Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 107 (2022) 28–33

www.elsevier.com/locate/procedia

55th CIRP Conference on Manufacturing Systems

Novel heuristic approach to integrating task sequencing and production system configuration

Daisuke Tsutsumi[a,*], András Kovács[b], Ádám Szalóki[b]

[a]*Hitachi Ltd., Research & Development Group, 292 Yoshida-chyo, Totsuka-ku, Yokohama-shi, Kanagawa, 244-0817, Japan*
[b]*EPIC Centre of Excellence in Production Informatics and Control, SZTAKI, Kende 13-17, H-1111 Budapest, Hungary*

* Corresponding author. Tel.: +81-70-7013-7229; fax: +81-45-443-9447. *E-mail address:* daisuke.tsutsumi.jh@hitachi.com

Abstract

The robot system integration (SI) business targets the construction of production lines for various products that differ from customer to customer. Improving the efficiency of line design is important for increasing profits in the SI business. Especially in the construction of assembly lines, the optimal system configuration, with an appropriate combination of robotic and human stations, is critical for maximizing the return on investment of customers. Yet, the optimal system configuration depends heavily on the task sequence in the process plan. In this paper, task sequencing and system configuration are regarded as two interdependent sub-problems of line design. The exact joint optimum of the two problems can be found only if they are solved together, in an integrated way. However, this is computationally intractable for problems of industrially relevant sizes. To overcome this challenge, a heuristic method exploiting industrial knowledge is proposed for solving the two problems simultaneously. The heuristic constructs promising task sequences according to the capabilities of the applicable resources, considering aspects such as, among others, human work quality or robot tool change times. Then, a close-to-optimal system configuration is computed by considering the promising task sequences and their various relevant combinations. The proposed method is illustrated in a case study on the assembly of electrical components in the automotive industry.

Keywords: task sequencing; system configuration; assembly lines; optimization

1. Introduction

With the declining working-age population in many developed countries, the importance of automation and robotization in the manufacturing industry is increasing. Estimated annual sales of industrial robots worldwide increased by more than 10% every year from 2015 to 2018, and robot utilization in the automation market is expected to grow by an average of 14.4% annually from 2018 to 2023 [7].

In robot system integration (SI), which builds production lines for assembly and welding processes utilizing industrial robots, it is important to coordinate process planning with customers based on product and production conditions for each customer project. After that, the line is designed and manufactured by solving the control, equipment, and layout design problems. Hence, line design is the process of determining the optimal system configuration and manufacturing process in the engineering chain, and this procedure has a great impact on production efficiency.

However, line design, which takes into account a huge variety of resource candidates and task allocations, is said to be one of the areas in the engineering chain where automation has been delayed [1], and it still depends on the know-how of engineers. In order to meet the increasing demand for automation projects, technology for deriving the optimal system configuration at an early stage is required. Therefore, in this research, the development of automated line design technology that optimizes the system configuration and task allocation of production lines is investigated.

Within the line design workflow, process planning defines the set of production tasks, together with the interdependencies between them, that are required to manufacture the product according to its specification. Task sequencing departs from these predefined tasks and the precedence constraints between them, and determines a complete order of the production tasks. System configuration is responsible for computing the combination of resources, as well as the assignment of those resources to production tasks, in order to satisfy the given demand in the most efficient way. The geometrical layout of the produc-

2212-8271 © 2022 The Authors. Published by Elsevier B.V.

10.1016/j.procir.2022.04.005

tion system, as well as the micro-level process plan (e.g., robot paths) are determined in downstream planning steps.

Yet, task sequencing and system configuration are mutually interdependent: the optimal task sequence depends on the system configuration, and vice versa, the optimal system configuration depends on the task sequence. This means that the exact joint optimum of the two problems can be found only if they are solved together, in an integrated way.

Approaches to task sequencing include methods that depart from the product CAD model and the available knowledge about the assembly process [6], as well as methods that capture the problem as a constrained optimization problem built up automatically from a generic feature-based representation of the product and the assembly process [8].

