Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 79 (2019) 137–142

www.elsevier.com/locate/procedia

12th CIRP Conference on Intelligent Computation in Manufacturing Engineering, 18-20 July 2018, Gulf of Naples, Italy

Automated design of multi-station assembly lines

Daria Leiber[a],*, Veit Hammerstingl[a], Felix Weiß[a], Gunter Reinhart[a]

[a] *Institute for Machine Tools and Industrial Management, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany*

* Corresponding author. Tel.: +49-(0)89-289-15471; fax: +49-(0)89-289-15555. *E-mail address:* daria.leiber@iwb.tum.de

Abstract

Automated assembly lines are a fundamental part of today's manufacturing industry. Due to shortening product life cycles and an increasing number of product variants, assembly lines have to be designed with increasing frequency. Currently, all design decisions are based on the knowledge and expertise of engineers and the design process results in a high amount of manual effort. This paper therefore presents an approach to making the knowledge of these experts explicit and using it as the basis for the automated planning of multi-station assembly lines. The proposed planning method includes the scheduling of the assembly line and the selection of resources and their positioning into a feasible layout.

Keywords: Assembly line design; Resource positioning; Capability modelling; Knowledge-based planning

1. Introduction

Automated assembly lines are a fundamental part of today's manufacturing industry. Small mechatronic products, like cameras or loudspeakers, are often produced by fully automated systems. Besides the minimization of labor costs, this also allows for highly accurate, stable processes and high production output. Due to shortening product life cycles and an increasing number of product variants, automated assembly lines have to be designed or redesigned with increasing frequency [1]. This includes the scheduling of the process chain and the selection of the resources and their placement [2]. Performing these planning steps requires lots of expert knowledge and results in a high amount of manual effort. Research in the field therefore tries to automatize different aspects of the planning process. However, the concepts usually don't integrate all the different stages of designing a multi-station assembly line into one framework. The focus of existing methodologies often lies on scheduling and resource selection, while the geometric positioning of the resources is excluded [3]. Current research concerning positioning usually focuses on finding the optimal position for one particular type of resource (often an industrial robot) performing a specific task [4, 5].

This paper presents an approach for automatically designing fully automated multi-station assembly lines that includes the geometric positioning of the resources.

1.1. Related Works

The field of research called assembly line balancing (ALB) focuses on the scheduling of assembly lines. It studies how to assign production tasks to the different stations of an assembly line while optimizing different objectives like the number of stations, the cycle time or the cost of the line [6, 7]. Some of the presented methods also take into account the selection of resources to perform the tasks [3, 8]. However, these approaches only consider the economic characteristics of the resources and do not look at their technical suitability.

Other authors focus on selecting technically feasible resources for a certain production process. One of the challenges thereby is the data consistency and the information modelling [9]. The basis for automated selection is a data model that includes a description of the domain product, process, and resource (PPR-model) and a specification of the

10.1016/j.procir.2019.02.029

constraints between them [10]. Such models are deployed in different use cases. They have been used as the basis for a task-oriented programming system [11, 12], the reconfiguration of production systems [13, 14] and plug & produce applications [15]. The problem of semantic determination can be addressed by the use of taxonomies that classify the terms of a knowledge domain [16]. Similarly, the emergence of new capabilities when resources are combined can be described with the use of an ontology [17].

Research in the area of positioning has been mostly focused on the positioning of robots for specific tasks [18–20]. Some publications that also take into account other components of an assembly cell can be found [21, 22], but even here the transport between different stations is not considered. Therefore, the automated design of 3D-layouts for multi-station assembly lines is still one of the major challenges to be tackled [3].

Geometric planning is a complex field with multifaceted interdependencies between different planning decisions. There is also often no analytic way to model dependencies and relations. For automated planning and optimization in these situations, evolutionary algorithms are a suitable tool [23]. They allow for optimization simply by assessing possible drafts without requiring explicit modelling of the relationship between variables to be optimized and the resulting outcome.

1.2. Scope of application

The presented methodology addresses the problem of automatically designing an assembly line for the production of a new product. The input for any planning scenario is the production sequence that has to be performed and the maximum cycle time per station. It is assumed that the assembly line is fixedly linked.

