emerald PUBLISHING

Assembly Autor

Mixed-integer linear programming models for the type-II resource-constrained assembly line balancing problem

Journal:	*Assembly Automation*
Manuscript ID	AA-10-2021-0140.R3
Manuscript Type:	Original Article
Keywords:	Assembly line balancing, Resource constraints, Type-II problem, Mixed-integer linear programming
Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.	
Supporting_Information.7z	

SCHOLARONE™
Manuscripts

ORIGINAL ARTICLE: RESEARCH PAPER

Mixed-integer linear programming models for the type-II resource-constrained assembly line balancing problem

Adalberto Sato Michels* and Alysson M. Costa

School of Mathematics and Statistics, University of Melbourne, Melbourne, Australia

ARTICLE HISTORY
Compiled June 13, 2022

ABSTRACT

Purpose – Resource-constrained assembly lines are widely found in industries that manufacture complex products. In such lines, tasks may require specific resources to be processed. Therefore, decisions on which tasks and resources will be assigned to each station must be made. When the number of available stations is fixed, the problem's main goal becomes the minimisation of cycle time (type-II version). This variant of the problem lacks investigation in the literature.

Design/methodology/approach – In this paper, we propose Mixed Integer Linear Programming (MILP) models to minimise cycle time in resource-constrained assembly lines, given a limited number of stations and resources. Dedicated and alternative resource types for tasks are considered in different scenarios.

Findings – Besides, past modelling decisions and assumptions are questioned. We discuss how they were leading to suboptimal solutions and offer a rectification.

Originality/value – The proposed MILP models are applied to an existing dataset, results are compared against a Constraint Programming (CP) model, and new optimal solutions are obtained. Moreover, a dataset extension is proposed due to the simplicity of the current one and instances up to 70 tasks are optimally solved.

Practical implications – The proposed models and dataset fulfil more practical concerns by taking into account characteristics found in real-world assembly lines.

KEYWORDS
Assembly line balancing; Resource constraints; Type-II problem; Mixed-integer linear programming

1. Introduction

The design and operation of production lines have originated many complex combinatorial problems in academic research. The most popular one is the Assembly Line Balancing Problem (ALBP), which is broadly discussed in the literature (Boysen, Schulze, and Scholl 2021) under several classifications. The most basic version of the problem, the Simple Assembly Line Balancing Problem (SALBP), was formally defined by Baybars (1986) with several simplification hypotheses. The SALBP comprises the problem of deciding the assignment of a set of tasks to stations. These tasks cannot be arbitrarily assigned to any station; they must respect precedence constraints commonly represented by a precedence graph. Moreover, stations are organised in a straight, serial line. Such layout enforces the line's pace to take the time of the most

*Corresponding author. Email: a.satomichels@unimelb.edu.au

Assembly Automation

loaded station, which is known as the cycle time. Finally, the assembly line produces a unique model of a single product. The most common goals in a SALBP are the minimisation of the number of stations (SALBP-1) and the minimisation of the cycle time (SALBP-2). Important contributions to the SALBP literature were surveyed by Scholl and Becker (2006), which also list various exact algorithmic solution approaches and practical applications.

Another simplification hypothesis of SALBPs is assuming that any given task can be performed in any station. The underlying assumption is that all stations are homogeneously manned or equipped with the necessary resources, i.e. with the same workers, tools, and/or machines (Baybars 1986). Nonetheless, real-world assembly lines usually rely on limited heterogeneous resources, e.g. usage of special equipment and/or professional workers. In order to complete the task set, specific resources must also be assigned to different stations (Agpak and Gökçen 2005). Several examples of this application can be found in factories: cutting tools that make metal sheets in a specific width and shape, welding tools that can access certain locations in a car body, drilling tools for particular materials, and workers/robots that are skilled to operate such machinery (Boysen, Fliedner, and Scholl 2007).

By extending the SALBP to contemplate this feature, Agpak and Gökçen (2005) proposed the Resource-Constrained Assembly Line Balancing Problem (RCALBP). Becker and Scholl (2006) state that the RCALBP is a generalisation of the previous SALBP and surveys it alongside various practical extensions. Recently, Alakas, Pinarbasi, and Yüzükirmizi (2020) proposed a variant of the RCALBP with the goal of minimising both the cycle time and the resource usage for a given number of stations. Sample scenarios were solved with a constraint programming (CP) approach, and optimal solutions were found for all tested instances. Such approach had been successfully applied in a previous CP model that dealt with the ALBP with assignment restrictions (Pinarbasi, Alakas, and Yuzukirmizi 2019). However, the CP method modelled by Alakas, Pinarbasi, and Yüzükirmizi (2020) has one constrictive assumption: the objective of minimising cycle time is achieved by compulsorily assigning one resource at each station. In practice, by allowing resources to be distributed between stations more freely, it may be possible to achieve (i) the same cycle time with fewer resources or (ii) lower cycle times with the same number of resources.

Motivated by such potential improvements, the type-II Resource-Constrained Assembly Line Balancing Problem (RCALBP-2) is further investigated in this paper. Past modelling decisions and assumptions are questioned: we extend the previous concepts on the limit and availability of resources and propose new Mixed-Integer Linear Programming (MILP) models to flexibly cope with these more realistic scenarios, such as having a welding and drilling tool in the same station. We also discuss how these restrictive concepts were leading to suboptimal solutions and offer a rectification.

The remainder of this paper is organised as follows. In Section 2, a literature review is conducted for the RCALBP, emphasising particular objective functions, extensions, and methodologies investigated by each study. Section 3 precisely defines and explains the characteristics of a RCALBP-2 considered in this paper. Section 4 presents new MILP models, which contemplate different extensions regarding multiple-resource types and availability, as well as budget-oriented constraints, taking into account characteristics found in real-world assembly lines. Computational experiments use the dataset proposed by Alakas, Pinarbasi, and Yüzükirmizi (2020) as a benchmark. Furthermore, a dataset expansion is proposed in order to fulfil more practical concerns. All instances are solved and results are discussed throughout Section 5. Lastly, Section 6 concludes this paper by highlighting its main contributions and pro-

For Peer Review

viding further research directions.