System configuration has been studied mostly for fully human or fully robotized systems [2, 3]. However, there is an increasing need to build production lines that combine human and robot workforce [11]. Production lines are required that utilize various production resources, such as industrial robots of various sizes, mechanisms and price ranges, as well as workers with different skills and labor costs depending on the production area. The authors have also presented results on system configuration combining human and machine resources [10]. Moreover, classical methods for system configuration and the related assembly line balancing problem fail to capture the choice from different execution modes for each task, and accordingly, the variation of equipment within the different cells (e.g., robots and tools), which greatly affects the cost of the cell [4, 9]. These are all drastic simplifications of the configuration problem arising in an industrial setting. To overcome this challenge, this paper proposes a heuristic method for finding a close-to-optimal joint solution of the task sequencing and the system configuration problems from a huge combination of candidate task sequences and various resource candidates for executing the tasks.

2. Problem statement

There problem consists in configuring a dedicated production line for producing a given product, serving the forecasted demand of D over the planning horizon. The process plan is given in the input, and consists of a set of tasks connected by precedence constraints. The ensemble of precedence constraints forms a directed acyclic graph. Yet, there are alternative execution modes for each task, which require a different mix of resources and have different durations. A typical example of alternative execution modes is a screwing task that can be executed either by a human operator quickly using a manual screw driver, or by a robot at a speed depending on the robot type using a suitable robotic screw driver with a bolt picker. Different tasks can be executed in different modes.

The production line consists of a series of human or robotic stations, equipped with various tools. A single human operator or a single robot can serve a station. Planning departs from a given initial line configuration, which contains the ensemble of resources installed along the line. As a special case, the initial configuration can be an empty plant, which corresponds to planning from scratch. These initial resources cannot be removed or re-assigned to another station. The investment cost for building a new station, purchasing a robot of type r or a tool of type j is denoted by K^C, K^R_r, and K^J_j, respectively.

Each station of the line executes a subset of the tasks in the process plan. In contrast, each task must be assigned to exactly one station, i.e., parallel stations or executing a fraction of a task are not allowed. A station can execute a task τ in a given mode m if the station is equipped with all the required resources (robot, tool, or human operator). Then, executing this task takes $T_{\tau m}$ processing time. Moreover, if the task requires a tool different from the tool required by the previous task, then a changeover time of U_m is also necessary. For the *first* task executed in the station on a product, the *previous* task corresponds to the *last* task of the previous piece. The cycle time of the overall production line is determined by the slowest station, and it must be sufficiently low to serve the complete demand in a time period of length Θ.

The total investment cost I is calculated as the sum of the purchase price of the base stations, the robots and the tools. Linear depreciation is applied on all investments with a useful life of $\mathcal{D}$. Then, the objective is minimizing the investment depreciation I^D (for hardware resources) plus the labor cost L (for human resources) while satisfying the demand. The notation is summarized in Table 1.

Indices	
τ	Task (index)
s	Station (index)
m	Execution mode (index)
r	Robot type (index)
j	Tool type (index)
Input parameters	
D	Product demand [pcs]
$R(\tau, m)$	Robot type req'd by task τ in exec. mode m (index)
$J(\tau, m)$	Tool type req'd by task τ in exec. mode m (index)
$H(\tau, m)$	Human operator is req'd by task τ in exec. m (boolean)
$T_{\tau m}$	Processing time of task τ in mode m [sec]
U_m	Tool changeover time in mode m [sec]
Θ	Length of the planning horizon [sec]
K^C	Investment cost of a base station [yen]
K^R_r	Investment cost of robot type r [yen]
K^J_j	Investment cost of tool type j [yen]
$\mathcal{D}$	Useful life of resources for depreciation [time periods]
K^L	Labor cost [yen per period and worker]
α^0_s	Station s is built initially (binary)
ϱ^0_{sr}	Station s is equipped with robot type r initially (binary)
δ^0_{sj}	Station s is equipped with tool type j initially (binary)
Decision variables	
$x_{s\tau m}$	Task τ is assigned to station s in exec. mode m (binary)
α_s	Station s is built (binary)
ϱ_{sr}	Station s is equipped with robot type r (binary)
δ_{sj}	Station s is equipped with tool type j (binary)
γ_s	Station s is equipped with a human operator (binary)
$u_{s\tau}$	Changeover time before task τ on station s [sec]
Cost functions	
I	Total investment cost [yen]
L	Total labor cost [yen]