The concept's anticipated usage is by companies who want to decrease their effort and cost for each new designing scenario. Currently, engineers make the design decisions based on their knowledge and expertise. The presented system allows one to make the knowledge of these experts explicit and machine readable and use it as the bases for future planning scenarios. The relevant domains of knowledge are thereby the production processes themselves, potential resources to be used and the logical conditions for the selection of resources.

2. Methodology Overview

The proposed approach consists of five main planning steps, a data structure and a database (Fig. 1):

The proposed approach consists of five main planning steps, a data structure and a database (Fig. 1):

(1) First, a process that has to be executed is defined and entered by the user. He thereby has to specify the order of the process steps, all their relevant properties and the product parts that are affected. He also has to enter general requirements like the maximal cycle time for the line.

(2) Based on the user's input, the assembly line is then automatically balanced, taking into account the economic characteristics of the suitable resources.

(3) The suitable resources for the line are chosen by comparing the resources' capabilities with the processes requirements. This takes into account quantitative characteristics as well as geometric aspects. As the planning decisions (2) and (3) are closely interlinked, they have to be performed iteratively.

(4) Subsequently, the positioning of the chosen resources is automatically optimized, taking into account different characteristics to determine a potential solution's quality.

(5) Finally, the resulting configuration's functionality is validated via a rigid body simulation. If no feasible draft could be generated, the planning process starts anew with steps (2) and (3).

(6) To convey the necessary input information for the planning scenario and to ensure semantic determinacy, a data structure is included in the approach. The data structure includes the classical aspects product, process and resource of a PPR-Model, supplemented by a structure for the constraints used for the selection of resources.

(7) Available resources as well as predetermined tasks are stored in a database to be used in the planning scenario.

3. Data Structure

As discussed before, extensive knowledge about the process that needs to be executed, the product that has to be assembled and the resources that are available is a mandatory precondition for designing a new assembly line. It is also necessary to know how the resources' capabilities and the processes' requirements are to be matched. A consistent data structure is necessary to make this information accessible for automated planning via an algorithm and to ensure semantic determinacy.

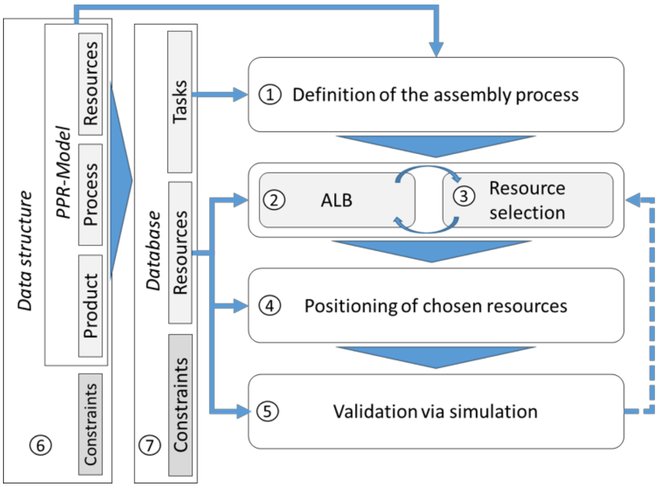

Fig. 1. Concept overview

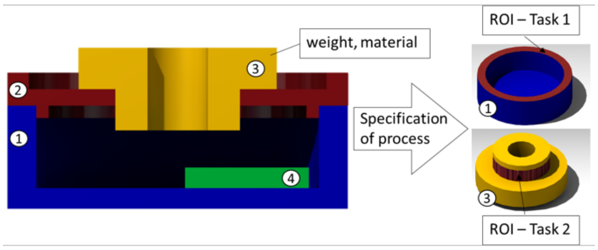

Fig. 2. Example of a product model

3.1. Product model

Along with the process and resources, the product itself can be seen as an input variable for the planning of an assembly line. The relevant information are thereby characteristics like weight or demanded surface quality as well as the geometric shape of the product in its different assembly stages. It is also relevant to consider at which point of the product the region of interest (ROI), where the process takes place, is located. The proposed product representation considers these aspects. It is based on a 3D-CAD model of the complete product that consists of the different parts that are to be assembled. These parts also contain data about their characteristics like their weight or material. When the assembly process is specified in the beginning of a planning event, the parts are numbered and augmented with information concerning the ROI of the respective process (Fig. 2).

