Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 104 (2021) 134–139

www.elsevier.com/locate/procedia

54th CIRP Conference on Manufacturing Systems

Linear optimization for dynamic selection of resources in constrained assembly line balancing problems

Marcel Albus*[a], Carsten Seeber[a]

[a] *Fraunhofer Institute for Manufacturing Engineering and Automation IPA, Nobelstraße 12, 70569 Stuttgart, Germany*

* Corresponding author. Tel.: +49 711 970-1663; *E-mail address:* marcel.albus@ipa.fraunhofer.de

---

Abstract

Assembly line balancing problems have been much investigated in the last decades. Most work affects static problems in which process times and resources are known in advance due to the complexity and diversity of such problems. However, the past has shown that static resource allocation for dynamic process scheduling is insufficient due to volatile market demands required by the reconfigurable production paradigm. Consequent change of resource selection by production requests affect feasible assembly lines. This paper introduces a mixed integer programming based linear optimization approach for solving the dynamic resource allocation for assembly lines modelled as Assembly Line Balancing Problem.

Keywords: optimization; equipment selection; assembly line balancing;

---

1. Introduction and literature review

Assembly lines are used in mass production of standardized products to reduce cost and production times. An assembly line consists of multiple workstations along a conveyor belt or similar material handling equipment, each performing either one or more tasks, moving the product through the line. At each station, consequent operations are performed, based on an assembly order, to assemble the final product. To manufacture the product on an assembly line, the assembly operations have to be divided into a set of elementary operations named *tasks*. These tasks are either performed manually or through automated equipment at each workstation, which requires a *task time* until the processing is finished. A *cycle time* constraint leads to a fixed production rate and since a task is an elementary operation that cannot be further divided, the cycle time cannot be smaller than the largest task time [1]. Dedicated equipment is used to reduce the task times and leverage high efficiency while minimizing production cost [1]. Volatile market demands for flexible and automated equipment lead to increased market availability of production resources, resulting in intractable many combinations and possibilities of conceptual assembly line design if manually planned. Due to the flexibility of equipment, there are often multiple equipment alternatives available for each task, which is known as the *equipment selection problem* [2, 3]. The equipment selection problem describes only a part of the manufacturing system, without external disruptions, contrary to, for example, resilient manufacturing [4, 5]. However, not every equipment is efficient at all tasks, it may be the case that one particular equipment is better suited for one task, but not for another [6]. To counteract the increasing combinatorial possibilities for assembly line design many researchers focus on the area of *Simple Assembly Line Balancing* (SALB), with the following simplifications [1, 7–9]:

(i) One homogeneous product is produced
(ii) Paced line with a fixed cycle time
(iii) Deterministic task times
(iv) No assignment restrictions other than precedence constraints
(v) Serial layout with one-sided stations
(vi) All stations are equally equipped with respect to production capabilities
(vii) Maximize the line efficiency

In manufacturing companies many of the mentioned simplifications are not applicable. Simplification (iv) neglects individual company requirements such as space, cost or task restrictions. Equipment are a specialized production resource (e.g. robots) with a designated characteristics to perform one set of tasks, such as handling or joining, which violates simplification

10.1016/j.procir.2021.11.023

(vi). Production equipment are assigned to workstations only if the task requires these equipment, to reduce investment cost, which differs from the SALBP formulation of [1–3, 6]. This results in workstations equipped with multiple different equipment in order to fulfill all task based requirements, which again violates simplification (vi). Additionally, the optimization objective may be cost reduction instead of line efficiency, as in (vii). The characteristics of such equipment determine the processing time for a task, which is furthermore neglected by the simplifications stated above.

The problem of assigning equipment to workstations is addressed in various SALBP formulations through multiple different approaches: heuristic methods, meta-heuristic methods and exact methods [9, 10]. An heuristic approach is presented in [6, 11, 12] which may lead to local optimum solutions. The meta-heuristic approaches use genetic algorithms, ant colony or particle swarm optimization [3, 13] whereas exact methods use branch-and-bound methods [6, 14–16] or mixed integer programming [17, 18]. The branch, bound and remember algorithm [8, 19] is an exact method combining the branch-and-bound algorithm and a dynamic programming method. This might be regarded as the current best performer for SALBP, achieving the optimality for all of Scholl's 269 problem instances [16] in short running times [10]. Still, all these approaches neglect multiple equipment allocation in one workstation as part of a sequential, conveyor-connected assembly line, as well as variable task times based on equipment capabilities. Additionally, the division of tasks in joining, handling or positioning as some of the main assembly types is neglected [20, 21].

