Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 00 (2024) 000–000

www.elsevier.com/locate/procedia

57th CIRP Conference on Manufacturing Systems

Managing Fluctuations in Production via an Optimal Portfolio of Assembly Line Configurations

Daisuke Tsutsumi*[,a], Júlia Bergmann[b,c,f], Péter Dobrovoczki[b,d], Naohiro Hayashi[a], András Kovács[b], Zsombor Szádoczki[b,e], Ádám Szaller[b,f]

[a]*Hitachi Ltd., Research & Development Group, 292 Yoshida-chyo, Totsuka-ku, Yokohama-shi, Kanagawa, 244-0817, Japan*
[b]*HUN-REN Institute for Computer Science and Control (SZTAKI), Budapest, Hungary*
[c]*Eötvös Loránd University, Doctoral School of Informatics, Budapest, Hungary*
[d]*Eötvös Loránd University, Doctoral School of Mathematics, Budapest, Hungary*
[e]*Department of Operations Research and Actuarial Sciences, Corvinus University of Budapest, Hungary*
[f]*EPIC InnoLabs Nonprofit Ltd., Budapest, Hungary*

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

Abstract

The traditional approach to mass production involves maintaining a consistent volume of the same product type for an extended period, minimizing fluctuations in parameters outlined in the process plan to ensure planned production efficiency. However, in mixed human-machine production lines, delays in human work and machine malfunctions are common, leading to significant fluctuations in assumed parameters and decreased throughput during the production control period. This research aims to develop technology for swiftly establishing production lines, accounting for fluctuations such as variations in task duration in mixed human-machine production lines (proactive planning approach). To manage configurations robustly, the approach involves pre-generating alternative configurations based on fluctuation scenarios and selecting and implementing suitable configurations to maximize throughput within budget constraints. Additionally, a novel technology is proposed to enable production continuity through flexible re-planning in response to operational fluctuations (reactive planning approach). The goal is to reduce the production preparation period and maintain target throughput during operation. Validation of the introduced method is demonstrated through computational analysis using an industrial case study simulating a battery assembly line for electric vehicles. The experiments results show throughput improvement even with the implementation of a small number of alternative configurations. This research contributes to enhancing production line adaptability and efficiency in the face of fluctuating operational conditions, ultimately improving overall manufacturing performance and competitiveness.

Keywords: production line; system configuration; dynamic re-planning

1. Introduction

The production system configuration problem is a crucial sub-problem of production line design [7]. It entails defining a production system capable of producing given products in desired quantity and quality in the most efficient manner possible. Decisions to be made during system configuration include investment in new resources such as production lines, stations, robots, tools, storage, etc., selecting the appropriate execution mode for each production task, and assigning these tasks to stations.

In the conventional mass production method, a certain amount of the same variety is produced for a long period of time [6]. Therefore, the fluctuation of the parameters—target value such as required quantity, input value such as process capability—assumed in the production system configuration is small enough not to affect productivity, and the production efficiency as planned can be achieved. However, in the operation of a mixed line of humans and machines, work delays by humans and malfunctions of machines frequently occur. Consequently, there is a problem where the fluctuation of the assumed parameters in the process plan becomes significant during the production control period, leading to a decrease in throughput.

For instance, in the production line of EV battery modules operated on dedicated lines for each product, maximizing

2212-8271

throughput poses a significant challenge. Due to the backdrop of high labor costs, efforts are underway to automate production lines as much as possible using robots and dedicated machines. However, certain processes such as parts kitting, wire harnessing, and final assembly remain difficult to automate, necessitating a hybrid approach involving both human and robotic labor. Therefore, during line operation, the task duration for workers varies due to factors such as absenteeism due to health issues, the influx of inexperienced workers, and the allocation of tasks beyond individuals' specialties. Additionally, the task duration for robots fluctuates due to retries resulting from positional errors and robot failures. These variations in task durations lead to uneven line balancing and a subsequent decrease in throughput. To maintain productivity despite fluctuations, the production system needs to detect these variations and actively respond to them. This paper focuses on finding ways to appropriately address task duration variations at the level of system configuration.