2. Literature review

The Resource-Constrained Assembly Line Balancing Problem (RCALBP) was introduced by Agpak and Gökçen (2005) along with a mathematical mixed integer programming (MIP) model. They generalised the usage of special equipment and/or professional workers as resources. Each resource is capable of performing different processes in a factory. Besides, two resource classifications were considered in their study: *dedicated* and *alternative*. While the former considers that each task can only be performed by one resource type, the latter makes the problem more flexible by defining tasks that can be assigned to different resources. A task can only be assigned to a station with the required resource (dedicated RCALBP) or one among the possible required resources (alternative RCALBP). For instance, if a task is assigned to a station, the resource type which can perform such task must also be assigned to that station in a dedicated RCALBP. Correspondingly, an alternative RCALBP would require at least one of the resource types which can perform the task to be assigned to that station, meaning that some tasks can be performed by alternative resources. The proposed MIP approach's goal was to establish a balance of the assembly line that minimises the number of stations and resources. For this purpose, two MIP models (dedicated and alternative) are developed and applied to numerical examples, which are tackled by using a black-box commercial solver.

The RCALBP was further generalised when the *multiple* and *concurrent* versions of the problem were proposed by Corominas, Ferrer, and Pastor (2011). Previously, specific assumptions limited the problem: dedicated cases assume that tasks need an exclusive resource type (A), whereas the alternative version states that tasks may require one out of two or more resources (A $\vee$ B). They extended the problem by presenting cases in which each task needs dedicated, alternative, multiple, and/or concurrent resources. Some examples are provided (Corominas, Ferrer, and Pastor 2011): (3A), (A $\wedge$ 4B $\wedge$ 3C), (3A $\vee$ 2B $\vee$ C), (A $\wedge$ B) $\vee$ (2C $\wedge$ D), or (A $\vee$ B) $\wedge$ (2C $\vee$ D). Out of that, three MIP variations are presented to mathematically model the problem: two in the conjunctive normal form (CNF-1 and CNF-2) and one in the disjunctive normal form (DNF), as defined by Friedman (1985). The equivalent linearised constraints for the generalised RCALBP containing logical expressions in CNF ($\wedge$ of $\vee$s form) or DNF ($\vee$ of $\wedge$s form) were developed. These three modelling variations were applied to a 560-instance dataset with up to 80 tasks in a black-box optimiser. Their effectiveness was compared and CNF-2 obtained the best results, with a greater number of instances solved to optimality and fewer instances with no feasible solution. Finally, an upper bound limit on the number of available resources was firstly considered in a mathematical model. Due to the complexity increase, the CNF-2 model with active resource limits was not as efficient as the model without these constraints.

Quite a few papers have developed ad hoc solution methods to solve specific versions of the RCALBP in the literature. Kao et al. (2010) proposed a shortest path algorithm to tackle the dedicated RCALBP. The method was tested on a sample problem composed of 51 tasks and obtained the same solution as the one found in Agpak and Gökçen (2005). Following that study, Kao, Yeh, and Wang (2011) developed a heuristic based on ranked positional weight rules. In order to show the effectiveness of their proposed method, numerical results were obtained from different sample problems up

to 21 tasks. Kamarudin and Ab Rashid (2017) applied a genetic algorithm (GA) to the dedicated RCALBP with skilled workers. Their GA used two rank-based crossover mechanisms and was tested on benchmark problems up to 148 tasks. Quyen, Chen, and Yang (2017) developed a hybrid GA to solve a RCALBP found in sewing lines of a footwear company. The experimental results showed that their algorithm outperformed the factory's manual procedure used at that time and Kao, Yeh, and Wang (2011)'s heuristic. Immediately after that, Quyen et al. (2017) proposed a dynamic programming (DP) and applied it to the same footwear company's sewing lines. Their DP was capable of obtaining equivalent or improved results for 16 out of 18 tested instances when compared to Quyen, Chen, and Yang (2017)'s GA.

Industrial extensions of the RCALBP have also been reasonably explored in the literature. For instance, when multiple workers are allowed to simultaneously perform tasks in the same station, the SALBP is extended to the so-called Multi-manned Assembly Line Balancing Problem (MALBP) (Michels et al. 2019). This concept can be integrated into the problem under study to create the Resource-Constrained Multi-manned Assembly Line Balancing Problem (RCMALBP). This extension raises an interesting practical question: should resources be assigned to workers or to stations? Yilmaz and Yilmaz (2017) considered the former: they proposed a mathematical programming model to minimise the number of workers, stations, and resources employed in the line. Workers can only perform tasks if they hold the necessary resource to do so. This model was tested with a numerical example to indicate its applicability. On the other hand, Chen, Cheng, and Li (2018) studied a practical case found in an automotive factory, in which resources (machines) are assigned to stations and shared among all workers also employed in the same station. A mathematical programming model was proposed to minimise the number of workers, stations, and resources. This model was then used as the basis for subsequent development of a suitable GA-based algorithm. Their proposed approach was compared to the traditional planning method. Results showed that better configurations could be obtained in terms of solution quality and production efficiency.

A similar extension is the Two-sided Assembly Line Balancing Problem (TALBP), in which both left-hand and right-hand sides of the line are used for manufacturing products (Bartholdi 1993). Yang et al. (2020) aimed to study the multi-constrained version of the TALBP, which involved positional, zoning, and synchronism constraints. They extended the basic TALBP-1 model and proposed a new mathematical programming model in which the cycle time is given. The weighted sum of the number of mated-stations and stations is minimised. They also designed an improved imperialist competitive algorithm (ICA) to solve the problem. Additionally, Yadav et al. (2020) developed a new mixed-integer mathematical programming model to tackle the parallel TALBP, a version in which both tools and tasks could be shared. The main aim of their model is maximising the workload on each station. Arguably, this goal works in the direction of minimising the line length and reducing idle times at each station. Their results compared the effects of these tools and tasks sharing approaches on efficiency, which was used as the performance measure.