The *Generalized Assembly Line Balancing* (GALB) involves more complex problems such as: parallel workstations or tasks, unequally equipped workstations, problems involving stochastic processing times and problems considering mixed or multimodel lines [13]. The S- and GALB are *assembly line balancing problems* which are proven to be an NP-hard optimization problem [22, 23]. This encourages the point that a planner of assembly lines cannot take all possible solution combinations into consideration, but only a small fraction of the design space. Therefore, the conceptual planing of a new assembly line requires a lot of planning effort [24].

In this paper an approach is presented to support the planner in the exploration of the solution space. The paper addresses the problem of selecting suitable equipment for workstations and assign tasks to this equipment, based on their capabilities subject to precedence and cycle time constraints. This simultaneously solves the equipment selection and assembly line balancing problem. The result is a series of workstations, where multiple equipment are placed in each workstation, and a set of tasks is assigned to this workstation to be performed by the selected equipment. The problem is formulated as a special case of the GALBP with some simplifications from the SALBP formulations, such as:

(i) One homogeneous product is produced
(ii) Paced line with a fixed cycle time
(iii) Deterministic task times, but depending on the equipment

To address the aforementioned problem we use a resource database which contains all available production equipment for the manufacturing process, defined by a given *bill of processes* (BOP). This database is then filtered based on the requirements of every task, the mechanical characteristics as well as supported operations of the equipment (e.g. a handling task cannot be performed on a joining equipment) [25]. As a result, for each task a feasible solution space is created which consists of the equipment cost, task time to process the task and equipment number. It is assumed that each solution space consists of at least one feasible equipment. Yet, not all equipment share the same mechanical production capabilities which results in varying task times for the same task. An example solution space with arbitrary entries is shown in Table 1. These equipment vary in their capabilities to perform different task types (some can perform multiple task types), their time to process a task and their investment cost.

Objective functions in the literature are either a reduction of the number of workstations [26], a makespan minimization [27] or even multi-objective optimization [28]. However, the objective of our model is to minimize the total investment cost, given a predefined cycle time. Addressing the requirements of companies to enforce the processing of a subset of tasks on the same workstation, additional task constraints for the model are introduced. This can become necessary due to quality inspection or transport distance limitations. An exact mixed integer linear algorithm is developed which is capable of solving a real-world assembly line balancing problem with a set of solution spaces consisting of possible equipment in reasonable time, achieving better results than a manual planning solution.

The remainder of the paper is organized as follows: In section 2 the notation and problem formulation is introduced and the model is developed, starting from a decision variable formulation to a full model. In section 3 the model is illustrated with two example problems, one is a real-world assembly line balancing problem. Finally, the conclusions and future prospects are presented in section 4.

Table 1: Example solution spaces for 4 tasks with 9 possible equipment. The possible task types are *s* for *separation*, *h* for *handling* and *j* for *joining*

Task	**Equipment name**	**Possible task types**	**Task time (s)**	**Investment cost (€)**
1	E1	s, h, j	11.0	10 000
	E2	s	7.0	15 000
	E3	s	10.0	12 000
	E4	s	14.0	5 000
2	E1	s, h, j	3.0	10 000
	E5	h, j	5.0	7 500
	E6	h	5.5	6 000
3	E1	s, h, j	16.0	10 000
	E5	h, j	12.0	7 500
	E7	j	8.0	16 000
	E8	j	13.0	8 000
4	E2	s	9.5	15 000
	E3	s	10.0	12 000
	E9	s	7.0	15 000

2. Problem formulation

This section introduces the notation as well as our assumptions and presents an mixed integer programming formulation of the problem. This formulation also includes lower and upper bounds for the number of workstations which are used to enhance the efficiency of the algorithm. The problem is defined by the following parameters:

Variable name	Variable description
n	Number of tasks.
m	Number of workstations.
r	Number of equipment.
t_{ij}	Processing time of task i when performed by equipment j.
t_i	Fastest processing time of task i among all equipment.
ct	Cycle time.
m_{min}	Lower bound on the number of workstations, $m_{min} = \lceil \sum_{i=1}^{m} t_i/ct \rceil$.
EC_j	Cost of equipment type j.
P	Set of pairs of tasks (g, h) such that there is immediate precedence between them.
PT_i	The set of tasks which precede task i.
ST_i	The set of tasks which succeed task i.
E_i	The earliest workstation to which task i can be assigned, $E_i = \lceil (t_i + \sum_{k \in PT_i}^{m} t_k)/ct \rceil$.
L_i	The latest workstation to which task i can be assigned, $L_i = m_{min} + 2 - \lceil (t_i + \sum_{k \in ST_i}^{m} t_k)/ct \rceil$.
m_{max}	Upper bound on the number of workstations, $m_{max} = max(L_i)$.