In the case of flow systems, modelling production system configuration—i.e., determining the optimal combination of equipment and resources, in addition to their assignment to production tasks for given process plans and demand volumes—corresponds to different versions of the *assembly line balancing problem* (see, for instance, [1, 2]). Adaptability and responsiveness are required more and more for future manufacturing systems [5]. A recent review of flexibility in manufacturing system design can be found in [14]. [11] proposes an integrated framework that generates multiple feasible configurations, and evaluates them by simulations and multi-objective optimization. [12] focuses on alternative plans managing resource failures. Facing uncertainty and changes in the duration of tasks studied, e.g., by [3, 4, 8, 9, 10, 15].

However, determining the optimal portfolio of alternatives that should be implemented on the shop floor in the case of stochastic task completion times has not been investigated in detail yet.

2. Problem statement

The focus of the paper is managing fluctuations—i.e., task duration variations—in the configuration problem of a serial production line, which consists of a number of either robotic or human-operated stations, equipped with different tools.

The duration of a given task can vary significantly, e.g., based on the difference in the experience of the operators or due to a potential retry of robot operation. Therefore, the bottleneck process can be different depending on the combination of task duration variations resulting in a lower throughput than expected at the time of planning.

Addressing the task duration fluctuations by robust configuration assumes the availability of necessary equipment, human operator instruction and training, as well as control programs, thus, the problem must be tackled in different stages. First, *alternative configurations* must be pre-generated according to *relevant fluctuation scenarios*. Then a reasonable *portfolio of the most appropriate configurations* must be identified and implemented on the shop floor.

The following types of fluctuations are assumed to be present in the examined task durations:

- Trend variations (sustained change, improvement or deterioration):
  - High frequency (occurs within a shift), e.g., worker fatigue;
  - Medium frequency (occurs after a few shifts), for instance, worker learning;
  - Low frequency (its effect occurs only in the long run), e.g., deterioration of machines and jigs.
- Cyclical (always reoccurring) variations:
  - High frequency (appears in each shift), for example, the difference in worker's abilities in each shift;
  - Low frequency (appears on a monthly basis), e.g., replacement of workers, maintenance of machines.

Based on that, it is important to investigate the different fluctuation scenarios of the system, i.e., the collections of appropriate resource-task duration pairs.

In the case of the production system configuration for the nominal process, the tasks have to be assigned to the individual stations according to their pre-defined process plan in such a way that the forecast demand of the product is satisfied and the total production cost—involving the investment depreciation and the labor cost—is minimized. A station executes a number of consecutive tasks, each of which must be assigned to exactly one station. Alternative execution modes are available for each task, which require different combinations of resources and have different nominal duration times. A task can only be executed by a station in a given mode if the station is equipped with all the required resources. A single human operator or a single robot can be assigned to each station, whereas in the industrial case study there are also stations with dedicated machines, which can only perform a handful of special tasks.

In the case of the *generation of alternative configurations*, the objective currently is to list all possible solutions. However, here the system is fixed according to one of the two following assumptions. (1) the line design cannot be changed compared to the initial configuration, but the assignment of the tasks can be modified (if possible). (2) main resources (robots and human operators) are fixed, while auxiliary resources (tools) can be changed arbitrarily together with the assignment of tasks.

As for the determination of the *optimal portfolio of alternative configurations*, the objective is selecting from the set of all generated alternative configurations a smaller portfolio for implementation on the shop floor that fits into the available implementation budget and ensures the best performance over all investigated scenarios. This is captured by maximizing the throughput over all scenarios when using the best selected and implemented configuration for the corresponding scenario and respecting the B^Y configuration budget, the B^R robot programming task budget, and the B^H human training task budget.

The notation of the latter problem is summarized in Table 1.

3. Solution approach

3.1. Overview

The proposed approach to managing fluctuations consists of the following steps, organized into a linear workflow:

1. Depart from the optimal production system configuration for the nominal process. This can be computed, e.g., by the MILP approach presented in [13];
2. Generate samples of relevant fluctuation scenarios;
3. Compute a large set of candidate alternative system configurations, potentially containing all alternative configurations that can be activated with a low changeover time upon facing fluctuations;
4. Select an optimal portfolio of alternative configurations that maximizes expected throughput over the fluctuation scenarios subject to constraints on the available engineering effort.