Other industrial variants considering RCALBPs are related to line layout, designing costs, and multiple product models. These factors are important attributes for the rapid advancement of Industry 4.0 (Rahman, Janardhanan, and Nielsen 2020). Ege, Azizoglu, and Ozdemirel (2009) investigated the possibility of parallel stations in assembly lines containing tasks that require specified dedicated resources (tooling/equipment). The cost structure ratio suggested by Askin and Zhou (1997) was used to generate dataset instances for up to 70 tasks. They attempted to minimise

Assembly Automation

costs related to station opening and resources. For that, two branch-and-bound (B&B) algorithms were proposed: one for optimal solutions and one for near-optimal solutions. Ogan and Azizoglu (2015) and Michels et al. (2018) further extended Ege, Azizoglu, and Ozdemirel (2009)'s work in different directions. Ogan and Azizoglu (2015) analysed U-shaped assembly lines with alternative resources. An MILP model that minimises the total equipment cost over all stations and a B&B algorithm were proposed. Both approaches were tested in a dataset with instances of up to 30 tasks. Michels et al. (2018) examined robotic assembly lines with the possibility of parallel stations. Dedicated and alternative resources (robots and equipment) were considered in a cost-oriented MILP model. Their objective function minimises the purchase cost of the line, which is composed of platform and transporter robots along with their tools, stations, and track-motion systems. Practical extensions to consider welding tasks were modelled and tests were also conducted in a cost structure ratio based dataset (Askin and Zhou 1997). Moreover, an automotive robotic assembly line case study was solved. Finally, Pereira (2018) studied a cost-oriented, multi-model RCALBP found in a textile industry. A straightforward and an alternative MILP formulations were developed. The former minimises the weighted sum of costs associated to stations and resources whilst the latter minimises the total cost of the stations. The straightforward MILP uses variable sets to decide on task-station, item-station, and resource-station assignments. The alternative model makes use of a feasible assignment concept that does not manage item and resource assignments explicitly. These decisions are transferred to an auxiliary pricing problem that iteratively identifies feasible assignments given a constraint set and calculates the assignment costs accordingly. A hybrid method that combines a modified Hoffmann heuristic (Hoffmann 1963) with an estimation of distribution algorithm (EDA) (Pereira 2015) was also proposed. Computational experiments were conducted using a benchmark-based dataset derived from the literature (Otto, Otto, and Scholl 2013), comprising instances with up to 100 tasks. The case study from the manufacturer that originated the problem was also solved.

Thus far, we can observe that researchers had only discussed the RCALBP by essentially considering the minimisation of resources and stations (i.e. RCALBP-1). Nevertheless, Alakas, Pinarbasi, and Yüzükirmizi (2020) most recently studied the RCALBP-2. Although the cycle time was considered in the objective, it had a secondary role. Instead, the minimisation of the total number of used resources was aimed as the primary goal. However, researchers generally focus on minimising the line's cycle time in type-II ALBPs, a fact that was acknowledged by the authors. Furthermore, they claimed that fixing a limit of at most one resource per available station satisfied the condition of resource minimisation. A new CP model to deal with dedicated and alternative scenarios was proposed and a new dataset was generated to test their proposed model. The implicit assumption adopted by Alakas, Pinarbasi, and Yüzükirmizi (2020) here was the mandatory use of exactly one resource for each available station. Optimal solutions were obtained for all instances under this simplification hypothesis of allowing only one resource per station. However, this assumption is further evaluated and discussed throughout Sections 3, 4, and 5. We extend the problem by relaxing the enforcement on having exactly one resource per station, addressing why this might unnecessarily exclude feasible solutions.

Accepted manuscript

3. Problem statement

The characteristics of the RCALBP-2 to be considered are introduced using the example depicted in Figure 1. The notation is detailed in Table I.

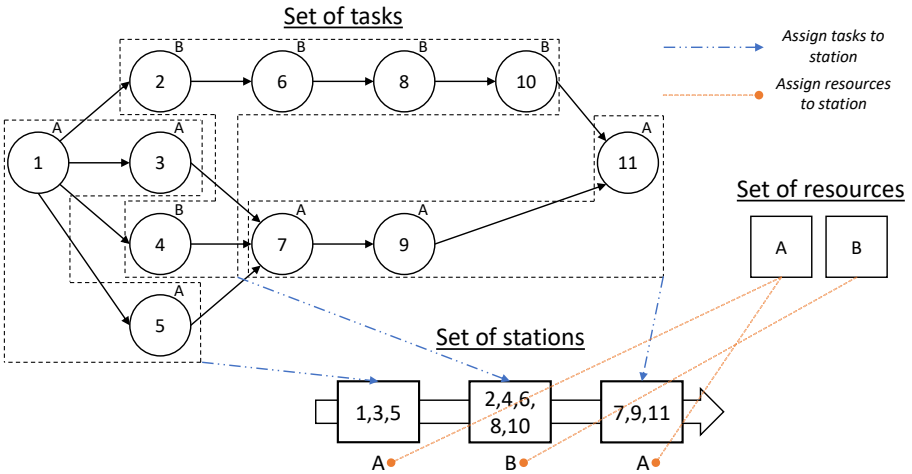


Figure 1. Illustrative example with a feasible solution for the RCALBP-2. Tasks are represented by nodes 1 to 11. Two resources, named A and B, are available. The resource requirement of each task is given on the upper right of the node: odd-numbered and even-numbered tasks require resources A and B, respectively. There are three available stations. Arrows and dotted lines show the assignment of subgroups of tasks and resources to stations, respectively.

A set of tasks (T) must be performed in order to assemble a product. These tasks must respect precedence relations (P) in their execution order. A precedence graph containing task indexes from 1 to 11 is presented in Figure 1. This precedence graph was also used by Agpak and Gökçen (2005) in their illustrative example. A set of resources (R) is also considered. In this illustrative example, two resource types (A and B) are considered and represented in the precedence graph for each task node. Resource A is necessary to perform odd indexed tasks and B for even indexed tasks. For this instance, there is a set of three available stations (S) to which tasks and resources must be assigned: $x_{ik} = 1$ and $y_{rk} = 1$ if task i and resource type r are assigned to station k, respectively. Tasks are indivisible and cannot be performed unless the required resource is present in the station. Finally, the cycle time (c) is determined by the sum of task processing times in the most loaded station.

Figure 1 also presents a feasible solution for the illustrative instance. Note that subsets of tasks with identical resource requirements are grouped together by dashed lines. Arrows assign these subgroups into stations: tasks 1, 3, and 5 to station 1, tasks 2, 4, 6, 8, and 10 to station 2, and tasks 7, 9, and 11 to station 3. Nonetheless, this is not enough to ensure feasibility; we must assign the appropriate resource types to stations as well. Dotted lines assign either resource type A or B to stations. Ultimately, we obtain a feasible solution in which three resource units were used (two of type A and one of type B).