We state the following assumptions to clarify the problem setting, the overall approach is shown in Figure 1:

(i) There is a given database containing all available production equipment for the manufacturing process. This database is filtered based on task requirements, the mechanical characteristics as well as supported operations of the equipment, into solution spaces for each task.

(ii) Each solution space consists of at least one feasible equipment and associates each equipment with a specific cost. The cost is the investment cost for the equipment.

(iii) The precedence relation between assembly tasks is known.

(iv) The assembly task is an elementary operation. As such, it cannot be further subdivided.

(v) The duration of a task is a deterministic floating-point value but depends on the equipment selected to perform the task.

(vi) A task can be performed at any station of the assembly line, provided that the equipment assigned to this station are capable of performing the task, and that precedence relations are satisfied.

(vii) The total duration time of tasks that are assigned to a given station should not exceed the predetermined cycle time.

(viii) The total duration time of tasks that are assigned to a given equipment should not exceed the predetermined cycle time.

(ix) Material handling, positioning, loading and unloading times are negligible or included in the duration of the task.

(x) The availability of equipment is not limited, i.e. every equipment can be acquired as often as necessary.

(xi) A subset of tasks can be enforced in one workstation.

(xii) Not all equipment in the solution space needs to be assigned.

An example for the solution space can be see in Table 1.

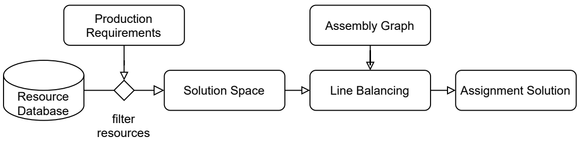


Fig. 1: Overview of the pipeline for the assembly line balancing.

This results in decision variables to address two different issues: (i) the equipment design issue, where the equipment has to be assigned to a workstation if it is selected, and (ii) the assignment of all tasks to the stations, such that all constraints are satisfied. The following binary decision variables correspond to the aforementioned issues, respectively. We define for every equipment j and every station k:

$$y_{jk} \begin{cases} 1, \text{ if equipment } j \text{ is assigned to workstation } k \\ 0, \text{ otherwise.} \end{cases}$$

In addition, we define for every task i, every equipment j and station k:

$$x_{ijk} \begin{cases} 1, \text{ if task } i \text{ is performed by equipment } j \text{ at workstation } k \\ 0, \text{ otherwise.} \end{cases}$$

The following is the mixed integer programming formulation of the problem:

$$MIN \sum_{j=1}^{r} \sum_{k=1}^{m_{max}} EC_j \cdot y_{jk} \tag{1}$$

subject to

$$\sum_{j=1}^{r} x_{ijk} = \sum_{j=1}^{r} x_{pjk} \qquad \forall j, i \tag{2}$$

with $p = i + 1, \dots, n$

$$\sum_{j=1}^{r} \sum_{k=E_i}^{L_i} x_{ijk} = 1 \qquad \forall i \tag{3}$$

$$\sum_{i=1}^{n} t_{ij} \cdot x_{ijk} \leq ct \cdot y_{jk} \qquad \forall j, k \tag{4}$$

$$\sum_{i=1}^{n} \sum_{j=1}^{r} t_{ij} \cdot x_{ijk} \leq ct \qquad \forall k \tag{5}$$

$$\sum_{j=1}^{r} \sum_{k=E_g}^{L_g} k \cdot x_{gjk} \leq \sum_{j=1}^{r} \sum_{l=E_h}^{L_h} l \cdot x_{hjl} \qquad \forall (g, h) \in P \tag{6}$$