Steps 2, 3 and 4 are presented in detail in the following sections.

3.2. Generating fluctuation scenarios

In this section data analytics approaches are discussed for characterizing fluctuations from historical data and how to use them for scenario generation. The biggest challenge is to correctly identify worker types with different experience level based only on their processing times. If these types are available, then seasonal decomposition can find the right model to represent the behavior of human learning and fatigue, and similar methods are applied to describe machine deterioration as well. The output of the following two submodules is then given to the scenario generator which creates a pool of future scenarios which can be used later.

The machines are assumed to have a fixed configuration, which means that the machine settings and the system layout will remain the same throughout the data analysis. However, the machines are expected to deteriorate over time and require maintenance to keep them in optimal condition. It is also assumed that the human workers in the system will continuously learn and improve over time, which is represented by a long-term trend. However, these workers will also experience fatigue, which follows a short-term seasonality pattern. The system operates in a single shift, which means that exactly one worker works during each shift at each station. The processing times of the system are subject to random noise, which can impact the overall system performance. Finally, the system assumes that the worker pool can change, which means that the workers available to work in the system can vary over time. This can impact the performance of the system and requires careful monitoring and analysis to ensure optimal performance.

3.2.1. Operator assignment via clustering

Each task has a technical minimum processing (denoted by y_j^{min} for task j) time meaning that in the long run if all workers are allowed to repeat the task infinitely many times then all will reach (or get arbitrarily close to) these minimum processing times, and each worker has an initial skill level of all tasks (y_{kj}^{start} for worker k on task j). Therefore if all the processing times are plotted in a high-dimensional real space then all the scatters coming from the same worker will be placed on (or be very close to) the same high-dimensional line (pointing from $\left(y_{k1}^{start}, y_{k2}^{start}, \dots y_{km}^{start}\right)$ to $\left(y_1^{min}, y_2^{min}, \dots, y_m^{min}\right)$, where m denotes the number of tasks). By fitting the closest high-dimensional line for each shift, shifts with the same worker will have lines with very similar steepness. In practice this is equivalent to finding the last principal component of all shifts, also called the direction vector of the sought high-dimensional line. Let us denote the direction vector of shift i by $v_i = (v_{i1}, v_{i2}, \dots, v_{im}) \in \mathbb{R}^m$.

The next task is to find automatically which direction vectors (v_i-s) belong to the same group, therefore to cluster the direction vectors. If the number of clusters is unknown, the elbow-method can be used to find the optimal value for it, label of shift i denoted by C_i^1. Based on experiments, most of the related shifts are put to the same cluster in most of the cases, however, it is a strong suspicion that some shifts are mislabeled. The following sections will give a solution for the refinement.

The proposed solution is inspired again by the shape of data point cloud. By nature, those shifts who belong to the same cluster are positioned around the same high-dimensional line, therefore there is a line corresponding to each cluster. Therefore a line must be fitted on all data points from each cluster: the closest line $\mathcal{L}_i$ is fitted by ordinal least squares on the processing time data points of cluster C_i^1, the label of the jth data point $\left(y_{j1}, y_{j2}, \dots, y_{jm}\right)$ is the closest line's label C_j^2.

However nothing guarantees that data points from the same shift fall into the same cluster label so far. A solution for mix-labeled shifts is performing a simple majority voting. For all shifts assign the label which occurs most frequently in the shift. Formally, the final label of shift i is $C_i := \text{argmax}_k \left|\left\{j \middle| \left(y_{j1}, y_{j2}, \dots, y_{jm}\right) \in S_i \wedge C_j^2 = k\right\}\right|$, where S_i denotes shift i.

The proposed clustering method is of course not infallible, but its efficiency in the current research field is unquestionable. The solution highly depends on the background engineering knowledge of the learning behavior of human workers. Our clustering method is highly effective and useful in grouping similar data points together based on a selected similarity metric. Through our method, identifying patterns in data and gain insights into the underlying structure of the data is possible, which can be used to make predictions and improve decision-making.