Table 1. Notation.

It is noted that some practical, but mathematically straightforward features of the implemented system are omitted in the paper for the sake of brevity, such as enabling or disabling certain types of investments or execution modes, considering additional equipment (e.g., storages), or a multi-period model that allows configuring a system that changes over time. Most importantly, the system configuration model captures multiple products assembled together in a common production system, on multiple lines. Yet, the task sequencing approach is implemented for a single product only, and hence, we stick to this reduced problem in the paper.

3. Solution approach

3.1. Overview

The proposed method combines the following two basic steps: (i) Finding the optimal task sequence for a given system configuration, and (ii) finding the optimal system configuration for a given task sequence. The two steps and their integration are introduced in the following subsections.

3.2. Task sequencing sub-problem

By heuristics based on the know-how of engineers, the generation of the optimal task sequence for a fixed system configuration, i.e., for a given number of stations in the line and the same given main resource (human or robot) at each station, is performed with special attention to the different characteristics of human and robotic resources.

Human operators can flexibly respond to the changes in part shapes and tool requirements between subsequent tasks, and hence, the impact of part geometry and tool changeovers on cycle times is limited. However, from the viewpoint of work quality, continuous work on spatially close parts can maintain concentration and high quality based on the knowledge of human work so far [5]. Accordingly, in human work, the positional relationship of parts is captured by minimizing the total center-of-gravity (CoG) distance of parts assembled during subsequent tasks. For computational efficiency, the Manhattan distance is used.

At the same time, *robots* work with high repetitive accuracy in the whole operating range, and the effect of task sequence on quality is negligible. On the other hand, robotic grippers must match part shapes, which leads to significant tool changeover times. Hence, in robot work, the minimization of tool changeover times is addressed.

In both cases, the problem of finding the optimal task sequence involves ordering the tasks and assigning continuous sections of this sequence to the predefined number of stations in such a way that the precedence constraints are respected, while the cycle time, i.e., the highest total processing time (plus tool changeover times in case of robotic stations; plus the penalty for total part distance in case of human stations) over the different stations is minimized. This task sequencing problem has been encoded into a mixed-integer linear program (MILP). The presentation of this MILP is omitted in this paper due to limited space.

3.3. System configuration sub-problem

The system configuration problem with with fixed task sequence can be encoded into a MILP as displayed below:

Minimize

$$I^D/\mathcal{D} + L \tag{1}$$

subject to

$$\sum_{s,m} x_{s\tau m} = 1 \qquad \forall\, \tau \tag{2}$$

$$\alpha_s \geq x_{s\tau m} \qquad \forall\, s,\tau,m \tag{3}$$

$$\varrho_{sR(\tau,m)} \geq x_{s\tau m} \qquad \forall\, s,\tau,m \tag{4}$$

$$\delta_{sJ(\tau,m)} \geq x_{s\tau m} \qquad \forall\, s,\tau,m \tag{5}$$

$$\gamma_s \geq x_{s\tau m} \qquad \forall\, s,\tau,m : H(\tau,m) \tag{6}$$

$$\alpha_s \geq \alpha_s^0 \qquad \forall\, s \tag{7}$$

$$\varrho_{sr} \geq \varrho_{sr}^0 \qquad \forall\, s,r \tag{8}$$

$$\delta_{sj} \geq \delta_{sj}^0 \qquad \forall\, s,j \tag{9}$$

$$u_{s\tau} \geq U_m(x_{s(\tau-1)m} + x_{s\tau m} - 1) \qquad \forall\, s,\tau,m:\; J(\tau,m) \neq J((\tau-1),m) \tag{10}$$