$$x_{ijk} = \{0, 1\} \qquad \forall i, j, k \tag{7}$$

$$y_{jk} = \{0, 1\} \qquad \forall j, k \tag{8}$$

The objective function (1) represents a minimization formulation of the total investment cost for all resources. Note that L_i and m_{max} serve as upper bound for the latest workstation a task can be assigned to and the upper bound on the number of workstations, respectively. Constraint (2) allocates task i and task p to the same workstation. This allows for company specific requirements in the task to workstation assignment. The assignment constraint (3) implies that each task must be assigned to exactly one workstation. Constraint (4) ensures that the total task time at each equipment of the workstation cannot exceed the cycle time, constraint (4) also ensures that a task cannot be performed by an equipment in a workstation if this equipment is not assigned to the respective station. Due to the possibility of multiple equipment at one workstation, constraint (5) ensures that the total task time of all tasks at one workstation cannot exceed cycle time. Constraint (6) imposes the precedence relations of tasks and constraint (7) and (8) define the decision variables to be binary.

3. Case Study

We evaluate the developed model on two example problems, the first example is adapted from [6], the second one is a real-world example. The first example problem has three available equipment, 10 tasks and a cycle time constraint of 50s. These three equipment have varying task times and cost. Equipment 1 (E1) can perform all tasks while E2 is faster, but only capable of performing some tasks and E3 being the least expensive, yet slowest equipment. This example problem is without additional constraints, such as the requirement to perform a subset of tasks in one workstation, or similar. Additionally it is without defined task-types, but the available equipment can be seen as a resource database which can be processed, as seen in the overview Figure 1, into a solution space for every task, shown in Table 2. For more detailed information on the example we refer to [6]. An optimal configuration is calculated using *Google or-tools* with a linear optimization *CP-SAT* solver in less than 0.05s while reproducing the optimal investment cost solution of $360 000 with four workstations, illustrated in Table 3. A dash in the investment cost column of Table 3 indicates that the equipment is already purchased for this workstation due to an earlier task to equipment assignment. Therefore, the investment cost for the particular equipment is only listed once. In this example problem the single equipment per workstation assignment is a result of the capabilities of equipment E1 or E2, which can process close to every task. This makes an assignment of multiple equipment per workstation unnecessary. If we reduce the capabilities of the resources to perform only a small fraction of selected tasks a multi-equipment per workstation solution is achieved.

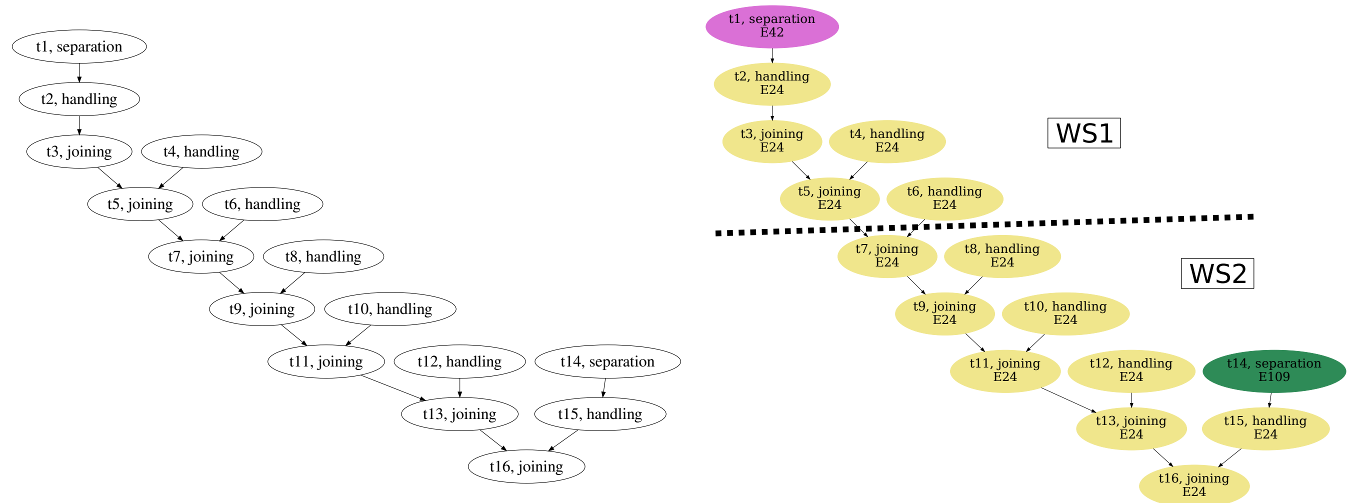
Table 2: The solution space for the first example problem, generated based on the information by [6]. Task types are omitted because they aren't defined for this problem.