3.2.2. Time series analysis

Time series analysis (TSA) is particularly important in situations where changes over time can impact decision-making, such as in production processes. The processing time logs time series data, which consists of a negative exponential trend component (due to long-term learning), a monotone increasing seasonal component (due to short-term fatigue), and some small random noise, provides an excellent example of the value of time series analysis. There are two very different types of processing modes: human and machine modes. Intuitively, human operations' processing times will get shorter over repetition, but it is also expected that workers get tired by the end of their shift, therefore some processing time growth periodically is expected. On the other hand, machines are famous for continuous workload, however maintenance is required from time to time.

In case of human operations, it is assumed to have either the result of the clustering from previous section or the real schedule timetable of the operators. Either way, the below described steps are to be applied to each of the worker group separately.

The timestamps are typically not identically distributed, therefore resampling is necessary. Resampling involves either increasing or decreasing the number of data points in the time series by interpolating the values of the series at new time points. A mixture of upsampling and downsampling is used to get a regularly spaced time data. The seasonal decomposition of processing times shows the operator fatigue as the seasonal component, and the operator learning as the trend component. Assuming that the background learning curve is a monotone decreasing function which drops rapidly in the beginning and then slowly approaches to the technical minimum time of the given task, it can be modelled with a negative exponential function ($f(x) = a \cdot e^{-bx} + c$) by non-linear least squares regression. The parameters a, b and c of this background trend function are later used in scenario generation.

In case of machine operations, the focus switches from trend to seasonality. Machinery processing times have a stable trend component, as there are no long-term learning present in case of a machine, however, due to deterioration, maintenance events happen on regular basis, which results in increasing processing times. Since the target is to forecast future processing times, the deterioration must be modelled. Deterioration is a monotone increasing component and has the highest effect before maintenance events, therefore is estimated by fitting a function in the shape of $f(x) = a \cdot x^b + c$ on the seasonal component. The fitting method is the same as in case of human stations, and a, b and c parameters are again used later in scenario generation.

3.2.3. Scenario generation

Scenario generation consists of three steps: history mining, forecasting and translation. However, before any scenario generation process, scenario as a concept must be introduced. A scenario is a series of estimated or expected processing times for all tasks and for all process mode (human, machine, robot) pairs. By using the results of the previous sections (clustering and TSA), expected scenarios of the future are possible to forecast for a given period of time. First, the latest shift's mean processing time is calculated and it is considered to be the current state of the system. Then by using the results of TSA, the future processing times are estimated by shifting current states via learning and deterioration curve's parameters.

3.3. Generating alternative configurations

The proposed approach focuses on two classes of alternative configurations that can be activated with low changeover times when the system faces fluctuations: (1) alternatives that share *all resources* with the nominal configuration, but apply a *different task assignment*; and (2) alternatives that share the *stations, robots and machines* with the nominal configuration, but may involve *different tools* and *different task assignments*. Since the investigated industrial case study is characterized by very limited flexibility in the system due to a large number of dedicated machines, the number of such alternatives is relatively small, and the proposed approach focuses on generating *all* alternative configurations.

Dimensions, indices	
S	Number of scenarios
C	Number of configurations
i	Scenario index
j	Configuration index
s	Station index
t	Task index
Input parameters	
Θ_{ij}	Throughput if configuration j is applied to scenario i [pcs/shift]
r^0_{ts}	Robotic task t is already implemented at station s (binary)
h^0_{ts}	Human task t is already implemented at station s (binary)
y^0_j	Configuration j is already implemented on the shop floor (binary)
R	Set of tuples (j, t, s) such that in configuration j, task t is performed by a robot at station s
H	Set of tuples (j, t, s) such that in configuration j, task t is performed by a human at station s
B^Y	Available budget for implementing new configurations
B^R	Available budget for implementing new robotic tasks
B^H	Available budget for implementing new human tasks
Decision variables and objective	
r_{ts}	Robotic task t is selected for implementation at station s (binary)
h_{ts}	Human task t is selected for implementation at station s (binary)
y_j	Configuration j is selected for implementation on shop floor (binary)
z_{ij}	Configuration j is assigned to scenario i (binary)

Table 1. Notations of the portfolio optimization problem.