3.2. Process model

To optimally combine the need for semantic determinacy with high flexibility for the experts to model their knowledge about different assembly processes, the modeling follows an object-oriented approach. The data structure provides an abstract model for a task that can be instantiated and filled with specific information (expert knowledge) about a certain type of task. This instantiation doesn’t require manipulation of source code. The instances are then saved in a database, and can later be copied to model the production process in a specific planning scenario (Fig. 3).

This modeling approach guarantees maximal flexibility, as all the process types are objects of the same class task and can be customized (e. g. by adding parameters) at any time without the strict restrictions of inheritance. However, a taxonomy-like structure is still integrated in the proposed data model. Each task has a link to its parent-task and links to possible child-tasks. This allows users to dynamically generate their own taxonomy by adding task types to the database and specifying their place in the taxonomy. The class also has another list of references to other tasks: the sub-tasks. A task’s relationship with its subtasks doesn’t correlate with inheritance, but with aggregation. The subtasks represent the different steps that, in sum, add up to the task itself (e. g. the task gluing might have the subtasks surface preparation, glue application and drying). Here it has to be noted that the same task can be used as a subtask by different tasks, while each task can only be a child-task to one parent-task.

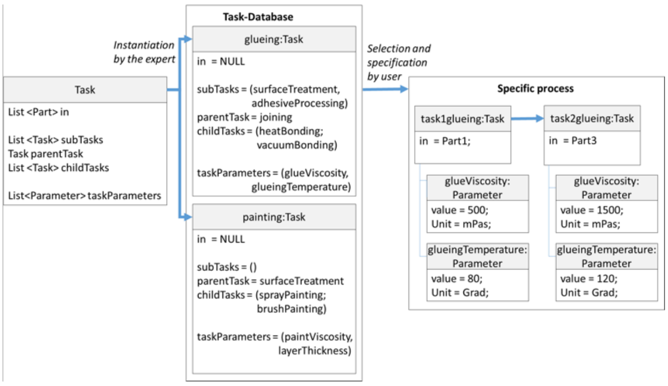

Fig. 3. Process model

3.3. Resource model

Besides tasks that can be used as building blocks to describe the actual manufacturing process of one specific product, the second necessary input for the system is a collection of all the available resources. The data structure is similar to the structure used for modelling the tasks. A class resource is instantiated to generate objects that represent the resources. As with the tasks, the resources can be classified in a taxonomy to allow for the derivation of properties from existing resources when creating a new one. Analogous to the task model, a resource can also be aggregated from other resources.

A resource’s parameters are its characteristics that are necessary for assembly-line balancing and the selection of resources to create the line (e. g. physical properties, economic properties, abilities…). To generate the assembly line’s 3D-layout and to position the individual resources, purely quantitative information isn’t sufficient. To enable the positioning, the first relevant aspect is a model of the resource’s own geometry and kinematics to ensure that the created layout is collision free. For all direct resources that interact with the product directly (actuators as well as sensors), a workspace has to be defined. The workspace describes the volume where the resource can perform its task, but the possible positions inside the workspace are not equal. They can be evaluated and assigned a position quality value that describes how favorable a position is. The criteria used for the evaluation of the position is different for different resource types. For a kinematic resource like an industrial robot, the position quality could indicate the strain reaching a certain position puts on its axes and for sensors it could correspond with the error rate at a certain position. The resource models also need to contain information about their interfaces to their environment (e. g. which surfaces need to be mounted or where other resources can be mounted on them). In the case of a locomotive actuator (e. g. an industrial robot or a linear axis), a description of its kinematics is also needed (Fig. 4).

