 52:1590–1596. <https://doi.org/10.1016/j.ifacol.2019.11.427>
26. Järvenpää E, Siltala N, Hylli O, Lanz M (2019) The development of an ontology for describing the capabilities of manufacturing resources. *J Intell Manuf* 30:959–978. <https://doi.org/10.1007/s10845-018-1427-6>
27. Järvenpää E, Hylli O, Siltala N, Lanz M (2018) Utilizing spin rules to infer the parameters for combined capabilities of aggregated manufacturing resources. *IFAC-PapersOnLine* 51:84–89. <https://doi.org/10.1016/j.ifacol.2018.08.239>
28. Keddiss N, Kainz G, Zoitl A (2014) Capability-based planning and scheduling for adaptable manufacturing systems. *Proceedings IEEE Emerging Technology and Factory Automation (ETFA)*. <https://doi.org/10.1109/ETFA.2014.7005213>
29. Hammerstingl V, Reinhart G (2018) Skills in assembly. *Tech Rep*. <https://mediatum.ub.tum.de/1428286>
30. Gonnermann C, Weth J, Reinhart G (2020) Skill modeling in cyber-physical production systems for process monitoring. *Procedia CIRP*, 93, 1376–1381. 53rd CIRP Conference on Manufacturing Systems 2020. <https://doi.org/10.1016/j.procir.2020.03.095>
31. Battaïa O, Dolgui A (2013) A taxonomy of line balancing problems and their solution approaches. *Int J Prod Econ* 142:259–277
32. Hazir ö, Delorme X, Dolgui A (2015) A review of cost and profit oriented line design and balancing problems and solution approaches. *Ann Rev Control* 40:14–24. <https://doi.org/10.1016/j.arcontrol.2015.09.001>
33. Boysen N, Fließner M, Scholl A (2009) Production planning of mixed-model assembly lines: overview and extensions. *Prod Plan Control* 20:455–471. <https://doi.org/10.1080/09537280903011626>
34. Alghazi A, Kurz ME (2018) Mixed model line balancing with parallel stations, zoning constraints, and ergonomics. *Constraints* 23:123–153. <https://doi.org/10.1007/s10601-017-9279-9>
35. Bukchin J, Dar-El EM, Rubinovitz J (2002) Mixed model assembly line design in a make-to-order environment. *Comput Ind Eng* 41:405–421. [https://doi.org/10.1016/S0360-8352\(01\)00065-1](https://doi.org/10.1016/S0360-8352(01)00065-1)
36. Choi G (2009) A goal programming mixed-model line balancing for processing time and physical workload. *Comput Ind Eng* 57:395–400. <https://doi.org/10.1016/j.cie.2009.01.001>
37. Dolgui A, Proth J-M (2010) *Supply chain engineering: useful methods and techniques*. Springer Science & Business Media. <https://doi.org/10.1007/s10601-017-9279-9>
38. Battaïa O, Delorme X, Dolgui A, Hagemann J, Horlemann A, Kovalev S, Malyutin S (2015) Workforce minimization for a mixed-model assembly line in the automotive industry. *Int J Prod Econ* 170:489–500. <https://doi.org/10.1016/j.ijpe.2015.05.038>
39. Biele A, Mönch L (2018) Hybrid approaches to optimize mixed-model assembly lines in low-volume manufacturing. *J Heuristics* 24:49–81. <https://doi.org/10.1007/s10732-017-9357-6>
40. Dolgui A, Kovalev S, Kovalyov MY, Malyutin S, Soukhal A (2018) Optimal workforce assignment to operations of a paced assembly line. *Eur J Oper Res* 264:200–211. <https://doi.org/10.1016/j.ejor.2017.06.017>
41. Khan AS, Homri L, Dantan JY, Siadat A (2022) An analysis of the theoretical and implementation aspects of process planning in a reconfigurable manufacturing system. *Int J Adv Manuf Technol* (pp. 1–32). <https://doi.org/10.1007/s00170-021-08522-0>
42. Leiber D, Hammerstingl V, Weib F, Reinhart G (2019) Automated design of multi-station assembly lines. *Procedia CIRP* 79:137–142. <https://doi.org/10.1080/00207543.2020.1845411>
43. Oesterle J, Amodeo L, Yalaoui F (2019) A comparative study of Multi-Objective Algorithms for the Assembly Line Balancing and Equipment Selection Problem under consideration of Product Design Alternatives. *J Intell Manuf* 30:1021–1046. <https://doi.org/10.1007/s10845-017-1298-2>