The illustrated version of the problem assumes that resources are dedicated, i.e. each task only has one possibility of resource type to enable its execution. In Section 4, we model this basic case and extend our formulation to contemplate the alternative version of the problem, in which multiple-resource possibilities are considered. Furthermore, we eliminate the simplification hypothesis that imposes exactly one re-

Emerald Publishing Limited

source to be assigned to each available station, discussing why previous approaches may be eliminating conceivable optimal solutions.

4. MILP models

We develop the following problem formulation for the dedicated RCALBP-2 and its extensions using the notation introduced in Table I, in which definitions of indexes, sets, parameters, and variables are presented.

Table I. Notation: indexes, sets, parameters, and variables.

i, j	Task indexes
r	Resource index
k	Station index
T	Set of tasks
R	Set of resource types
S	Set of stations
P	Set of precedence relations $(i,j) \in T \mid i \prec j$
T_r	Set of tasks that can be performed with resource r
R_i	Set of resources required by task i
τ_i	Processing time of task i
α	Available number of resources
ρ_r	Cost of resource r
β	Available budget
γ	Number of resources allowed per station
x_{ik}	Binary variable equal to 1 if task i is assigned to station k
y_{rk}	Binary variable equal to 1 if resource r is assigned to station k
c	Continuous variable for cycle time

$$\text{Min } z_1 = c, \tag{1}$$

$$\text{Min } z_2 = \sum_{r \in R} \sum_{k \in S} y_{rk}, \tag{2}$$

subject to:

$$\sum_{k \in S} x_{ik} = 1, \quad \forall\, i \in T, \tag{3}$$

$$\sum_{k \in S} k \cdot x_{ik} \leq \sum_{k \in S} k \cdot x_{jk}, \quad \forall\, (i,j) \in P, \tag{4}$$

$$\sum_{i \in T} \tau_i \cdot x_{ik} \leq c, \quad \forall\, k \in S, \tag{5}$$

$$\sum_{i \in T_r} x_{ik} \leq |T_r| \cdot y_{rk}, \quad \forall\, r \in R;\ k \in S, \tag{6}$$

with

$$x_{ik} \in \{0,1\}, \quad \forall\, i \in T;\ k \in S, \tag{7}$$

$$y_{rk} \in \{0,1\}, \quad \forall\, r \in R;\ k \in S, \tag{8}$$

$$c \in \mathbb{R}_{\geq 0}. \tag{9}$$

Two goals are considered in this model by a lexicographical objective function. The primary one, given by Expression (1), is the minimisation of cycle time (c). The

Assembly Automation

minimisation of resources is aimed in Expression (2) as a secondary goal. Since this is a type-2 problem, the importance of c is much higher than the number of utilised resources. The literature for generalised type-2 ALBPs supports this approach, e.g. on lines with two-sided (Kim, Song, and Kim 2009) or multi-manned stations (Michels, Lopes, and Magatão 2020), with task setup times (Zohali, Naderi, and Roshanaei 2022), and resource-constrained lines (Alakas, Pinarbasi, and Yüzükirmizi 2020).

The model is subjected to Constraints (3)–(6). Task assignments are enforced by Constraints (3), ensuring that each task is assigned to exactly one station. Constraints (4) guarantee that the precedence relations are respected. A task i can only be assigned to a station if all its predecessors $j \mid i \prec j$ have been assigned to that station or earlier ones. Constraints (5) limit the total processing time of tasks for each station to be at most the cycle time (c). They also measure and identify the highest workload to be minimised. Lastly, Constraints (6) indicate that a task i can only be assigned to a station k if the required resource type r to perform task i is also assigned to the referred station k. For these last constraints, a bound on the number of tasks that can be executed in the station is needed. The cardinality of the subset of tasks that can be performed with resource r (T_r) is clearly a valid bound.

4.1. *Resource availability constraints*

An important generalisation proposed by Corominas, Ferrer, and Pastor (2011) was to limit the number of resources to an available quantity. This condition has not been extended to the RCALBP-2. Constraint (10) bounds the total number of resources to be used in the line to α. Naturally, this can be readily extended to consider a limit on each type of resource. Additionally, Constraints (11) are valid inequalities to avoid gaps in station numbering and thus reducing the symmetry of the model.

$$\sum_{r \in R} \sum_{k \in S} y_{rk} \leq \alpha, \tag{10}$$

$$\sum_{r \in R} y_{rk} \leq \sum_{r \in R} |R| \cdot y_{rk-1}, \quad \forall\, k \in S \mid k > 1. \tag{11}$$

4.2. *Multiple-resource possibilities*

As mentioned by Agpak and Gökçen (2005), Constraints (6) fail to assess scenarios in which alternative resources are considered. Common tasks that can be performed by different resources must be represented by a separate set of alternative resources required by task i (R_i). Constraints (12) may entirely replace the previous ones (6) when multiple-resource possibilities exist. This feature has not previously been considered in an MILP model for the RCALBP-2 extension.

$$x_{ik} \leq \sum_{r \in R_i} y_{rk}, \quad \forall\, i \in T;\ k \in S. \tag{12}$$

Assembly Automation

4.3. *Simplification hypothesis*

An implicit simplification hypothesis adopted by Alakas, Pinarbasi, and Yüzükirmizi (2020) in their CP model was the mandatory use of exactly one resource for each available station. They state that by assigning only one resource at each station, the minimum number of resources is guaranteed and their CP model can focus on achieving the objective of minimising cycle time. In an MILP formulation, such assumption can be integrated by appending either Constraints (13) or Constraints (14) with $\gamma = 1$. The former are equivalent to the ones employed in Alakas, Pinarbasi, and Yüzükirmizi (2020), but the latter are more flexible. While Constraints (13) explicitly prohibit the assignment of tasks that require different resources ($i, j \in T \mid R_i \cap R_j = \emptyset$) to the same station, one can easily modify the constant on the right-hand side of Constraints (14) to reflect a more realistic limitation on the number of resources that are allowed in each station. More interestingly, Constraints (13) and (14) can be combined to define incompatible resources as part of a RCALBP-2 extension. Constraints (13) may prohibit specific resource pairs from going into the same station, while Constraints (14) restrict the maximum number of resources allowed in the same station.