The problem of searching for feasible alternative configurations is formulated as a constraint satisfaction problem, with variables corresponding to the index of the station assigned to each of the tasks. Constraints capture that all resources required by the task must be available in the station, according to the applicable class of alternatives (see above); and the precedence relations between tasks, which state that if task t_1 is located earlier in the process plan than task t_2, then t_1 must be assigned to a station not later than task t_2 in the serial assembly line. Then, feasible alternative configurations are explicitly enumerated by a depth-first search procedure.

In potential applications where generating *all* alternative configurations is computationally intractable, a straightforward heuristic approach is generating the optimal alternative configuration for each of the fluctuation scenarios.

3.4. Computing optimal portfolio of configurations

While a large number of system configurations can be generated effectively, the prerequisites and the process of the implementation of those configurations on the shop floor potentially limits the number of configurations to be considered. Hence, in applications where the modification of the configuration or the task assignment requires substantial effort (e.g., robot and PLC programming, cross-training human operators, ensuring proper parts supply), a smaller subset (portfolio) of alternative configurations must be selected carefully.

The objective is selecting a portfolio of alternative configurations from the set of all generated alternative configurations that fits into the available implementation budget and ensures the best performance over all investigated scenarios, i.e., has the maximal average throughput. The problem is formulated as a MILP as in (1). Problem parameters, indices and variables are summarized in Table 1.

$$
\begin{aligned}
&\text{maximize} && \frac{1}{S}\sum_{i,j}\Theta_{ij}z_{ij} && && \text{(1a)}\\
&\text{subject to} && \sum_{j} z_{ij} = && 1 \quad \forall i && \text{(1b)}\\
& && z_{ij} \leq && y_j \quad \forall i,j && \text{(1c)}\\
& && y_j \leq && r_{ts} \quad \forall (j,t,s)\in R && \text{(1d)}\\
& && y_j \leq && h_{ts} \quad \forall (j,t,s)\in H && \text{(1e)}\\
& && y_j \geq && y_j^0 \quad \forall j && \text{(1f)}\\
& && r_{ts} \geq && r_{ts}^0 \quad \forall t,s && \text{(1g)}\\
& && h_{ts} \geq && h_{ts}^0 \quad \forall t,s && \text{(1h)}\\
& && \sum_{j:\, y_j^0=0} y_j \leq && B^Y && \text{(1i)}\\
& && \sum_{t,s:\, r_{ts}^0=0} r_{ts} \leq && B^R && \text{(1j)}\\
& && \sum_{t,s:\, h_{ts}^0=0} h_{ts} \leq && B^H && \text{(1k)}\\
& && r_{ts}, h_{ts}, y_j, z_{ij} \in \{0,1\} && \forall t,s,i,j && \text{(1l)}
\end{aligned}
$$

The objective is to minimize the average throughput over all scenarios (1a). Constraint (1b) states that exactly one configuration is assigned to each scenario, while constraint (1c) ensures that a configuration that is assigned to a scenario is also selected into the portfolio. Constraints (1d,1e) state that if a configuration is selected for the portfolio, then its robotic and human tasks must be implemented (respectively). If a configuration, robotic task, or a human task is already implemented, then they are available later on (1f,1g,1h). The implementation budget of new configurations (1i), new robotic tasks (1j), and new human tasks (1k) must be respected.

4. Experimental evaluation

4.1. Industrial case study

Computational experiments were conducted to test the effectiveness of the proposed method. The test problem, reflecting the design problem of an EV battery assembly line, aimed to design a line for one product with 51 tasks over a period of one year. The main resources of the line's stations included three types of robots, four types of dedicated machines, and human operators. Tasks on the stations could be executed in any of the valid execution modes. The stations could be equipped with 13 types of tools, with a maximum limit of three tools per station. The carry-in and carry-out times of workpieces on the stations were uniformly fixed at 2 seconds. The objective was to minimize the depreciation cost of investment, labor cost, and maintenance cost, while the cycle time needed to meet different constraints given as input. If deterministic task execution times were used, they were approximated by the sample mean.