44. Sivasankaran P, Shahabudeen P (2014) Literature review of assembly line balancing problems. *Int J Adv Manuf Technol* 73:1665–1694. <https://doi.org/10.1007/s00170-014-5944-y>
45. Leiber D, Reinhart G (2021) A bi-level optimisation approach for assembly line design using a nested genetic algorithm. *Int J Prod Res* 59:7560–7575. <https://doi.org/10.1080/00207543.2020.1845411>

46. Bongomin O, Mwasiagi JI, Nganyi EO, Nibikora I (2020) A complex garment assembly line balancing using simulation-based optimization. *Eng Rep* 2. <https://doi.org/10.1002/eng2.12258>
47. Petroodi SEH, Eynaud ABD, Klement N, Tavakkoli-Moghaddam R (2019) Simulation-based optimization approach with scenario-based product sequence in a reconfigurable manufacturing system (RMS): a case study. *IFAC-PapersOnLine*, 52, 2638–2643. 9th IFAC Conference on Manufacturing Modelling, Management and Control MIM 2019. <https://doi.org/10.1016/j.ifacol.2019.11.605>
48. Kamrani A, Abouel Nasr E, Al-Ahmari A, Abdulhameed O, Mian SH (2015) Feature-based design approach for integrated cad and computer-aided inspection planning. *Int J Adv Manuf Technol* 76:2159–2183. <https://doi.org/10.1007/s00170-014-6396-0>
49. Eigner M, Gerhardt F, Gilz T, Handschuh S (2010) Proposal for a guideline to integrate kinematics within lightweight formats. In *DS 60: Proceedings of DESIGN 2010, the 11th International Design Conference, Dubrovnik, Croatia*
50. Al-wswasi M, Ivanov A, Makatoris HG (2018) A survey on smart automated computer-aided process planning (acapp) techniques. *Int J Adv Manuf Technol* 97:809–832. <https://doi.org/10.1007/s00170-018-1966-1>
51. Garcia F, Lanz M, Järvenpää E, Tuokko R (2011) Process planning based on feature recognition method. In *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM)* (pp. 1–5). <https://doi.org/10.1109/ISAM.2011.5942296>
52. Jones T, Reidsema C, Smith A (2006) Automated feature recognition system for supporting conceptual engineering design. *KES Journal* 10:477–492. <https://doi.org/10.3233/KES-2006-10606>
53. Battaïa O, Dolgui A, Guschinsky N, Levin G (2012) A decision support system for design of mass production machining lines composed of stations with rotary or mobile table. *Robot Comput Integr Manuf* 28:672–680. <https://doi.org/10.1016/j.rcim.2012.04.005>
54. Neb A (2019) Review on approaches to generate assembly sequences by extraction of assembly features from 3d models. *Procedia CIRP* 81:856–861. <https://doi.org/10.1016/j.procir.2019.03.213>
55. Verma A, Rajotia S (2010) A review of machining feature recognition methodologies. *Int J Comput Integr Manuf* 23:353–368. <https://doi.org/10.1080/09511921003642121>
56. Gonnermann C, Zels B, Reinhart G (2021) Automated generation of alternatives for process monitoring in cyber-physical assembly systems. *Procedia CIRP*, 104, 732–737. 54th CIRP CMS 2021 - Towards Digitalized Manufacturing 4.0. <https://doi.org/10.1016/j.procir.2021.11.123>
57. Michniewicz J, Reinhart G, Boschert S (2016) CAD-based automated assembly planning for variable products in modular production systems. *Procedia CIRP* 44:44–49. <https://doi.org/10.1016/j.procir.2016.02.016>
58. Hashemi-Petroodi SE, Gonnermann C, Paul M, Thevenin S, Dolgui A, Reinhart G (2019) Decision support system for joint product design and reconfiguration of production systems. (pp. 231–238). organizationSpringer IFIP International Conference on Adv Prod Manag Syst. <https://doi.org/10.1007/978-3-030-30000-5>
59. Ghandi S, Masehian E (2015). Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design* 67-68. <https://doi.org/10.1016/j.cad.2015.05.001>