Evaluation of redundant configurations in assembly lines with fractional tasks

Paolo Renna

Received: 23 March 2022 / Accepted: 13 May 2022 / Published online: 7 June 2022
© The Author(s) 2022

Abstract
Assembly lines with fractional tasks increase the throughput with a better workload balancing among the stations. The possibility to share tasks between two consecutive stations can also support the design of assembly lines with redundancies of the tasks. The redundancy of the tasks allows reacting to short failures or variations in processing time. This paper proposes a framework to support the design and control of redundancy assembly lines. The first problem concerns a modified mixed-integer linear programming model used for the fractional allocation assembly line problem to design redundancy assembly lines. Then, this research proposes control policies to manage the allocation of the tasks shared between two consecutive stations. Finally, the simulation evaluates the performance of the proposed methods to handle short failures and uncertain processing times. The numerical results show the potential improvements of the proposed methods with a limited number of redundancies. The main improvements concern the reduction of throughput time and work in process.

Keywords Assembly line · Fractional task allocation · Line balancing · Redundancy · Simulation

1 Introduction and motivation

Many manufacturing contexts use assembly lines such as automotive, electronics, and many others. The design of assembly lines concerns the line balancing assigning the tasks to the stations [1]. The assignment of the tasks to the stations is binary, but some works in the literature change this hypothesis considering dynamic balancing or work sharing [2, 3]. These works studied how some tasks can be shared between two stations in particular conditions as: e.g. each station pair can only share one task, and each task can only be shared by two stations. The concept of fractional task allocations introduces the possibility to reduce the cycle time allowing higher throughput.

Moreover, the introduction of automation such as industrial robots is relevant in the last years to improve the performance of assembly lines [4].

Then, it is more important the return on investment of these production lines, therefore, the throughput should be maximized and needs to be robust to random failures. The short failures [5] cause throughput loss reducing the profitability of the assembly lines with starvation or blocking of the stations. To reduce the effect of short failures, in the literature are proposed some works on the redundancy configurations [6, 7], while other works evaluate the introduction of buffers with preventive maintenance, but these approaches increase the costs of the items in queues [8, 9].

The redundancy configurations allow some tasks to be moved to downstream stations when failures occur. This needs that some downstream stations have the equipment necessary to perform other tasks previsioned in the upstream stations.

The digital transformation referred to as “Industry 4.0” can support the rapid reconfiguration of the stations to react when failures occur [10]. The main characteristics of industry 4.0 are the connectivity to collect data in real-time from the stations using sensors or by the people involved in the process [11]. This allows measuring the performance and introduces the improvements in real-time. The collection of the information from the line provides knowledge on the effects of improvements and make easier the use of the intelligent algorithm or artificial intelligence to improve the introduction of self-adapting smart systems.

This research integrates the two above concepts of the fractional assembly line and redundancy configurations to
improve the performance of assembly lines face to unforeseen events such as failures or process time fluctuations. A mathematical model to design a fractional assembly line with redundancy is proposed. Control policies to support the allocation of the tasks shared between stations were proposed. Then, simulation models are used to test the results of the design model and evaluate the performance compared to a model proposed in the literature.

The article is structured as follows: Sect. 2 discusses the main works proposed in the literature about the Fractional Assembly line problem and redundancy. Section 3 presents the reference context and the notation. Section 4 describes the method proposed in terms of mixed-integer linear model and control policies for the shared tasks. The design of the experiments is presented in Sect. 5, while Sect. 6 discusses the numerical results with the main findings. Finally, the conclusions and future developments are presented in Sect. 7.

2 Literature review

This section discusses the more recent works proposed in the literature about the fractional assembly line problem and redundancy models in assembly lines.

2.1 Fractional assembly lines

A complete survey on the dynamic line balancing was proposed by [12]. The classical assembly line balancing problem did not concern the possibility to share a task among two or more stations. Chen and Askin [2] define the dynamic line balancing the assignment of some tasks fixed to stations while others are allowed to be shared between two adjacent stations. The numerical results demonstrated that this approach improves productivity.

Anuar and Bukchin [3] studied the work-sharing between two consecutive stations of assembly lines. They proposed a model to identify the tasks to share and the allocation rule. The assignment of the tasks can be changed dynamically but in a long-term state. The simulations of case studies for dynamic line balancing under heterogeneous workers were presented by [13] and [14].