$$x_{ik} + x_{jk} \leq 1, \quad \forall\, k \in S;\ i, j \in T \mid R_i \cap R_j = \emptyset, \tag{13}$$

$$\sum_{r \in R} y_{rk} \leq \gamma, \quad \forall\, k \in S. \tag{14}$$

4.4. *Budget-oriented model*

One last useful modification in the proposed MILP models is taking into account individualised costs for each type of resource (ρ_r). Constraints (15) limit expenditure on used resources to a given budget amount (β) instead of the total quantity (Constraints (10)). Alternatively, a model that prioritises total cost (stations and resources) minimisation could also be proposed as a cost-oriented RCALBP-2 extension.

$$\sum_{r \in R} \sum_{k \in S} \rho_r \cdot y_{rk} \leq \beta. \tag{15}$$

4.5. *Motivational example*

The first MILP model herein proposed is composed of Expressions (1)–(9) and (10)–(13). Since resource assignment to each station is limited to one – but not mandated to be one –, this model is named Restricted Model (RM). The second MILP model is hereafter called Flexible Model (FM): it excludes Constraints (13) from the RM. Therefore, FM does not impose a limitation on the number of resources that can be used in each station. Notice that both RM and FM are able to cope with alternative multiple-resource possibilities.

Even though the RM permits only one resource per station, it does not oblige each available station to take a resource as in the Alakas, Pinarbasi, and Yüzükirmizi (2020) CP model (hereafter called AM) used for comparison. With both proposed models defined, this advantage can be demonstrated in a sample scenario provided by Alakas, Pinarbasi, and Yüzükirmizi (2020). In this comparative instance, there are three types of resources ($|R| = 3$) and some tasks can be performed by two or

Assembly Automation

more resources (alternative). The number of available stations is taken as $|S| = 8$. Consequently, the number of available resources is also $\alpha = 8$ for the comparison. The precedence graph with 11 tasks ($|T| = 11$) from Figure 1 is considered. Task processing times and resource requirements for each task are given in Table II.

Table II. Task processing times and resource requirements for the comparative instance.

Task index	1	2	3	4	5	6	7	8	9	10	11
Task time	6	2	5	7	1	2	3	6	5	5	4
Resources	A	A,B	A,C	B	A,B,C	B	A,C	A,B,C	A	B	A

Detailed solutions for the previous AM and the proposed models (RM and FM) are given in Table III. Columns show which tasks and resources were assigned to each station. The total number of used resources (#R) and the optimised cycle time (c) for each model are also presented. As expected, AM uses 8 resources by compulsorily assigning one in each available station. With that configuration, AM yields a cycle time of 8 time units. The proposed RM and FM, on the other hand, are able to obtain the same cycle time ($c = 8$) with fewer resources (#R = 7). This advantage is made possible by allowing unused stations and directly considering the use of resources, instead of simply fixing exactly one resource per station. Indeed, both FM and RM found equivalent solutions in terms of objective function value for this case, but FM may yield better solutions for larger instances with more resource alternatives, as further shown in the results from Section 5.

Table III. Detailed solutions for the comparative instance.

	AM		RM		FM	
Station	Task(s)	Resource	Task(s)	Resource	Task(s)	Resource
1	1,5	A	1,2	A	1,2	A
2	4	B	4,5	B	6,8	B
3	3,7	A∨C	3,7	A∨C	10	B
4	2,9	A	9	A	4,5	B
5	6	B	6,8	B	3,7	A∨C
6	8	A∨B∨C	10	B	9	A
7	10	B	11	A	11	A
8	11	A	–	–	–	–
	#R	8	#R	**7**	#R	**7**
	c	8	c	8	c	8

5. Computational results

This section carries out two computational studies: the first one to compare the proposed models to the previously developed CP model (Section 5.1) and a second one on a newly proposed dataset (Section 5.2). All instances were solved on Gurobi 8.1 (Gurobi Optimization 2019), which was employed as universal solver. Four threads of a 64-bit Intel Core™ i5-7200U CPU (2.5 GHz) with 8GB of RAM were used to run the models. The models and their interaction with input and output data files were coded in Microsoft Visual Basic 2019 programming language. The time limit is set to 3600 seconds for all cases.

Assembly Automation

5.1. *Available dataset*

There is only one dataset available in the literature for the studied RCALBP-2 (Alakas, Pinarbasi, and Yüzükirmizi 2020). This dataset samples five classical problem instances (Mitchell, Heskiaoff, Sawyer, Kilbridge, and Tonge) from Hoffmann (1990), totalling 40 instances consisting of 21 to 70 tasks ($|T|$) from a well-known literature benchmark, which is available for download at <www.assembly-line-balancing.de>. Scenarios with dedicated and alternative resource requirements are included, namely ii and iii, respectively. For each scenario, instances with 4 and 5 resource types ($|R|$) are designed. The total number of available stations ($|S|$) and resources (α) are fixed to 15, 18, 20, 25, 30, or 35, depending on the instance. The computational experiments were performed by comparing each model's results in terms of cycle time (c) and number of used resources along the line (#R). Lastly, we noticed that some precedence arcs were missing in the instances made available by Alakas, Pinarbasi, and Yüzükirmizi (2020) when compared to the literature benchmark. For fair comparison purposes, we decided not to include the missing arcs to test our models (RM and FM) against the AM. Instead, we make the instances and detailed results available in the Supporting Information, highlighting the missing arcs where appropriate. The summarised results for each instance are given in Table IV.

Since AM and RM limit the number of resources to one in each station, the most straightforward comparison is performed between them. RM reached optimality for the entire dataset. Out of 40 instances, RM obtained 3 solutions with improved cycle times and the same number of resources, 27 solutions with fewer used resources and the same cycle time, and 10 equivalent solutions when compared to AM. The instances in which RM obtained better cycle times than AM are proven feasible and the feasibility check is made available in the Supporting Information. Ultimately, having unnecessary stations operating with costly resources along the assembly line is not a desired feature. This experiment demonstrates that mandatorily fixing one resource in each available station may lead models to obtain suboptimal solutions.