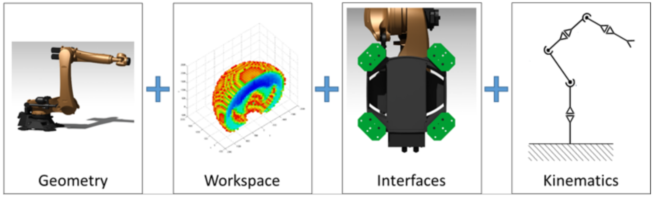

Fig. 4. Resource model of an industrial robot

3.4. Constraints

For the selection of the resources, their characteristics have to be matched with the processes' requirements. Constraints are a data structure for describing the conditions of this matching. They have defined input variables that refer to a resource's or process's parameters and a method for comparing these input values that returns whether the condition is satisfied. The implemented logical function can be as simple as determining whether input A is greater than input B or it can contain complex mathematical expressions.

4. Database

Based on the data structure presented above, the experts in a company can save their knowledge to a database and make it available as the basis for future automated planning scenarios. As these engineers usually don't have a background in software engineering, an intuitive user interface for them to insert their knowledge is provided. In accordance with the explained data structure, they can define tasks, resources and constraints and save them to the database. Concerning the tasks, they can define task parameters, as well as their sub- and child-tasks. This way, over time a customized taxonomy is formed. Alternatively, it would also be possible to use an established taxonomy like the one presented in [16] as a starting point and then customize it. The resource-models can be largely provided by their manufacturer and only self-designed resources have to be saved to the database.

5. Detailed Steps of the Methodology

5.1. Definition of the assembly process

In the first planning step, the assembly process that has to be executed and the product that is to be assembled by the line are uploaded by the user. The users thereby employs the predetermined tasks stored in the database as building blocks for the description of the production process and use their parameters to determine the requirements of the production process. For each process, they also specify which parts of the product are manipulated and where the ROI is located.

5.2. Assembly line balancing and selection of resources

Based on the description of the process entered by the user and the available resources with the description of their functionality in the database, the first two planning decisions, the scheduling of the line and the selection of suitable resources for the tasks are carried out.

The allocation of tasks to different production cycles (or stations of the assembly line) has already been thoroughly studied [6, 8] and will not be discussed further at this point. However, as an input for these scheduling algorithms, the resources need to be preselected considering technical criteria. The selection is thereby conducted from a high level of abstraction to a lower level. For each task, the algorithm first checks whether a resource is available that can perform the whole task. Only if this fails, are the different sub-tasks looked at recursively in the same way. For a resource to be considered suitable for a process, it has to have abilities that match the task. Furthermore, all the constraints that interlink the process's parameters to the resource's capabilities have to be satisfied.

5.3. Positioning

After choosing resources for the different tasks and grouping them into stations, the next planning step is to find an optimized position for the chosen resources in the assembly line. Thereby, two main aspects can be separated. The first one is the positioning of the stations (or more precisely the product's holding points) and the second is the positioning of the individual resources assigned to a station.

For the positioning of the stations, the implementation of the material transfer is of major significance. If the product is moved from one station to the next on a conveyor belt, the line's structure will be quite different from a line with stations arranged around a rotary table. Choosing the material flow concept is therefore the basis for choosing the stations' positions and the first decision to be reached. Two approaches to this problem are possible: automatically adding *handling* tasks to the task sequence and select corresponding resources, or choosing them independently from the process model. For reasons of simplicity, the latter approach is used. The selection of a material transfer system restricts the positioning of the stations relative to one another (e. g. a conveyer belt means that stations are to be located at the same height, preferably in a straight line).

Besides the resources used for the material transfer, the remaining resources can also be classified into different categories. *Direct resources* are those resources whose positioning influences the cycle time and the quality of the manufacturing process directly (e. g. a sensor or a robot with its end effector). *Indirect resources* also influence the cycle time of the assembly line, because they indirectly affect the movements and actions of the direct resources. These resources' placement is restricted by the work area of the direct resources they interact with (e. g. a magazine that provides parts to be assembled must be placed in the work-area of the robot that takes the parts). *Peripheral resources* are those that are necessary for the functionality of the line but don't face any strict placement conditions. An example would be a control system.