The dynamic line balancing where workers each worker can help only the immediately downstream station was studied by Jeong and Jeon [15]. This work focused on the skill of the workers and considered the unlimited buffers between the stations.

Some works as Lopes et al. [16–18] introduced additional flexibility in mixed-model lines by identical parallel stations. This approach leads to layout complications, and can increase costs more than a fractional task allocation.

Lopes et al. [19] presented a mixed-integer linear programming model to support the fractional allocation assembly line problem. The numerical results suggested the possibility to improve resource utilisation and throughput with relatively low costs. They argued that fractional allocations can also lead to more robust balancing regarding demand uncertainty.

2.2 Redundancy configurations

The redundancy in production lines becomes more relevant in the last years with the digital transformation of the manufacturing systems. Kahan et al. [20] studied a real industrial case of an automotive body shop considering equipment failures and redundancy to minimize throughput loss. They proposed a mixed-integer formulation and tested the model with the discrete event simulation.

Müller et al. [6, 21] proposed design models for automatic flow lines to include the redundancy of the operations. Subsequently, Muller et al. [7] improved the proposed models by including the possibility of maximizing the level of redundancy or distributing the redundancy number along the line. The numerical results compared redundancy configurations, but did not compare the results with a line without redundancy.

Gu et al. [22] studied different configurations of production lines as redundancy, flexibility, and parallel stations to handle the propagation of unexpected disruptive events.

Another solution to improve the robustness of production lines is the introduction of reconfigurable machines with higher scalability and responsiveness but require higher investment costs [23].

Renna [24] proposed an integration between a mathematical model and simulation to design and evaluate the performance of production lines with redundancy. He proposed, also, a model with a backup station that improves the performance but increases the costs with the introduction of an additional station.

Weckenborg et al. [25] studied the planning models to configure manual assembly lines with collaborative robots. The approach proposed assigns the robots to the stations to improve the distribution of the workload and the performance.

The discussion of the literature review highlights how the fractional task allocations problem and the potential of the tasks’ redundancy are not investigated together. The research proposed overcomes the limits of the literature by proposing a mathematical model to solve the fractional task assembly problem by introducing the increment of the number of tasks shared between two stations keeping the cycle time close to the optimal cycle time. Then, a simulation model is proposed to test a control policy to change in real-time the fraction of tasks allocated to each station to react the unforeseen events
such as failures and processing time variations. Then, the first research question of this paper is the following:

**RQ1:** Can the mathematical model proposed support the design of the fractional assembly line with optimized cycle time by increasing the number of fractionated tasks?

The model proposed in the literature assign a static fraction of the tasks to the stations, then the second research question is the following:

**RQ2:** Can the dynamic allocation of the fraction of the tasks improve the performance of the assembly line when uncertain as short failures and processing time variations occur?

### 3 Research context

The research context concerns the fractional allocation assembly line balancing problem in which a set of tasks $T$ with a given processing time $PT_t$ are assigned to a set of stations $S$ under the precedence constraints. The first objective considered is to minimize the line’s cycle time.

The basic model is proposed by Lopes et al. [19] and is briefly summarized below.

The main assumptions of the model proposed by Lopes et al. [19] are the following: the number of stations is given; a task can only be split/shared by two adjacent stations; precedence relations must be respected for all products that flow through the line; only one task can be shared between two stations; each station can share tasks with either their upstream or downstream neighbor, but not both at once; both stations that share the task can perform it, and the task’s duration is constant, regardless of the station that performs it.

The notation used in the following:

| Notation | Definition |
|----------|------------|
| $T$      | It is the total number of tasks to assemble the product |
| $t$      | It is the index of the tasks $t=1,...,T$ |
| $S$      | It is the number of the stations that composes the assembly line |
| $s$      | It is the index of the station $s=1,...,S$ |
| $CT$     | It is the cycle time of the assembly line |
| $PT_t$   | It is the processing time of the task $t$ |