The second half of the experiment is designed to empirically test theoretical advantages of allowing more than one resource to be assigned to the same station. Under the computational processing time limit, the FM was able to yield optimal solutions for 28 instances and feasible integer solutions for the remaining 12 cases. Compared to the RM, the FM obtained 3 solutions with improved cycle times and the same number of resources, 11 solutions with fewer used resources and the same cycle time, and 26 equivalent solutions. Table V exemplifies a solution with improved cycle times and the same number of resources. By taking the Mitchell scenario ii $|R| = 4$ instance as an example, Table V shows how the cycle time can be improved with the same number of resources, as the flexibility of admitting more than one resource per station allows a range of new assignment possibilities.

Finally, adding the flexibility to allow more than one resource per station is perfectly reasonable in the context of assembly lines, particularly when large-size products (such as cars, buses, and trucks) are manufactured. In any case, the maximum number of resources per station can readily be adjusted for each type of resource by modifying Constraints (14). This experiment empirically demonstrated that improved solutions can be obtained when the flexibility of assigning more than one resource per station is incorporated into the model.

For Peer Review Only

Table IV. Result comparison for the dataset presented in Alakas, Pinarbasi, and Yüzükirmizi (2020).

Problem	Scenario	$\|R\|$	$\|S\| = \alpha$	AM		RM		FM	
				c	#R	c	#R	c	#R
Mitchell	ii	4	15	17	15	17	15	**13**	**15**
			18	13	18	13	16	**13**	**15**
		5	15	13	15	13	15	**13**	**14**
			18	13	18	13	15	**13**	**14**
	iii	4	15	13	15	**13**	**10**	**13**	**10**
			18	13	18	**13**	**10**	**13**	**10**
		5	15	13	15	**13**	**11**	**13**	**11**
			18	13	18	**13**	**11**	**13**	**11**
Heskiaoff	ii	4	15	108	15	**108**	**13**	**108**	**13**
			18	108	18	**108**	**13**	**108**	**13**
		5	15	108	15	**108**	**13**	**108**	**13**
			18	108	18	**108**	**13**	**108**	**13**
	iii	4	15	108	15	**108**	**11**	**108**	**11**
			18	108	18	**108**	**11**	**108**	**11**
		5	15	108	15	**108**	**12**	**108**	**12**
			18	108	18	**108**	**12**	**108**	**12**
Sawyer	ii	4	15	34	15	34	15	34	15
			18	25	18	25	18	25	18
		5	15	32	15	32	15	32	15
			18	26	18	26	18	26	18
	iii	4	15	26	15	**25***	**15**	**25**	**15**
			18	25	18	**25**	**15**	**25**	**15**
		5	15	35	15	**26***	**15**	**26**	**15**
			18	25	18	**25**	**16**	**25**	**16**
Kilbridge	ii	4	20	55	20	55	20	**55**^{I}	**16**
			25	55	25	55	20	**55**^{I}	**16**
		5	20	55	20	55	20	**55**^{I}	**19**
			25	55	25	55	20	**55**^{I}	**19**
	iii	4	20	55	20	55	15	**55**	**13**
			25	55	25	55	15	**55**	**13**
		5	20	55	20	55	15	**55**	**14**
			25	55	25	55	15	**55**	**14**
Tonge	ii	4	25	231	25	231	25	**205**^{I}	**25**
			30	156	30	**156**	**29**	**156**^{I}	**29**
		5	30	197	30	197	30	**180**^{I}	**30**
			35	156	35	**156**	**33**	**156**^{I}	**33**
	iii	4	25	187	25	**159***	**25**	**159**^{I}	**25**
			30	156	30	**156**	**26**	**156**^{I}	**26**
		5	30	156	30	**156**	**27**	**156**^{I}	**27**
			35	156	35	**156**	**27**	**156**^{I}	**27**

*New optimal solutions.
^{I} Integer feasible solutions.

5.2. *Proposed dataset*

An extended dataset for the RCALBP-2 is herein proposed to test the FM. We consider that the previous dataset mentioned in Section 5.1 did not have the most appropriate and challenging instances for the studied problem. The reason for that is mainly related to the excessive number of stations and, consequently, resources. Given the limitation of one resource per station, the large number of available instances would always guarantee the existence of feasible solutions. In fact, for the simpler SALBP-2 version, the number of available stations was large enough to the point that cycle times were actually bounded only by the largest task processing time.

However, with the flexibility implemented on the FM, infeasible solutions can be

Assembly Automation

Table V. Detailed solutions for the Mitchell scenario ii $|R| = 4$ instance.

	AM		RM		FM	
Station	Task(s)	Resource	Task(s)	Resource	Task(s)	Resource
1	1	A	1	A	1,3	A∧C
2	3	C	3	C	4	D
3	4,12	D	4,12	D	5	A
4	5	A	5	A	7	C
5	2,6	B	2,6	B	2,6,14	B
6	7	C	7	C	8,12	D
7	8	D	8	D	9,11,13	A∧C
8	9,13,21	A	9,13,21	A	10,15,18,19	B∧C
9	10	B	10	B	21	A
10	11,15	C	11,15	C	16	D
11	16	D	16	D	17	A
12	17	A	17	A	20	D
13	14,18	B	14,18	B	–	–
14	19	C	19	C	–	–
15	20	D	20	D	–	–
	#R	15	#R	15	#R	15
	c	17	c	17	c	**13**

easily avoided even if we only have a few available stations. This is due to the fact that, in the worst case scenario, all different resource types could be assigned to the same station along with all tasks, which implies in a cycle time as high as the sum of all task processing times.

The proposed dataset is presented in Table VI and instances are made available as Supporting Information. The same precedence graphs considered before are used, with $|T|$ varying from 21 to 70 tasks. Nevertheless, corrections have been made by including the missing arcs from the original benchmark instances. Dedicated (ii) and alternative (iii) scenarios with 4 or 5 resource types ($|R|$) are once again taken into account. The number of available stations ($|S|$) is much lower, varying from 8 to 12, depending on the problems' size. The last variation is explicitly evaluating the effects of having different numbers of available resources (α), ranging from 8 to 15 units. These modifications add 40 new instances to the RCALBP-2 dataset.

Table VI. Proposed dataset extension.