For the positioning of the individual resources, an evolutionary optimization algorithm is used. This type of optimization algorithm is a stochastic optimization approach and well suited for complex problems where an exact, analytical solution is not possible in a reasonable timeframe [23]. The basis for an evolutionary optimization algorithm is a population of so-called individuals that represent potential solutions. The individuals of the initial population are usually randomly generated. In an iterative process over multiple generations, the individuals are evaluated with a fitness function and modified (recombined and mutated) to form the next generation. Individuals with better fitness are thereby favored. In the following, the optimization problem for the positioning of the resources is described.

1) *Representation of the individuals*

When optimizing the layout of an assembly line, the potential solutions are 3D layout drafts. These individuals are then encoded into a genotype G. G consists of n poses P_{Prod}, each describing the positioning and orientation of the product in one of the stations and of m poses P_{Res} describing the positioning and orientation of a resource, as its chromosomes. The poses themselves are composed by three Cartesian coordinates x, y, z that define the position and the Euler angles α, β, γ that describe the orientation. As the poses of the resources are thereby not expressed in an absolute coordinate system but relative to other objects in the drafted layout, each chromosome P also includes a variable P_{rel} that defines its reference pose.

$$G = (P_{Prod,1}, P_{Prod,2}, \dots P_{Prod,n}, P_{Res,1}, P_{Res,2}, \dots P_{Res,m})$$
$$mit\ P_i = (x_i, y_i, z_i, \alpha_i, \beta_i, \gamma_i, P_{rel,i}),\ i = 1 \dots n,\ 1 \dots m \tag{1}$$

As the logical reference point for the direct resources in a station is the product (or more exactly the process's ROI on the product), it makes sense to express their poses relative to the product. Indirect resources' positions are described relative to one of the direct resources they interact with. Peripheral resources are positioned in a global coordinate system with P_{rel} referring to the surroundings.

2) *Boundary conditions*

The constraints for the placement of the stations P_{Prod} follow from the type of material transfer system that is chosen. In the case of a conveyer belt that only consists of straight elements and 90° turns, for example, one of the following conditions needs to be fulfilled.

$$\begin{aligned} &P_{Prod,i+1} = (x_{x+1}, 0, 0, 0, 0, 0, P_{Prod,i})\ or \\ &P_{Prod,i+1} = (0, y_{i+1}, 0, 0, 0, 0, P_{Prod,i})\ or \\ &P_{Prod,i+1} = (y_{i+1}, z_{i+1}, 0, 0, \pm\pi/2, P_{Prod,i})\ \forall i = 1 \dots n-1 \end{aligned} \tag{2}$$

For other material transfer systems like round tables the constraints of the product positions can be adjusted according to their characteristics.

Direct resources are only able to perform their tasks if the product (or, respectively, the ROI) is inside their workspace. The first boundary condition of the optimization problem is therefore that the ROI has to be in the workspace WS of each direct resource i.

$$P_{ROI} \in WS_{Direct,i}\ \forall i = 1 \dots m_{Direct} \tag{3}$$

Indirect resources don't need to have the product in their workspace. However, their placement is restricted by the work area of the direct resources they interact with (e. g. a magazine that provides parts to be assembled must be placed in the workspace of the robot that takes the parts). The second boundary condition is therefore that a indirect resource has to be placed in the workspace of all the direct resources it interacts with.

$$P_{Indirect} \in WS_{Direct} \forall i = 1 \dots m_{Indirect} \tag{4}$$

For peripheral resources, no such restrictions that limit their positioning relative to other objects exist.

Another boundary condition is the collision freeness of all the resources. It is immediately evident that two production resources can't be placed at the same position in space. Therefore, all solutions that aren't collision free are simply not feasible and collision freeness is a hard constraint.

$$Vi \cap Vj = \{\}\ \forall i, j = 1 \dots m,\ i \neq j \tag{5}$$

It has to be noted that collision freeness has to be ensured not only for the initial position, but also for the movements the resource has to execute. The maximal cycle time is also a condition that needs to be honored. However, designs that violate this condition are still physically feasible. It is therefore a soft constraint.