| Notation | Definition |
|----------|------------|
| $P_{t,p}$ | It is a binary value that is equal to 1, if the task $t$ must precede the task $p$ and 0 otherwise |
| $K$      | It is a parameter that indicates the possible subdivision values of a task between two adjacent stations |
| $MaxCycleTime$ | It is the upper level for the cycle time for the redundancy model |
| $Nr$     | It is maximum number of tasks shared for the redundancy model |
| $KF$     | It is a parameter that determines the lower level of the fractional task as 1-KF that can be assigned to a station |
| $X_{t,s}$ | It is a binary value that is equal to 1, if the task $t$ is assigned to station $s$ and 0 otherwise |
| $Y_{t,s}$ | It is a binary value that is equal to 1, if the task $t$ is shared between station $s$ and $s+1$, and 0 otherwise. This is because the task can only be shared between two adjacent stations |
| $Z_{t,s}$ | It is the percentage of the task $t$ performed by the station $s$; this value is equal to 0, if the task is not shared between two stations ($Y_{t,s}=0$) |

Model base [19]

Minimize $CT$  

Subject to:

\[
\sum_{s=1}^{S} (x_{t,s} + y_{t,s}) = 1, \forall t \in T
\]  

\[
\sum_{s=1}^{S} z_{t,s} = 1, \forall t \in T
\]  

\[
\sum_{s=1}^{S} s \ast (x_{t,s} + y_{t,s}) \leq \sum_{s=1}^{S} s \ast (x_{p,s} + y_{p,s}) \forall P_{t,p} = 1
\]  

\[
\sum_{s=1}^{S} s \ast z_{t,s} \leq \sum_{s=1}^{S} s \ast z_{p,s} \forall P_{t,p} = 1
\]  

\[
\sum_{t=1}^{T} PT_t \ast z_{t,s} \leq CT, \forall s \in S
\]
Then, it is added two new constraints (16 and 17) that limits the lower value of fraction of the tasks considering the redundancy.

\[ z_{t,s} > y_{t,s} - KF, \forall t \in T, \forall s \in S \text{ with } s < S \]  

\[ z_{t,s+1} > y_{t,s} - KF, \forall t \in T, \forall s \in S \text{ with } s < S \]  

Finally, expression (18) constrains the cycle time to an upper bound limit.

\[ \text{CycleTime} \leq \text{MaxCycleTime} \]  

The proposed redundancy model increases the number of shared tasks and allows for more than one shared task for each station. The redundancy model is used in conjunction with the basic model to avoid a significant increase in cycle time. Figure 1 describes the use of the two models for assembly line design. The first step is to use the base model to find a solution to minimize the cycle time of the assembly line. The optimized cycle time is used as a constraint for the redundancy model. The redundancy model is performed by increasing the cycle time from the optimized value to provide different solutions that increase the number of redundancies. The different configurations obtained will be evaluated with simulation models.

4.1 Production control of the assembly line

The solutions provided by the redundancy model need a control policy to be effective.

The proposed control policy is based on the buffer evaluation upstream and downstream of a generic assembly line station (see Fig. 2).

When a part leaves the upstream buffer to enter the station, the level of the downstream buffer is evaluated. Two control policies have been proposed: the first based on one threshold level of downstream buffer, while the second evaluates three thresholds (see Fig. 3a, b).

The first policy evaluates the downstream buffer level and if this level is under the level Th, then the entire shared task is allocated to the downstream station. This policy can lead to more fluctuations in the workload allocated to the stations. Therefore, the second policy evaluates the downstream buffer level considering three thresholds: Th, Th1, and Th2. If the downstream buffer level is under Th, then the entire shared task is allocated to the downstream station.

If the downstream buffer level is greater than Th but less than Th2, 50% of the activity is allocated to the downstream buffer. Finally, if the downstream buffer level is greater than Th2, 0% of the activity is allocated to the downstream buffer.
5 Simulation environment

The simulation environment consists of two parts: the first concerns the design of the assembly line by the mathematical models described above, while the second part describes the simulation model and the relative performance measures.

5.1 Assembly line design

The proposed approach is tested using an illustrative example extracted from the classical assembly line balancing dataset [1] and [26] (instance_n=20_525). Table 1 reports the process time and precedence constraints of the tasks of the numerical example.

The mathematical model solutions are provided by the Lingo® software package.

The solution of the basic model is shown in Table 2 considering four stations. Table 2 shows the solution without sharing activities and sharing activities with two fractional percentages 10% and 25% (parameter K). The parameter KF is fixed to 0.9 for all simulations, that is 0.1 (1-KF) the lower level of the fractional task that can be assigned to a station. As the reader can see, task sharing reduces the cycle time from 679 to 676.5. The two fractional percentage models have the same cycle time.