$$u_{s\tau} \geq U_m(x_{s\tau m} + x_{s\tau' m} - x_{s(\tau-1)m} - x_{s(\tau'+1)m} - 1) \qquad \forall\, s,\tau,\tau',m:\; J(\tau,m) \neq J(\tau',m) \tag{11}$$

$$\sum_{m} x_{s\tau m} \leq \sum_{s' \leq s, m} x_{s'(\tau-1)m} \qquad \forall\, s, \tau > 1 \tag{12}$$

$$D\left(\sum_{\tau} u_{s\tau} + \sum_{\tau, m} T_{\tau m} x_{s\tau m}\right) \leq \Theta \qquad \forall\, s \tag{13}$$

$$I^D = \sum_{s}(K^C(\alpha_s - \alpha_s^0)) + \sum_{s,r}(K_r^R(\varrho_{sr} - \varrho_{sr}^0)) + \sum_{s,j}(K_j^J(\delta_{sj} - \delta_{sj}^0)) \tag{14}$$

$$L = \sum_{s}(K^L \gamma_s) \tag{15}$$

$$x_{s\tau m},\ \alpha_s,\ \varrho_{sr},\ \delta_{sj},\ \gamma_s \in \{0, 1\} \qquad \forall\, s, \tau, m, r, j \tag{16}$$

$$u_{s\tau} \geq 0 \qquad \forall\, s, \tau \tag{17}$$

The objective is minimizing the sum of the investment depreciation and the labor cost (1). For investments, linear depreciation is applied with a useful life of $\mathcal{D}$. Each task must be assigned to a station, in one of its possible execution modes (2). Yet, a task can be assigned to a station only if the station is built (3), and it is equipped with the appropriate robot (4), tool (5), and human operator (6) as required by the selected execution mode. The model allows installing new stations (7), robots (8) and tools (9), but removing resources is prohibited.

Tool changeover times are defined in constraints (10)-(11). Inequality (10) states that a changeover is needed between two subsequent tasks executed at the same station if their tool requirements differ. The changeover is also required before the first task executed at the given station if the first and the last tasks at the station require different tools (11). Note that in the actual implementation, slightly different versions of this constraint are required for the first and last tasks in the production plan, where the predecessor and successor tasks do not exist. Line (12) encodes the precedence constraints between the subsequent tasks of a product. Constraint (13) declares that the cycle time on each station, i.e., the sum of changeover times and task durations, must be appropriately low to satisfy all demand.

The total investment and labor costs are calculated by constraints (14) and (15), respectively. Finally, the binary and the non-negative variables are enumerated in constraints (16) and (17).

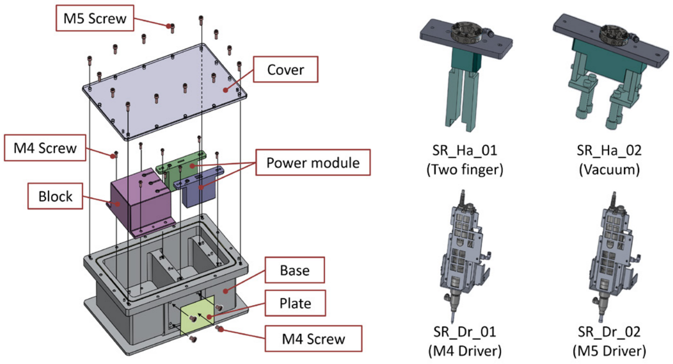
3.4. Integrating task sequencing and system configuration

Given the solvers above for task sequencing and for system configuration, the following procedure is proposed for solving the two interrelated sub-problems simultaneously:

1. **Constructing basic configurations:** The algorithm is initialized by constructing a set of *basic configurations*. Each basic configuration is defined by a given *number of stations* and an *execution mode*, assuming that the product is produced on a line consisting of the given number of stations, using the given execution mode for every task. All possible combinations of execution modes and number of stations, taken from a given interval that can be determined from the relevant range of cycle times, is used to define a basic configuration.
2. **Computing basic sequences:** The optimal task sequence is computed for each basic configuration by solving the corresponding task sequencing problem. The set of solutions to the different task sequencing problems will be referred to as *basic sequences*.
3. **Crossover of sequences:** A *one-point crossover* operator is applied to derive further task sequences by recombining pairs of basic sequences. A break point is generated on the first basic task sequence. The head of the new task sequence equals the beginning of the first task sequence up to the break point. From then on, each task that is not present in the head is inserted into the new task sequence in the order that they appear in the second basic sequence. The crossover operator is applied to each directed pair of two different basic sequences, using each possible break point.
4. **System configurations:** The system configuration problem is solved for each candidate task sequence, including both basic sequences and additional sequences received by crossover. Finally, the algorithm returns the system configuration that yields the lowest cost.

The above algorithm is a *generate-and-test heuristic* in the sense that it tries many different task sequences, but it is not guaranteed to reach the exact joint optimum for task sequencing and system configuration. Its efficiency lies in the idea that the few basic sequences capture the characteristics of task sequences that may become optimal under different conditions. It is noted that the crossover operator is partly motivated by genetic algorithms, but the overall algorithm cannot be called a genetic algorithm.

In experiments on problems of moderate size with up to 35 tasks, the computational effort required–given that both the task sequencing and the system configuration problems could be solved very efficiently–allowed considering all sequences received by crossover. However, on larger problems, where this might become computationally intractable, the required effort might have to be restricted by appropriate search limits. While



(a) Target product (b) Sub-resources for robot

Fig. 1. Product and four robotic tool models in the case study.

this simple approach showed high effectiveness for the available sample problems, the elaboration of more sophisticated algorithms for integrating task sequencing and system configuration should be the focus of further research.

4. Industrial case study

4.1. Experimental scenarios

The effectiveness of the developed approach was verified in a case study involving the assembly of the automotive inverter shown in Fig 1. It is composed of 36 assembled parts. The graph of precedence constraints between the tasks is displayed in Fig 2. Every task can be performed by a human operator or 3 different types of robots, resulting in a total of 4 execution modes. Processing times are defined for the combination of modes and tasks. In addition, tasks require 6 different tools, two robot grippers, two robot screw drivers, and two manual screw drivers; one of them is required by each execution mode of every task. A tool changeover time of 15 seconds occurs when two consecutive tasks within a robotic station are performed using different tools. The depreciation period for every resource is 5 years. Task sequencing and system configuration are executed with two different target cycle time values, 100 seconds and 720 seconds as input.

4.2. Computational results

For this problem instance, the combination of 4 target system sizes and 4 execution modes during task sequencing resulted in 16 basic scenarios, which led to 15 different basic sequences (one was a duplicate). After crossover, the set of all sequences contained 2422 elements[1]. The computation time for generating the basic sequences was 983 seconds, while all other sequences could be generated by crossover quickly, in less than a second.

[1] Theoretically, crossover could generate $N(N-1)(T-1)$ sequences, where N is the number of basic sequences and T is the number of tasks. The two sequences can be selected in $N(N-1)$ different ways, and the $(T-1)$ different break points can be applied. For $N = 15$ and $T = 35$, this results in 7140 sequences, but many were duplicates.

For the cycle time limit of 100 seconds and the original task sequence used currently in the factory, the optimal system configuration could be computed in 6 seconds, and yielded a depreciation cost of 1 078 333 yen for 9 stations. The generation of the basic sequences and the solution of the corresponding system configuration problems took altogether 1 111 seconds, and resulted in a depreciation cost of 1 035 833 yen for 8 stations, which is a 3.94% reduction compared to the original sequence. The total computation time for all the sequences was 30 286 seconds, and the best sequence incurred a depreciation cost of 1 027 500 yen, again for 8 stations, but a better selection of the resources. This is a 4.71% reduction compared to the original sequence, and a 0.77% further reduction compared to the basic sequences. The results are summarized in Table 2.