Problem	Tasks ($\|T\|$)	Scenarios	Resource types ($\|R\|$)	Available stations ($\|S\|$)	Available resources (α)
Mitchell	21	ii,iii	4,5	8	8,10
Heskiaoff	28	ii,iii	4,5	8	8,10
Sawyer	30	ii,iii	4,5	10	10,12
Kilbridge	45	ii,iii	4,5	10	10,12
Tonge	70	ii,iii	4,5	12	12,15

Table VII presents the results obtained by the FM when it is applied to the proposed dataset (Table VI). The comparison here is made between the minimised cycle time obtained by the FM (c) and the optimal SALBP-2 cycle time (c_s). As the number of available resources is scarce, the model always uses all of them. In Table VII, Ratio stands for the numerical value obtained by the fraction $\frac{c}{c_s}$ and CPU(s) for the computational processing time, given in seconds. Once again, detailed result files are provided in the Supporting Information.

From Table VII, it can be observed that the cycle times obtained by the resource-constrained version of the problem are always higher than when this limitation is not considered, which is an expected outcome. Nonetheless, there are many interest-

Assembly Automation

ing aspects to analyse in order to extract managerial insights. All other parameters equal, the optimal cycle time value quickly converges to the SALBP-2 lower bound as the available number of resources increases, which shows the possibility of a trade-off between cycle time and the number of used resources for practical situations in the decision making process. As long as both the cycle time and used resources are reduced to monetary units, this trade-off can be promptly evaluated by the assembly line's manager by using the budget-constrained extension proposed in Section 4. Furthermore, whilst alternative resources boost assignment possibilities, the increment of resource type requirements contributes to stricter choices. Lastly, the FM was able to achieve and prove optimal solutions for all proposed instances.

Table VII. Experimental results for the proposed dataset extension.

Problem	Scenario	$\|R\|$	$\|S\|$	α	c	c_s	Ratio	CPU(s)
Mitchell	ii	4	8	8	53		3.79	0.53
				10	35		2.50	1.03
		5	8	8	76		5.43	0.47
				10	51		3.64	1.00
	iii	4	8	8	18	14	1.29	0.53
				10	15		1.07	0.18
		5	8	8	31		2.21	2.25
				10	15		1.07	0.24
Heskiaoff	ii	4	8	8	274		2.12	2.64
				10	153		1.19	1.42
		5	8	8	513		3.98	1.11
				10	250		1.94	7.41
	iii	4	8	8	139	129	1.08	4.15
				10	135		1.05	4.77
		5	8	8	175		1.36	1.77
				10	139		1.08	2.82
Sawyer	ii	4	10	10	108		3.18	14.69
				12	59		1.74	27.61
		5	10	10	133		3.91	8.48
				12	95		2.79	31.21
	iii	4	10	10	37	34	1.09	7.07
				12	35		1.03	8.49
		5	10	10	41		1.21	15.80
				12	35		1.03	6.95
Kilbridge	ii	4	10	10	163		2.91	28.73
				12	110		1.96	157.20
		5	10	10	266		4.75	12.45
				12	161		2.88	23.28
	iii	4	10	10	78	56	1.39	132.75
				12	59		1.05	118.98
		5	10	10	94		1.68	63.72
				12	64		1.14	145.80
Tonge	ii	4	12	12	1130		3.84	270.78
				15	835		2.84	1375.30
		5	12	12	1434		4.88	628.12
				15	1145		3.89	1474.63
	iii	4	12	12	860	294	2.93	661.35
				15	412		1.40	2049.35
		5	12	12	927		3.15	428.08
				15	481		1.64	2608.09

Assembly Automation

6. Conclusion

The Resource-Constrained Assembly Line Balancing Problem (RCALBP) with the objective of minimising the cycle time under a limited number of stations and resources has been addressed in this paper. The existing literature on both types I and II of the problem has been reviewed, focusing on mathematical models and including all practical variants. The main contribution herein presented is the extension of a recently proposed problem (Alakas, Pinarbasi, and Yüzükirmizi 2020). Instead of restricting the assembly line to compulsorily assign exactly one resource to all available stations, we propose a more holistic approach, in which both task and resource assignments are decisions to be taken, avoiding unnecessary use of stations and resources. This situation is modelled and solved, including the possibility of allocating only one or multiple resources into the same station.

In order to solve the proposed optimisation problem, two new Mixed-Integer Linear Programming (MILP) models are developed and named Restricted Model (RM) and Flexible Model (FM). These models ensure that all tasks are performed by the correct resources while minimising cycle time and the total number of resources used along the line. The RM allows at most one resource to be assigned to each station, whereas the FM is free to have one resource of each type per station, as long as the total number of available resources is respected. A cost-oriented extension to directly evaluate a trade-off between productivity rate and used resources is also outlined. The results obtained from a literature dataset indicate that forcing each available station to take one resource may lead to suboptimal solutions. Thus, new optimal solutions were obtained by explicitly taking into account resource assignment decisions. A new dataset is proposed to represent shorter and more realistic lines. The FM was able to produce optimal solutions to all new instances. In general, both models were shown to be effective and flexible when dealing with assembly line balancing problems with resource limitations, providing optimal solutions and applicable results for manufacturing industries, as well as valuable managerial insights on the financial trade-off between cycle time and resource usage.

Following the line of research in the RCALBP-1, the MILP models herein proposed can be expanded to deal with different line layouts (U-line, parallel stations) and product variety (multi- and mixed-model lines). Additionally, further research may also focus on cost-oriented formulations and (meta)heuristic algorithms can be developed for very large-size problems.

Assembly Automation

Acknowledgements

This research was partially funded by the Australian Government through the Australian Research Council Industrial Transformation Training Centre in Optimisation Technologies, Integrated Methodologies, and Applications (OPTIMA), Project ID IC200100009.

References

Agpak, Kürsad, and Hadi Gökçen. 2005. "Assembly line balancing: Two resource constrained cases." *International Journal of Production Economics* 96 (1): 129–140.