Table 3 reports the solutions of the redundancy model using the cycle time of the base model without sharing as a constraint. It has considered 4 and 10 as the maximum number of tasks shared between the stations. The cycle time obtained by the different solutions is closer to the cycle time of the base model.

5.2 Simulation data and performance

Table 4 reports the unforeseen events studied in the simulation models to evaluate the performance of the proposed methods. The unforeseen events considered are the failures with two Mean Time Between Failures (MTBF) following an exponential distribution. The Mean Time To Repair (MTTR) is considered similar to the cycle time following an exponential distribution. The third event is the fluctuations of the processing time of the stations following a normal distribution with 25% of standard deviations.

The simulation model to evaluate the performance is developed in Simul8® and considers a production line with the higher throughput rate possible with the raw items always available for the first station. In this way, the results do not depend on the particular inter-arrival time of the items. The simulations are conducted following the terminating simulation approach. For each experimental class, a number of replications able to assure a 5% confidence interval and 95% of confidence level for each performance measure have been conducted.

The performance measures considered are the following:
• Throughput rate of the assembly line;
• The average time in the system of the items;
• The standard deviation of the average time in systems;
• The work in process of the assembly line as the sum of the items in queues.

6 Numerical results

The numerical results are reported as percentage difference compared to the model without tasks sharing. Table 5 reports the identificatory used for each model simulated in the following graphs.

### Table 1 Process time and precedence of the tasks

| Task | Time [sec] | Precedence | Task | Time [sec] | Precedence |
|------|------------|------------|------|------------|------------|
| 1    | 71         | –          | 11   | 124        | 1-2-3-6    |
| 2    | 40         | –          | 12   | 82         | 10         |
| 3    | 173        | –          | 13   | 214        | 11         |
| 4    | 53         | –          | 14   | 37         | 11         |
| 5    | 176        | –          | 15   | 182        | 11         |
| 6    | 226        | 5          | 16   | 224        | 11         |
| 7    | 67         | 4          | 17   | 92         | 12         |
| 8    | 177        | 5          | 18   | 131        | 15         |
| 9    | 81         | 5          | 19   | 88         | 15         |
| 10   | 267        | 5          | 20   | 200        | 17         |

Total process time = 2705

### Table 2 Solutions of the base model

| Station 1 | Tasks assigned | Processing time |
|-----------|----------------|-----------------|
|           | 5-6-10         | 669             |
|           | 1-2-3-7-8-11-15-19 | 678             |
|           | 4-13-14-18     | 679             |
|           | 9-12-16-17-20  | 679             |

Cycle time = 679

| Tasks assigned | Processing time |
|----------------|-----------------|
| Max redundancy 2 | 676.5 |
| Fraction 10% | 676.5 |
| Shared tasks—percentage | 676.5 |
| Processing time | 676.5 |

| Station 1 | Tasks assigned | Processing time |
|-----------|----------------|-----------------|
|           | 1-5-6-8        | 676.5           |
|           | 2-3-11-15-18   | 676.5           |
|           | 7-9-10-12      | 676.5           |
|           | 13-14-17-19-20 | 675.8           |
|           | 13-14-17-19-20 | 675.8           |

Cycle time = 676.5

| Tasks assigned | Processing time |
|----------------|-----------------|
| Max redundancy 2 | 676.5 |
| Fraction 25% | 676.5 |
| Shared tasks—percentage | 676.5 |
| Processing time | 676.5 |

| Station 1 | Tasks assigned | Processing time |
|-----------|----------------|-----------------|
|           | 1-5-6-8        | 676.5           |
|           | 2-3-11-15-16-19 | 675.5         |
|           | 7-10           | 675.5           |
|           | 9-12-14-17-18-20 | 675.5     |

Cycle time = 676.5
Figure 4 shows the throughput rate of the models evaluated. The better values are obtained for the process time variability for cases 10 and 14. These two cases are characterized by a low number of tasks shared and the dynamic control of the shared processing time. These cases are better also when the failures are more frequent F2 (MTBF 100).

Case 17 leads to better results for the line without disruptions and failure with MTBF 200.