3) *Fitness function*

The fitness function is used to evaluate different solution candidates. As there isn't a single characteristic that accurately represents a solution's quality, an aggregation function that combines different criteria is used. Before an individual can be assessed, the genotype has to be decoded. In the case of designing an assembly line, this means building a 3D model of the line according to the indicated resource poses in the genotype.

One of the main quality criteria is the positioning of the direct resources relative to the product, because they directly influence quality. As described before, all the possible positions inside the workspace are therefore assigned a *position quality* value that describes how favorable a position is. This value PQ_{Direct} is the first criterion used to determine the fitness of an individual.

$$PQ_{Direct} = \Sigma PQ_i \tag{6}$$

The positioning of the indirect resources influences the draft's quality in two ways. First, their position relative to the direct resources they interact with can be evaluated. Analogous to the position quality of direct resource, this is done by judging the quality of the direct resource's workspace at the indirect resources position. Besides, the indirect resources' distance to the product is relevant because the cycle time directly correlates with the distance the direct resource has to travel.

$$PQ_{Indirect} = \Sigma PQ_i + d_{Product} \tag{7}$$

A second metric besides the position quality for assessing a solution candidate is the *mounting cost*. It describes how much effort any chosen positioning for a resource causes. Ideal would be a position directly on the ground or a suitable interface of another resource. Worst would be a position where a support structure has to be individually designed. The basis for the calculation of the mounting cost are the mounting interfaces that are defined in the resource models.

$$MC = \Sigma MC_i \tag{8}$$

In conclusion, the fitness function f can be expressed as a weighted aggregation of these characteristics.

$$f(x) = PQ_{Direct} + PQ_{Indirect} + MC \tag{9}$$

4) *Solution approach*

The first step is the random generation of an initial population of individuals. To ensure that the boundary conditions (2) to (4) are met, the solution space is limited accordingly. The stations' poses P_{Prod} are chosen only among poses that satisfy (2) and for the resources' poses P_{Res}, only those positions located in the respectively relevant workspace are considered. The fourth boundary condition (5) is ensured by the use of a death penalty approach, meaning that individuals that are not feasible are killed and not considered further.

The remaining individuals of the initial population are evaluated with the fitness function and then used as the basis for further generations that are generated by recombination and mutation. Thereby, the fitter individuals are favored. After a defined number of iterations, the optimization is ended.

5.4. Validation by simulation

After the optimization, the best individuals are evaluated in a rigid body simulation to validate the feasibility. Thereby, the simulation model is set up automatically based on the positioning information in the solution candidate's genotype. The control code is also automatically generated with the use of a task-oriented programming system.

In the simulation, whether the reachability is achieved and the requested cycle time can be reached are evaluated. The feasible assembly line drafts are then shown to the user. If no viable solution candidate can be found, the process starts again, with the selection of resources thereby choosing different resources for the station that was not feasible in the original configuration.

6. Conclusion and Outlook

Despite modern technological possibilities, designing a multi-station assembly line still results in a high amount of manual effort. This paper presents a concept for supporting the designers of such lines by automatizing different planning decisions. First, it demonstrates a way to save the knowledge of experienced engineers to make it available for future use. It then presents a concept for automatizing the planning steps of scheduling a line and selecting the resources and positioning them. The final layout is then validated via simulation.

The presented concept is set up for the use case of planning new assembly lines. Based on the theoretical considerations presented in this paper, the next step will be the exemplary application of the method in different case studies. Another prospect is the extension of the concept to not only suit new planning scenarios, but also the reconfiguration of existing assembly lines.

Acknowledgment

The authors of this paper would like to thank the German Federal Ministry of Education and Research (BMBF) and the Project Management Agency Research Center Karlsruhe (PTKA) for funding this work, which is part of the research and development project ProMoA (funding number 02P15A100).

References

[1] Abele E, Reinhart G. Zukunft der Produktion: Herausforderungen, Forschungsfelder, Chancen. Carl Hanser Fachbuchverlag, 2011.

[2] Bullinger H-J, Ammer D. Systematische Montageplanung: Handbuch für die Praxis. München: Hanser, 1986.

[3] Michalos G, Fysikopoulos A, Makris S, Mourtzis D, Chryssolouris G. Multi criteria assembly line design and configuration – An automotive case study. CIRP Journal of Manufacturing Science and Technology, vol. 9, pp. 69–87, 2015.