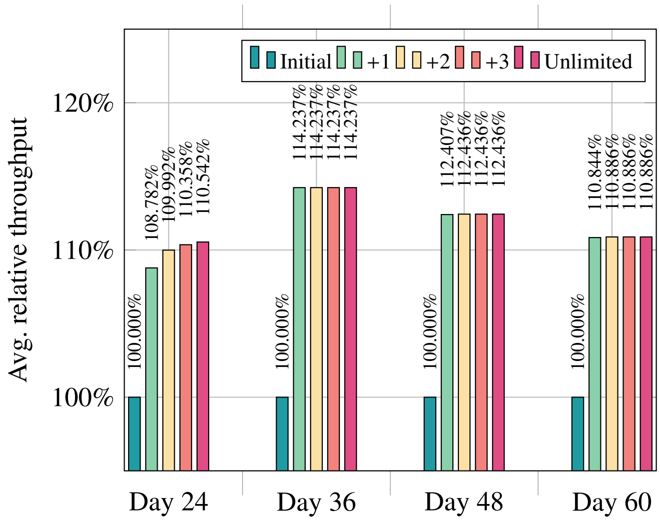

Fig. 1. Experimental results: throughput with different portfolios of alternatives as percentage of throughput with initial configuration

4.2. Results

In this experiment, the impact of switching between alternative system configurations upon fluctuations was evaluated. All alternative configurations for a given initial configuration were computed and the fluctuation scenarios were generated by data analytics. The solution of the optimal portfolio problem was evaluated by simulation.

Experiments were executed on a rolling horizon with four planning steps, each corresponding to 24, 36, 48 and 60 working days after line installation. In each planning step, the decisions made in the previous steps on the alternatives to be implemented were taken as constraints.

The initial configuration—which consisted of 16 stations—was the optimal line for the EV battery use case with a cycle time limit of 60 seconds, which corresponds to producing 480 pieces per 8-hour shift. It was assumed that initially no alternative configurations were available at the shop floor.

The set of alternative configurations consisted of 648 alternatives, with the assumption that the stations and their main resources are fixed, whereas auxiliary resources can be added to the stations whenever required. The generation of the compact representation of the alternatives took 0.025 seconds by constraint programming, whereas enumerating and writing them took 5.508 seconds. The engineering budget was defined as number of new configurations implemented per planning step, while there was no separate budget imposed on the number of robot and human tasks to be implemented. The five different values of the budget evaluated and compared included 0,1,2,3 and infinite alternative configurations per planning step. In the case of infinite budget it is allowed to implement arbitrary number of alternatives at the beginning of the planning horizon, and serves as an upper estimation on the gain that can be achieved by using alternative configurations to manage fluctuations.

In the baseline case, where the initial configuration is used throughout the horizon, only 272-330 pieces can be produced per shift, which is 31.1%-43.2% less than planned. When alternatives can be exploited without limitations, the average throughput increases to 306-365 pieces. Figure 1 shows the comparison of throughput for optimal portfolios of alternative configurations subject to different engineering budgets. Notable that this improvement can be achieved with a rather small engineering budget as well. Implementing only one alternative configuration resulted in an improvement of 8.8% in the throughput, while maximal improvement in the first planning step was realized by implementing only 5 out of the 648 alternatives, resulting in an improvement of 10.5%.

It is also worth mentioning that in each of the planning steps, it only took 1.48 seconds on average to solve the configuration portfolio optimization problem (excluding the time of loading the input data).

5. Conclusions and future works

This paper addressed the management of fluctuations in production systems via dynamic reconfiguration. Emphasis was placed on generating alternative configurations a priori that can be activated when certain fluctuations are detected, in order to maintain system throughput while also keeping the number of alternative configurations, and hence, the required engineering effort in bay. Data-driven analysis techniques were proposed to predict future behavior from past data, characterizing potential future variability. Then, a novel method was introduced to compute an optimal portfolio of alternative configurations that fit within engineering budgets and maximize average throughput for the future scenarios. In computational experiments on an EV battery assembly use case, throughput improved by 10-14% with the implementation of only 5 alternative configurations for a highly complex system.