	Num. seq.	Obj. (yen)	Comp. time (sec)		
			TS	SC	Total
Original sequence	1	1 078 333	0	6	6
Basic sequences	15	1 035 833	983	128	1 111
All seq. (1h limit)	324	1 035 833	983	2 617	3 600
All sequences	2422	1 027 500	983	29 302	30 286

Table 2. Integrated task sequencing and system configuration: results on the cycle time limit of 100 seconds. Computation times are displayed separately for task sequencing (TS) and system configuration (SC).

The results for a cycle time limit of 720 seconds are displayed in Table 3. The task sequences were the same as in the previous case. The original task sequence yielded a depreciation cost of 203 333 yen, and the optimal system configuration could be computed in 138 seconds. The generation of the basic sequences and the solution of the corresponding system configuration problems took 1858 seconds, and resulted in a depreciation cost of 143 333 yen, which is a 29.5% reduction compared to the original sequence. This system configuration consisted of a single robot (the fastest and most expensive robot was required to meet the cycle time limit) and all the necessary grippers and screw drivers in a single station. Obviously, this system configuration cannot be improved further by any sequence.

	Num. seq.	Obj. (yen)	Comp. time (sec)		
			TS	SC	Total
Original sequence	1	203 333	0	138	138
Basic sequences	15	143 333	983	875	1858
All seq. (1h limit)	34	143 333	983	2617	3600
All sequences	2422	143 333	983	*	*

Table 3. Integrated task sequencing and system configuration: results on the cycle time limit of 720 seconds. *The experiments were aborted before computing the system configuration for all sequences, since the configuration for a basic sequence was provably optimal.

The results show that the gain by better task sequencing depends largely on the system and the problem instance. In general, for small-scale systems, the chance for reducing the system size by one station may result in significant relative savings. In the above case study with the cycle time limit of 720 seconds, the number of stations is halved from 2 to 1 by optimizing the task sequence, minimizing the tool changeover times, and also

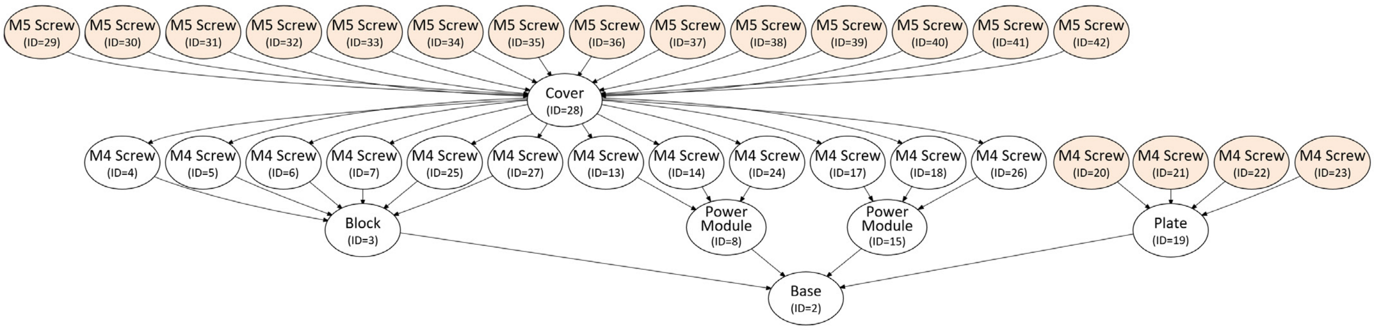


Fig. 2. Precedence graph between the assembly tasks in the case study.

shortening the processing times by using an expensive robot. As a result, total costs were reduced by 29.5%. On the other hand, in the case of a large-scale system with strict restrictions on the target cycle time, the relative savings are naturally lower. In the case study with a cycle time limit of 100 seconds, the number of stations is reduced from 9 to 8, and the total cost reduction rate is 4.7%, which is still a significant gain.