[4] Lim ZY, Ponnambalam SG, and Izui K. Nature inspired algorithms to optimize robot workcell layouts. Applied Soft Computing, vol. 49, pp. 570–589, 2016.

[5] Zhang J, Fang X. Challenges and key technologies in robotic cell layout design and optimization. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, vol. 231, no. 15, pp. 2912–2924, 2016.

[6] Boysen N, Fliedner M, Scholl A. Assembly line balancing: Which model to use when?. International Journal of Production Economics, vol. 111, no. 2, pp. 509–528, 2008.

[7] Hazır Ö, Delorme X, Dolgui A. A review of cost and profit oriented line design and balancing problems and solution approaches. Annual Reviews in Control, vol. 40, pp. 14–24, 2015.

[8] Oesterle J, Amodeo L, Yalaoui F. 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, vol. 57, no. 2, p. 415, 2017.

[9] Vogel-Heuser B, Sarda-Espinosa A. Current status of software development in industrial practice: Key results of a large-scale questionnaire. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), Emden, 2017, pp. 595–600.

[10] Pfrommer J, Schleipen M, Beyerer J. PPRS: Production skills and their relation to product, process, and resource. In: 2013 IEEE 18th Conference on Emerging Technologies & Factory Automation (ETFA), Cagliari, Italy, 2013, pp. 1–4.

[11] Backhaus J, Reinhart G. Digital description of products, processes and resources for task-oriented programming of assembly systems. J Intell Manuf, vol. 28, no. 8, pp. 1787–1800, 2017.

[12] Backhaus J, Ulrich M, Reinhart G. Classification, Modelling and Mapping of Skills in Automated Production Systems. In: Zaeh MF, editor. Enabling Manufacturing Competitiveness and Economic Sustainability, Cham: Springer International Publishing, 2014, pp. 85–89.

[13] Michniewicz J, Reinhart G, Boschert S. CAD-Based Automated Assembly Planning for Variable Products in Modular Production Systems. Procedia CIRP, vol. 44, pp. 44–49, 2016.

[14] Michniewicz J et al. Automatisierte digitale Anlagenplanung: Methode für die automatisierte Planung von Montageanlagen auf Basis digitaler Produktdaten. wt Werkstattstechnik online, vol. 107, no. 9, pp. 582–589, 2017.

[15] Hammerstingl V, Reinhart G. Unified Plug&Produce architecture for automatic integration of field devices in industrial environments. In: 2015 IEEE International Conference on Industrial Technology (ICIT), Seville, 2015, pp. 1956–1963.

[16] Hammerstingl V, Reinhart G. Skills in Assembly. 2018.

[17] Jarvenpaa E, Luostarinen P, Lanz M, Tuokko R. Presenting capabilities of resources and resource combinations to support production system adaptation. In: 2011 IEEE International Symposium on Assembly and Manufacturing (ISAM), Tampere, Finland, 2011, pp. 1–6.

[18] Pellegrinelli S, Pedrocchi N, Tosatti LM, Fischer A, Tolio T. Design and motion planning of body-in-white assembly cells. In: 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, Chicago, IL, USA, 2014, pp. 4489–4496.

[19] Tubaileh AS. Layout of robot cells based on kinematic constraints. International Journal of Computer Integrated Manufacturing, pp. 1–13, 2014.

[20] Papakostas N, Alexopoulos K, Kopanakis A. Integrating digital manufacturing and simulation tools in the assembly design process: A cooperating robots cell case. CIRP Journal of Manufacturing Science and Technology, vol. 4, no. 1, pp. 96–100, 2011.

[21]Milberg J, Woenckhaus C. Rechnergestütztes System zur automatisierten 3D-Layoutoptimierung. Berlin, Heidelberg: Springer 1994.

[22] Roßgoderer U. System zur effizienten Layout- und Prozessplanung von hybriden Montageanlagen. München: Utz, 2002.

[23] Weicker K. Evolutionäre Algorithmen. 3rd ed. Wiesbaden: Springer Vieweg, 2015.