Alakas, Haci Mehmet, Mehmet Pinarbasi, and Mustafa Yüzükirmizi. 2020. "Constraint pro-

gramming model for resource-constrained assembly line balancing problem." *Soft Computing* 24 (7): 5367–5375.
Askin, R G, and M Zhou. 1997. "A parallel station heuristic for the mixed-model production line balancing problem." *International Journal of Production Research* 35 (11): 3095–3106.
Bartholdi, J. J. 1993. "Balancing two-sided assembly lines: A case study." *International Journal of Production Research* 31 (10): 2447–2461.
Baybars, Ilker. 1986. "A survey of exact algorithms for the simple assembly line balancing problem." *Management Science* 32 (8): 909–932.
Becker, Christian, and Armin Scholl. 2006. "A survey on problems and methods in generalized assembly line balancing." *European Journal of Operational Research* 168 (3): 694–715.
Boysen, Nils, Malte Fliedner, and Armin Scholl. 2007. "A classification of assembly line balancing problems." *European Journal of Operational Research* 183 (2): 674–693.
Boysen, Nils, Philipp Schulze, and Armin Scholl. 2021. "Assembly line balancing: What happened in the last fifteen years?" *European Journal of Operational Research* .
Chen, Yin Yann, Chen Yang Cheng, and Jia Ying Li. 2018. "Resource-constrained assembly line balancing problems with multi-manned workstations." *Journal of Manufacturing Systems* 48 (April): 107–119.
Corominas, Albert, Laia Ferrer, and Rafael Pastor. 2011. "Assembly line balancing: General resource-constrained case." *International Journal of Production Research* 49 (12): 3527–3542.
Ege, Yunus, Meral Azizoglu, and Nur E Ozdemirel. 2009. "Computers & Industrial Engineering Assembly line balancing with station paralleling." *Computers & Industrial Engineering* 57 (4): 1218–1225.
Friedman, Arthur D. 1985. *Fundamentals of Logic Design and Switching Theory*. USA: Computer Science Press, Inc.
Gurobi Optimization. 2019. "Gurobi Optimizer reference manual." .
Hoffmann, Thomas R. 1963. "Assembly line balancing with a precedence matrix." *Management Science* 9 (4): 551–562.
Hoffmann, Thomas R. 1990. "Assembly line balancing: a set of challenging problems." *International Journal of Production Research* 28 (10): 1807–1815.
Kamarudin, N H, and M F F Ab Rashid. 2017. "Assembly line balancing with resource constraints using new rank-based crossovers." *Journal of Physics: Conference Series* 908 (1).
Kao, Hsiu-hsueh, Din-Horng Yeh, and Yi-Hsien Wang. 2011. "Resource Constrained Assembly Line Balancing Problem Solved with Ranked Positional Weight Rule." *Review of Economics & Finance* (1986): 71–80.
Kao, Hsiu-hsueh, Din-horng Yeh, Yi-hsien Wang, and Jui-cheng Hung. 2010. "An optimal algorithm for type-I assembly line balancing problem with resource constraint." *African Journal of Business Management* 4 (10): 2051–2058.
Kim, Yeo Keun, Won Seop Song, and Jun Hyuk Kim. 2009. "A mathematical model and a genetic algorithm for two-sided assembly line balancing." *Computers & Operations Research* 36 (3): 853–865.
Michels, Adalberto Sato, Thiago Cantos Lopes, and Leandro Magatão. 2020. "An exact method with decomposition techniques and combinatorial Benders' cuts for the type-2 multi-manned assembly line balancing problem." *Operations Research Perspectives* 7: 100163.
Michels, Adalberto Sato, Thiago Cantos Lopes, Celso Gustavo Stall Sikora, and Leandro Magatão. 2018. "The Robotic Assembly Line Design (RALD) problem: Model and case studies with practical extensions." *Computers & Industrial Engineering* 120: 320–333.
Michels, Adalberto Sato, Thiago Cantos Lopes, Celso Gustavo Stall Sikora, and Leandro Magatão. 2019. "A Benders' decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem." *European Journal of Operational Research* 278 (3): 796–808.
Ogan, Dilek, and Meral Azizoglu. 2015. "A branch and bound method for the line balancing problem in U-shaped assembly lines with equipment requirements." *Journal of Manufacturing Systems* 36: 46–54.

Assembly Automation

Otto, Alena, Christian Otto, and Armin Scholl. 2013. "Systematic data generation and test design for solution algorithms on the example of SALBPGen for assembly line balancing." *European Journal of Operational Research* 228 (1): 33–45.

Pereira, Jordi. 2015. "Empirical evaluation of lower bounding methods for the simple assembly line balancing problem." *International Journal of Production Research* 53 (11): 3327–3340.

Pereira, Jordi. 2018. "Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem." *International Journal of Production Research* 56 (11): 3994–4016.

Pinarbasi, Mehmet, Hacı Mehmet Alakas, and Mustafa Yuzukirmizi. 2019. "A constraint programming approach to type-2 assembly line balancing problem with assignment restrictions." *Assembly Automation* 39 (5): 813–826.

Quyen, Nguyen Thi Phuong, James C Chen, and Chao Lung Yang. 2017. "Hybrid genetic algorithm to solve resource constrained assembly line balancing problem in footwear manufacturing." *Soft Computing* 21 (21): 6279–6295.

Quyen, Nguyen Thi Phuong, R J Kuo, James C Chen, and Chao Lung Yang. 2017. "Dynamic programming to solve resource constrained assembly line balancing problem in footwear manufacturing." *2017 4th International Conference on Industrial Engineering and Applications, ICIEA 2017* 66–70.

Rahman, Humyun Fuad, Mukund Nilakantan Janardhanan, and Peter Nielsen. 2020. "An integrated approach for line balancing and AGV scheduling towards smart assembly systems." *Assembly Automation* 40 (2): 219–234.

Scholl, Armin, and Christian Becker. 2006. "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing." *European Journal of Operational Research* 168 (3): 666–693.

Yadav, Ashish, Ramawatar Kulhary, Rupesh Nishad, and Sunil Agrawal. 2020. "Parallel two-sided assembly line balancing with tools and tasks sharing." *Assembly Automation* 40 (6): 833–846.

Yang, Mingshun, Li Ba, Erbao Xu, Yan Li, Yong Liu, and Xinqin Gao. 2020. "Optimization of a multi-constraint two-sided assembly line balancing problem using an improved imperialist competitive algorithm." *Assembly Automation* 40 (2): 273–282.

Yilmaz, Hamid, and Mustafa Yilmaz. 2017. "Multi-manned assembly line balancing problem: Resource constrained case." *Proceedings of the International Conference on Industrial Engineering and Operations Management* 1: 752–759.

Zohali, Hassan, Bahman Naderi, and Vahid Roshanaei. 2022. "Solving the Type-2 Assembly Line Balancing with Setups Using Logic-Based Benders Decomposition." *INFORMS Journal on Computing* 34 (1): 315–332.

Assembly Automation