Task	**Equipment**	**Task time (s)**	**Investment cost ($)**
1	E1 E2	8 6	100 000 100 000
2	E1 E3	13 14	100 000 60 000
3	E1 E2	49 40	100 000 100 000
4	E1 E3	15 17	100 000 60 000
5	E1 E2	18 14	100 000 100 000
6	E1 E2 E3	15 12 20	100 000 100 000 60 000
7	E1 E2	10 8	100 000 100 000
8	E1 E2 E3	10 8 13	100 000 100 000 60 000
9	E1 E3	33 38	100 000 60 000
10	E1 E3 E3	25 20 28	100 000 60 000 60 000

Table 3: The balancing solution for the first example. Task types are omitted because they aren't defined for this problem. Total investment cost: $360 000.

Workstation	**Task**	**Equipment**	**Investment cost ($)**
1	1 3	E2 E2	100 000 -
2	2 5 6	E1 E1 E1	100 000 - -
3	9	E3	60 000
4	5 7 8 10	E2 E2 E2 E2	100 000 - - -

The presented second example is a real-world assembly line balancing problem of a paced, manual assembly line with 16 tasks of types: $\{separation, handling, joining\}$, shown in Figure 2a. For this example, a resource database with 110 unique equipment is used, separated into solution spaces for each task where every entry corresponds to the capability of an equipment to perform this task, the necessary time to finish it and the investment cost (see Table 1 for an example). This depicts the characteristics of equipment (e.g. a *handling* equipment cannot perform a *joining* task) to perform one, or multiple, of the aforementioned task types with different trade-offs regarding cost and efficiency. An additional requirement for this example is that tasks of type *separation* directly preceding a *handling* task must be assigned to the same workstation. This requirement can is achieved with the definition of constraint (2):



(a) Assembly graph used for the example.

(b) Assembly graph colored with respect to the assigned equipment.

Fig. 2: Both pictures show the assembly graph used for the example, where the name indicates the task number and task type. 2a shows the precedence relations and 2b is colored based on the chosen equipment listed as E* below the task number and type. The colors indicate identical equipment for the task. The dashed line indicates the different workstations, all equipment above are assigned to workstation 1 (WS1) and all equipment below are assigned to workstation 2 (WS2).

$\sum_{j=1}^{r} x_{ijk} = \sum_{j=1}^{r} x_{pjk}$ with $p = i + 1$ if $type(i)$ = separation and $type(p)$ = handling.

This problem is solved with our optimal algorithm optimizing for minimal investment cost (1), determining equipment selection, the task to equipment assignment and equipment to workstation assignment subject to a global cycle time constraint of 16.5 seconds. The optimal configuration is again calculated using *Google or-tools* with a *CP-SAT* solver and is shown in Table 4. An assembly graph colored with the task to equipment assignment as well as the equipment to workstation assignment is shown in Figure 2b. The dashed line shows the separation between two workstations, all equipment in the upper half are assigned to workstation 1 and all equipment in the lower half to workstation 2. Table 4 shows the corresponding costs for each equipment, the assigned tasks and the task types performed by each equipment. A dash in the investment cost column of Table 4 indicates that the equipment is already purchased for this workstation due to an earlier task to equipment assignment. Therefore, the investment cost for the particular equipment is only listed once.

Table 4: The balancing solution for the assembly graph in Figure 2a subject to the resource database. Task type $\{s, h, j\}$ corresponds to $\{separation, handling, joining\}$, respectively. Total investment cost: 47 085€.

Workstation	**Task**	**Equipment**	**Task type**	**Investment cost (€)**
1	1	E42	s	10 030
	2	E24	h	10 774
	3	E24	j	-
	4	E24	h	-
	5	E24	j	-
	6	E24	h	-
2	7	E24	j	10 774
	8	E24	h	-
	9	E24	j	-
	10	E24	h	-
	11	E24	j	-
	12	E24	h	-
	13	E24	j	-
	14	E109	s	15 507
	15	E24	h	-
	16	E24	j	-

The solution uses 14 times equipment 24 (E24), 1-time E42 and E109 with a total investment cost of 47 085€ and was calculated in 17.24 seconds. This outperforms the manual solution with a total investment cost of 49 327€ and effort of multiple days with the same resource database as input data. Contrary to the solution of the first example, multiple equipment are assigned to one workstation to achieve the minimum investment cost. This is due to the limited capabilities of the available equipment to perform multiple, or all tasks, as E2 and E1 of the first example are capable of. As a conclusion, the capabilities of the equipment to perform all tasks, or a multitude of them, in the first example are unrealistic in a real-world production environment. This confirms the assumptions of the introduction, that equipment are rather limited in their abilities because they are highly specialized production resources to perform a set of tasks and hence achieve high efficiency.