Then, the dynamic model is always the better approach to improve the throughput rate. The higher tasks shared and task fractional percentage (10%) is adapted when the disruptions do not occur or the failures are less frequent. The disruption in processing time and higher failure frequencies is characterized by lower tasks shared and lower fractional percentage (25%).

Figure 5 shows the throughput time of the items. For this performance, more relevant benefits can be obtained from the redundancy models. The model without disruption does not have relevant benefits from the shared tasks. The benefit is relevant when disruptions occur. The better cases when the disruption on processing time occurs are 10, 14, and 17; these cases are characterized by the dynamic approach for sharing the processing time of the shared tasks. The failure with lower frequencies has the better cases for 9 and 15, while the higher frequency of the failures has the better performance for case 15. Furthermore, the increase in shared

### Table 3: Solutions of the redundancy model

| Station 1 | Tasks assigned | Fraction 10% | Processing time |
|-----------|----------------|--------------|----------------|
| 3-4-5-10  | 1 (10%)        | 676.1        |
| 2-6-11-15 | 1 (90%)        | 676.4        |
| 7-8-14-16-19 | 9 (50%) | 676.3        |
| 12-17-18-20 | 13 (80%)     | 676.2        |

**Cycle time = 676.4**

| Station 1 | Tasks assigned | Fraction 25% | Processing time |
|-----------|----------------|--------------|----------------|
| 1-3-5-6   | 2(75%)         | 676          |
| 4-10-11-15| 2(25%)–9(50%)  | 676.5        |
| 12-16-17-18-19 | 9(50%)–14(50%) | 676         |
| 7-8-13-20 | 14(50%)        | 676.5        |

**Cycle time = 676.5**

| Station 1 | Tasks assigned | Fraction 25% | Processing time |
|-----------|----------------|--------------|----------------|
| 3-10-12   | 2(30%)–4(40%)–6(30%)–7(30%) | 676.5 |
| 3-8       | 2(70%)–4(60%)–6(70%)–7(70%) | 676.5 |
| 1-9-11-20 | 13(50%)–14(70%)–15(20%)–16(10%) | 676.5 |
| 18        | 13(50%)–14(30%)–15(80%)–16(90%) | 676.5 |

**Cycle time = 676.5**

| Station 1 | Tasks assigned | Fraction 25% | Processing time |
|-----------|----------------|--------------|----------------|
| 5-6       | 2(75%)–7(75%)–10(50%)–12(25%) | 676 |
| 3-8-17    | 2(25%)–7(25%)–10(50%)–12(75%) | 677 |
| 1-11-19   | 9(75%)–14(50%)–15(75%)–16(50%)–18(50%) | 676.25 |
| 13-20     | 9(25%)–14(50%)–15(25%)–16(50%)–18(50%) | 675.75 |

**Cycle time = 677**

### Table 4: Unforeseen events

| Unforeseen event | MTTR [sec] | MTBF [sec] |
|------------------|------------|------------|
| Failures         |            |            |
| Case 1 [F1]      | Exponential (600) | Exponential (1200) |
| Case 2 [F2]      | Exponential (600) | Exponential (6000) |
| Processing time  |            |            |
| Case 3 [PT]      | Normal (average, 25%) |          |

Figure 4 shows the throughput rate of the models evaluated. The better values are obtained for the process time variability for cases 10 and 14. These two cases are characterized by a low number of tasks shared and the dynamic control of the shared processing time. These cases are better also when the failures are more frequent F2 (MTBF 100).

Case 17 leads to better results for the line without disruptions and failure with MTBF 200.
### Table 5  Model identificatory

| Tasks shared 2 K = 10% | id  | Tasks shared 2 K = 10%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | id  |
|-----------------------|-----|--------------------------------------------------------|-----|
| Tasks shared 2 K = 25%| 1   | Tasks shared 2 K = 10%—queue < 1                       | 10  |
| Tasks shared 4 K = 10%| 2   | Tasks shared 2 K = 25%—queue < 2                       | 11  |
| Tasks shared 4 K = 25%| 3   | Tasks shared 2 K = 25%—queue < 3                       | 12  |
| Tasks shared 10 K = 10%| 4   | Tasks shared 2 K = 25%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 13  |
| Tasks shared 10 K = 25%| 5   | Tasks shared 2 K = 25%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 14  |
| Tasks shared 2 K = 10%—Th = 1| 6   | Tasks shared 4 K = 10%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 15  |
| Tasks shared 2 K = 10%—Th = 2| 7   | Tasks shared 4 K = 25%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 16  |
| Tasks shared 2 K = 10%—Th = 3| 8   | Tasks shared 10 K = 10%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 17  |
| Tasks shared 2 K = 10%—Th = 3| 9   | Tasks shared 10 K = 25%—dynamic (Th = 1; Th1 = 2; Th2 = 3) | 18  |