5. Conclusions and future works

In this research, an optimization approach for combined task sequencing and system configuration was proposed for system integration businesses focusing on human-robot assembly lines. From the viewpoint of optimization, the challenge was to reduce the search space in a complex combination of interrelated problems. To solve this problem, a heuristic method was developed that focuses on the difference in the work ability of human and robotic resources, generates a task sequence for each candidate resource, and optimizes the system configuration based on these task sequences. In a case study involving an automotive inverter assembly line, the proposed approach reduced the investment cost by about 4 to 29% depending on the target cycle time. Hence, the main benefit of the approach is that despite its simplicity, it generates close-to-optimal solutions with reasonable computation times for the combination of two optimization problems that are already challenging on their own.

We see three major limitations of the approach. First, in case of many productions tasks (e.g., more than 40) or high order flexibility (very few precedence constraints), it might be impossible to compute the system configuration for each candidate task sequence. Hence, algorithms have to be elaborated to limit search for the most promising sequences. Second, in case of multiple products in a common production system, the approach ignores the potential interrelations between different products. Extension to large-scale multi-product problems may also require replacing the current commercial MILP solver by powerful meta-heuristic solution approaches. Finally, the computed task sequence and system configuration may be sensitive to fluctuations in the production system, such as the variation of the demand. Robust approaches must be developed that can ensure good performance under such fluctuations as well.

Acknowledgements

The authors would like to express their gratitude to all who supported the continuous scientific cooperation between the Hitachi Ltd., Research & Development Group Japan, as CIRP Corporate Member, and the Institute of Computer Science and Control (SZTAKI), started more than fourteen years ago. The Hungarian authors of the paper were partially supported by the Ministry for Innovation and Technology and the National Research, Development and Innovation Office within the framework of the National Lab for Autonomous Systems and the ED_18-2-2018-0006 grant on "Research on prime exploitation of the potential provided by the industrial digitalisation".

References

[1] Alting, L., Zhang, H., 1989. Computer aided process planning: The state-of-the-art survey. International Journal of Production Research 27, 553–585.

[2] Battaïa, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. International Journal of Production Economics 142, 259–277.

[3] Battaïa, O., Dolgui, A., Guschinsky, N., 2020. Optimal cost design of flow lines with reconfigurable machines for batch production. International Journal of Production Research 58, 2937–2952.

[4] Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 168, 694–715.

[5] Enomoto, A., Aoyama, Y., Yamauchi, Y., Yamamoto, N., 2016. Near optimal assembly sequence generation, in: 2016 IEEE/SICE International Symposium on System Integration (SII), pp. 95–101.

[6] Enomoto, A., Yamamoto, N., Yamamura, Y., Sugawara, Y., 2020. Process knowledge integrated assembly sequence planning for control panel. International Journal of Automation Technology 14, 6–17.

[7] International Federation of Robotics (IFR), 2018. World Robotics – Industrial Robots 2018. VDMA Verlag.

[8] Kardos, C., Kovács, A., Váncza, J., 2020. A constraint model for assembly planning. Journal of Manufacturing Systems 54, 196–203.

[9] Scholl, A., Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research 168, 666–693.

[10] Tsutsumi, D., Gyulai, D., Kovács, A., Tipary, B., Ueno, Y., Nonaka, Y., Fujita, K., 2020. Joint optimization of product tolerance design, process plan, and production plan in high-precision multi-product assembly. Journal of Manufacturing Systems 54, 336–347.

[11] Wang, L., Wang, X.V., Váncza, J., Kemény, Z. (Eds.), 2021. Advanced Human-Robot Collaboration in Manufacturing. Springer.