High occurrence of equipment 24 indicates that the available equipment in the resource database are unevenly distributed concerning the capabilities of the equipment. Equipment 24's dominating occurrence indicates this particular equipment is capable of multiple task types in combination with low investment cost. An important conclusion drawn from this example is that the whole solution is highly dependant on valid and realistic input data, in our case the resource database. Our solution is deterministic on the objective function, i.e. the investment cost, but stochastic on the assigned resources if multiple equipment combinations meet the requirements.

4. Conclusion

In this paper, a mixed integer linear programming mathematical model is developed to solve the equipment selection and assembly line balancing of a flexible assembly line that consists of multiple different assembly equipment. The purpose of this model is to choose multiple equipment to assign to workstations of the paced assembly line and allocate tasks to these equipment, to minimize the total investment cost of the line. This problem is NP-hard since the simplified case of it is the simple assembly line balancing problem, which is known to be NP-hard. We present a formulation of the problem and a mathematical model to solve it, including constraints to account for additional requirements, such as assembly tasks performed on the same workstation based on their type. Yet, different sorts of requirements can be included with the formulation as well. This model is combined with a resource database containing possible equipment and cost for a real-world problem. This problem highlights the demand for multiple equipment to workstation assignments as well as additional constraints. However, the solution to the problem has shown flaws in the resource database but it showcases the ability to solve the problem for more than 100 possible equipment in a reasonable time. Additionally, it points out that real-world manufacturing problems use specialized equipment to leverage efficiency and high production rates which is contrary to example problems with highly capable equipment. Future research could investigate multi-objective optimization including required space or variable costs, which the current model does not account for.

References

[1] C. Becker, A. Scholl, A survey on problems and methods in generalized assembly line balancing, in: European Journal of Operational Research, Vol. 168, 2006, pp. 694–715. doi:10.1016/j.ejor.2004.07.023.

[2] J. Bukchin, J. Rubinovitz, A weighted approach for assembly line design with station paralleling and equipment selection, IIE Transactions (Institute of Industrial Engineers) 35 (1) (2003) 73–85. doi:10.1080/07408170304429.

[3] J. Oesterle, L. Amodeo, F. Yalaoui, A comparative study of Multi-Objective Algorithms for the Assembly Line Balancing and Equipment Selection Problem under consideration of Product Design Alternatives, Journal of Intelligent Manufacturing 30 (3) (2019) 1021–1046. doi:10.1007/s10845-017-1298-2.

[4] D. Mourtzis, J. Angelopoulos, N. Panopoulos, Robust engineering for the design of resilient manufacturing systems, Applied Sciences (Switzerland) 11 (7) (2021) 3067. doi:10.3390/app11073067.
URL https://doi.org/10.3390/app11073067

[5] A. Kusiak, Fundamentals of smart manufacturing: A multi-thread perspective (jan 2019). doi:10.1016/j.arcontrol.2019.02.001.

[6] J. Bukchin, M. Tzur, Design of flexible assembly line to minimize equipment cost, IIE Transactions (Institute of Industrial Engineers) 32 (7) (2000) 585–598. doi:10.1080/07408170008967418.

[7] I. Baybars, A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem, Management Science 32 (8) (1986) 909–932. doi:10.1287/mnsc.32.8.909.

[8] E. C. Sewell, S. H. Jacobson, A branch, bound, and remember algorithm for the simple assembly line balancing problem, INFORMS Journal on Computing 24 (3) (2012) 433–442. doi:10.1287/ijoc.1110.0462.

[9] O. Battaïa, A. Dolgui, A taxonomy of line balancing problems and their solutionapproaches, in: International Journal of Production Economics, Vol. 142, 2013, pp. 259–277. doi:10.1016/j.ijpe.2012.10.020.

[10] Z. Li, I. Kucukkoc, Q. Tang, A comparative study of exact methods for the simple assembly line balancing problem, Soft Computing 24 (15) (2020) 11459–11475. doi:10.1007/s00500-019-04609-9.