Fig. 4  Throughput rate

![Throughput rate graph](image)

Fig. 5  Throughput time

![Throughput time graph](image)
tasks (10) leads to a dramatic increase in time of throughput time.

Figure 6 shows the coefficient of variation of the throughput time to evaluate the stability of this performance. In the case of process time disruptions, case 17 is the better case and it is the same as the reduction of the throughput time. Failure case 1 is characterized by the best cases for 9 and 15, while failure cases 2, 9, and 13. Also for this performance, the increment of the tasks shared leads to the worst performance.

Figure 7 shows the work in process of the production line. The better cases in case of disruption of the processing time are 10, 14, 15 and 17. The case 15 is the best for the Failure 1 and Failure 2.

### 6.1 Insights from the study

The analysis of the numerical results provides the following issues. The fractional assembly line in which the tasks are shared between two consecutive stations with a policy that can change dynamically the processing time assigned to these stations is an effective method to improve the throughput time, the coefficient of variation of the throughput time, work in process and slightly the throughput. The dynamic method to allocate the processing time between two consecutive stations leads to improve the performance considering disruption as the failures or fluctuations of the processing times. The results show how the number of tasks shared should be limited because the higher number of tasks shared leads to the
worse performance due to the higher complexity to manage the different processing times.

The main benefit of the proposed method concerns the reduction of the work in process of the production line and then, the reduction of throughput time with higher stability. The improvement of the throughput is lower and, it is higher considering the fluctuation of the processing times than the failures.

7 Conclusions and future development paths

The redundancy of the tasks in production lines can be a relevant design method to improve the performance when disruption occurs. This research starts from a model proposed to support the fractional assembly line problem to include the redundancy of the tasks shared.

Then, the mathematical model is modified to maximize the shared task with the optimized cycle time obtained with the model proposed in the literature. The combined use of the two models makes it possible to provide a series of design alternatives on the assembly line. Simulation is used to test alternatives with introducing failures and uncertain processing times.

Then, in response to the first research question asked: “Can the mathematical model proposed to support the design of the fractional assembly line with optimized cycle time by increasing the number of fractionated tasks?” The numerical results of the simulation demonstrate that among the alternatives proposed by the model some of them can improve the performance of the assembly line. The integration of the base and redundancy model provides several assembly line configurations that support the decision-maker to choose the better solution considering the possible unforeseen events.

The original contribution of this research is the potential improvement of the design of fractional assembly lines considering the redundancy and dynamic allocation of the processing time of shared tasks. The second answer of the research question asks: Can the dynamic allocation of the fraction of the tasks improve the performance of the assembly line when uncertain as short failures and processing time variations occur?

The simulation results highlight the improvements of the performance when failures or processing time fluctuations occur. The main performance measures are the throughput time, work in process, and stability of the throughput time.

These performance measures have improvements close to 90%, but these improvements are obtained with a limited number of tasks shared. If the number of the tasks shared is higher the performance measures do not improve.

The required investment costs in equipment to share tasks is the main limitation of the proposed research. Moreover, a more complex control policy of the tasks shared allocation will be investigated.

Funding Open access funding provided by Università degli Studi della Basilicata within the CRUI-CARE Agreement.

Declarations

Conflict of interest The authors have no competing interests to declare that are relevant to the content of this article.