The results of the current project indicate several avenues for future research. Firstly, there is a need for data analytics on data from other (similar) production lines when historical data from the actual line is unavailable. Secondly, data analytics for resource breakdowns could help identify resources more prone to failure and optimize system configurations accordingly, but access to similar production line data is necessary due to the low occurrence rate of failures. Lastly, combining data-driven techniques with solid engineering knowledge could enhance predictions about future production system behavior by addressing discrepancies between data-driven predictions and engineering intuition.

Acknowledgements

Hungarian authors were supported by the TKP2021-NKTA-01 NRDIO grant and by the National Laboratory for Autonomous Systems (RRF-2.3.1-21-2022-00002). This research was supported by the Doctoral Student Scholarship Program of the Co-operative Doctoral Program of the Ministry of Innovation and Technology, financed by NRDIO.

References

[1] Becker, C., Scholl, A., 2006. A survey on problems and methods in generalized assembly line balancing. European Journal of Operational Research 168, 694–715.
[2] Boysen, N., Schulze, P., Scholl, A., 2022. Assembly line balancing: What happened in the last fifteen years? European Journal of Operational Research 301, 797–814.
[3] Derkinderen, V., Bekker, J., Smet, P., 2023. Optimizing workforce allocation under uncertain activity duration. Computers & Industrial Engineering 179, 109228.
[4] Diefenbach, J., Stolletz, R., 2022. Stochastic assembly line balancing: General bounds and reliability-based branch-and-bound algorithm. European Journal of Operational Research 302, 589–605.
[5] ElMaraghy, H., Monostori, L., Schuh, G., ElMaraghy, W., 2021. Evolution and future of manufacturing systems. CIRP Annals 70, 635–658.
[6] Hu, S.J., 2013. Evolving paradigms of manufacturing: From mass production to mass customization and personalization. Procedia CIRP 7, 3–8.
[7] Koren, Y., Hu, S.J., Weber, T.W., 1998. Impact of manufacturing system configuration on performance. CIRP Annals 47, 369–372.
[8] Lolli, F., Balugani, E., Gamberini, R., Rimini, B., 2017. Stochastic assembly line balancing with learning effects. IFAC-PapersOnLine 50, 5706–5711.
[9] Manzini, M., Demeulemeester, E., Urgo, M., 2022. A predictive–reactive approach for the sequencing of assembly operations in an automated assembly line. Robotics and Computer-Integrated Manufacturing 73, 102201.
[10] Pereira, J., Álvarez-Miranda, E., 2018. An exact approach for the robust assembly line balancing problem. Omega 78, 85–98.
[11] Poudel, L., Kovalenko, I., Geng, R., Takaharu, M., Nonaka, Y., Takahiro, N., Shota, U., Tilbury, D.M., Barton, K., 2022. An integrated framework for dynamic manufacturing planning to obtain new line configurations, in: 2022 IEEE 18th International Conference on Automation Science and Engineering (CASE), pp. 328–334.
[12] Tsutsumi, D., Bergmann, J., Dobrovoczki, P., Hayashi, N., Kovács, A., Szádoczki, Z., Szaller, Á., Umeda, S., 2023. Dynamic production line rebalancing by alternative plans for compensating equipment failures. Procedia CIRP 120, 810–815.
[13] Tsutsumi, D., Kovács, A., Szalóki, Á., 2022. Novel heuristic approach to integrating task sequencing and production system configuration. Procedia CIRP 107, 28–33.
[14] Weckenborg, C., Schumacher, P., Thies, C., Spengler, T.S., 2024. Flexibility in manufacturing system design: A review of recent approaches from operations research. European Journal of Operational Research 315, 413–441.
[15] Yılmaz Delice, E.K.A., Özcan, U., 2016. Stochastic two-sided u-type assembly line balancing: a genetic algorithm approach. International Journal of Production Research 54, 3429–3451.