[11] L. Capacho, R. Pastor, O. Guschinskaya, A. Dolgui, Heuristic methods to solve the alternative subgraphs assembly line balancing problem, in: 2006 IEEE International Conference on Automation Science and Engineering, CASE, Institute of Electrical and Electronics Engineers Inc., 2006, pp. 501–506. doi:10.1109/COASE.2006.326932.

[12] L. Capacho, R. Pastor, A. Dolgui, O. Guschinskaya, An evaluation of constructive heuristic methods for solving the alternative subgraphs assembly line balancing problem, in: Journal of Heuristics, Vol. 15, 2009, pp. 109–132. doi:10.1007/s10732-007-9063-x.

[13] J. Oesterle, L. Amodeo, Comparison of Multiobjective Algorithms for the Assembly Line Balancing Design Problem, IFAC-PapersOnLine 49 (12) (2016) 313–318. doi:10.1016/j.ifacol.2016.07.623.

[14] A. Scholl, N. Boysen, M. Fliedner, Optimally solving the alternative subgraphs assembly line balancing problem, Annals of Operations Research 172 (1) (2009) 243–258. doi:10.1007/s10479-009-0578-4.

[15] A. Scholl, M. Fliedner, N. Boysen, Absalom: Balancing assembly lines with assignment restrictions, European Journal of Operational Research 200 (3) (2010) 688–701. doi:10.1016/j.ejor.2009.01.049.

[16] A. Scholl, R. Klein, SALOME: A bidirectional branch-and-bound procedure for assembly line balancing, INFORMS journal on Computing 9 (4) (1997) 319–334.

[17] C. Öztürk, S. Tunalı, B. Hnich, M. A. Örnek, Balancing and scheduling of flexible mixed model assembly lines, Constraints 18 (3) (2013) 434–469. doi:10.1007/s10601-013-9142-6.

[18] L. A. Wolsey, Integer programming, Vol. 52, John Wiley & Sons, 1998.

[19] D. R. Morrison, E. C. Sewell, S. H. Jacobson, An application of the branch, bound, and remember algorithm to a new simple assembly line balancing dataset, European Journal of Operational Research 236 (2) (2014) 403–409. doi:10.1016/j.ejor.2013.11.033.

[20] VDI Richtlinie, VDI Richtlinie 2860: Montage-und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole, Düsseldorf: VDI-Verlag (1990).

[21] DIN 8580: 2003-09, Fertigungsverfahren: Begriffe, Einteilung, Beuth Verlag Berlin, 2003.

[22] R. M. Karp, Reducibility among combinatorial problems, in: Complexity of computer computations, Springer, 1972, pp. 85–103.

[23] A. Scholl, Balancing and sequencing of assembly lines, Tech. rep., Darmstadt Technical University, Department of Business Administration (1999).

[24] B. Rekiek, F. Pellichero, P. De Lit, E. Falkenauer, A. Delchambre, P. De, E. Falkenauer, A. Delchambre, N. E. Mastorakis, Towards physical layout of assembly lines, in: Progress in Simulation, Modeling, Analysis and Synthesis of Modern Electrical and Electronic Devices and Systems, World Scientific Engineering Society Press, 1999, pp. 307–312.

[25] M. Fechter, C. Seeber, S. Chen, Integrated Process Planning and Resource Allocation for Collaborative Robot Workplace Design, in: Procedia CIRP, Vol. 72, Elsevier B.V., 2018, pp. 39–44. doi:10.1016/j.procir.2018.03.179.

[26] M. Gansterer, R. F. Hartl, One-and two-sided assembly line balancing problems with real-world constraints, International Journal of Production Research 56 (8) (2018) 3025–3042. doi:10.1080/00207543.2017.1394599.
URL https://doi.org/10.1080/00207543.2017.1394599

[27] Z. Li, M. N. Janardhanan, Q. Tang, P. Nielsen, Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line, Engineering Optimization 50 (5) (2018) 877–893. doi:10.1080/0305215X.2017.1351963.
URL https://doi.org/0305215X.2017.1351963

[28] M. Rabbani, R. Siadatian, H. Farrokhi-Asl, N. Manavizadeh, Multi-objective optimization algorithms for mixed model assembly line balancing problem with parallel workstations, Cogent Engineering 3 (1) (2016). doi:10.1080/23311916.2016.1158903.