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

References

1. Scholl A (1999) Balancing and sequencing assembly lines, 2nd edn. Physica, Heidelberg
2. Chen J, Askin RG (2006) Throughput maximization in serial production lines with worksharing. Int J Prod Econ 99(1–2):88–101
3. Anuar R, Bukchin Y (2006) Design and operation of dynamic assembly lines using work-sharing. Int J Prod Res 44(18–19):40434065
4. Graves SC, Redfield CH (1988) Equipment selection and task assignment for multiproduct assembly system design. Int J Flex Manuf Syst 1(1):31–50
5. Muller C (2013) Fault-tolerant flow-line design, an example form an automotive body shop. Open Access Master’s Theses, p 134
6. Muller C, Weckenborg C, Grunewald M, Spengler TS (2016) Consideration of redundancies in the configuration of automated flow lines. In: Mattfeld S, Brinkmann G (eds) Logistics management. Springer International Publishing, Cham
7. Müller C, Grunewald M, Spengler TS (2017) Redundant configuration of automated flow lines based on “Industry 4.0”-technologies. J Bus Econ 87(7):877–898
8. Nahas N (2017) Buffer allocation and preventive maintenance optimization in unreliable production lines. J Intell Manuf 28:85–93
9. Renna P (2019) Adaptive policy of buffer allocation and preventive maintenance actions in unreliable production lines. J Ind Eng Int 15:411–421
10. Cohen Y, Faccio M, Galizia FG, Mora C, Pilati F (2017) Assembly system configuration through Industry 4.0 principles: the expected change in the actual paradigms. In: Dochain D, Henrion
D. Peaucelle D (eds) Proceedings of the 20th IFAC World Congress, pp 14958–14963

11. Kamblea SS, Gunasekaran A, Ghadge A, Raut R (2020) A performance measurement system for industry 4.0 enabled smart manufacturing system in SMMEs—a review and empirical investigation. Int J Prod Econ 229:107853

12. Schlüter MJ, Ostermeier FF (2022) Dynamic line balancing in unpaced mixed-model assembly lines: a problem classification. CIRP J Manuf Sci Technol 37:134–142

13. Pasupa T, Suzuki S (2019) Impact of work-sharing on the performance of production line with heterogeneous workers. Int J Ind Eng Manag 10(4):284–302

14. Grzechca W, Foulds LR (2015) The assembly line balancing problem with task splitting: a case study. IFAC Papers On Line 48(3):2002–2008

15. Jeong I-J, Jeon S (2020) Balanceability of a work-sharing line using floating workers and its comparison with floating work strategy. Int J Prod Res. https://doi.org/10.1080/00207543.2020.1795291

16. Lopes TC, Michels AS, Sikora CGS, Molina RG, Magatão L (2018) Balancing and cyclically sequencing synchronous, asynchronous, and hybrid unpaced assembly lines. Int J Prod Econ 203:216–224

17. Lopes TC, Sikora CGS, Michels AS, Magatão L (2020) Mixed-model assembly line balancing with given buffers and product sequence: model, formulation comparisons and case study. Ann Oper Res 286:475–500

18. Lopes TC, Pastre GV, Michels AS, Magatão L (2020) Flexible multimanned assembly line balancing problem: model, heuristic procedure, and lower bounds for line length minimization. Omega 95:102063

19. Lopes TC, Brauner N, Magatão L (2021) Assembly line balancing with fractional task allocations. Int J Prod Res. https://doi.org/10.1080/00207543.2020.1866224

20. Kahan T, Bukchin Y, Menassa R, Ben-Gal I (2009) Back-up strategy for robots’ failures in an automotive assembly system. Int J Prod Econ 120(2):315–326

21. Muller C, Spengler TS, Sodhi MS (2014) Robust flowline design for automotive body shops. In: Guan Y, Liao H (eds) Proceedings of the 2014 Industrial and Systems Engineering Research Conference, pp 3077–3085

22. Gu X, Jin X, Ni J, Koren Y (2015) Manufacturing system design for resilience. Procedia CIRP 36:135–140

23. Koren Y, Gu X, Guo W (2018) Choosing the system configuration for high-volume manufacturing. Int J Prod Res 56(1–2):476–490

24. Renna P (2019) Evaluation of redundant configurations and backup stations to support fault tolerant flow line design. Int J Adv Manuf Technol 101:825–837

25. Weckenborg C, Kieckhäfer K, Müller C et al (2020) Balancing of assembly lines with collaborative robots. Bus Res 13:93–132. https://doi.org/10.1007/s40685-019-0101-y

26. Otto A, Otto C, Scholl A (2013) Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing. Eur J Oper Res 228(1):33–45

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.