Assembly Line Balancing for two Cycle Times: Anticipating Demand Fluctuations

Thiago Cantos Lopes[a], Adalberto Sato Michels[b], Celso Gustavo Stall Sikora[c], Nadia Brauner[d], Leandro Magatão[a,*]

[a] *Graduate Program in Electrical and Computer Engineering, Federal University of Technology - Paraná, Brazil*
[b] *School of Mathematics and Statistics, University of Melbourne, Melbourne, Australia*
[c] *Institute for Operations Research, Hamburg Business School, University of Hamburg, Germany*
[d] *Université Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, F-38000 Grenoble, France*

Abstract

Unpredictable crises such as pandemics, as well as predictable oscillations such as seasonality, can produce significant demand fluctuations. Although it is possible to adapt the manufacturing system to these perturbations, there are significant opportunities in anticipating them in the design stage. This paper proposes the Economically Robust Assembly Line Balancing Problem (ERALBP), which addresses the issue by designing assembly lines to allow flexible alternation between two or more cycle times. A Mixed-Integer Linear Programming (MILP) model is introduced to describe the problem. Moreover, a heuristic procedure is implemented in order to quickly produce high-quality solutions. While the model failed to find solutions for most medium and large instances, the heuristic quickly produced high-quality solutions, reaching low solution gaps even for large instances. Finally, a case study with industrial data further highlights the advantages of the proposed strategy: by anticipating demand fluctuations, the proposed heuristic's solution facilitates alternation between two demand scenarios, both with the optimal number of stations. This approach is less costly than the re-balancing alternative, which requires re-assigning and re-positioning tasks. By enabling companies to perform this fast switching between output rates, we allow them to benefit from economic opportunities tied to increased seasonal effects or unexpected demand spikes.

Keywords: Flexible Manufacturing Systems, Assembly Line Balancing, Demand Fluctuation, Heuristic, Case Study

Computers & Industrial Engineering, Volume 162, December 2021, 107685
DOI: 10.1016/j.cie.2021.107685

*Corresponding author: magatao@utfpr.edu.br

1. Introduction

Reliable information is a default assumption of Operations Research (OR). It means that most OR-related problems are addressed in stable and generally predictable conditions. However, internal and external uncertainties are often unavoidable. The existing stochastic programming (Birge & Louveaux, 2011) and robust optimization (Ben-Tal et al., 2009) literature attempt to deal with these inescapable unreliabilities. The recent COVID-19 outbreak serves as an example (Hui et al., 2020). A new virus had been identified in late 2019 and a few months later, in early 2020, it became a global pandemic. This situation led most countries to adopt strict social and economic measures to prevent infections from spreading.

As a consequence, these imposed restrictions naturally impacted the demand for many manufactured goods. Depending on the industry, their products recently experienced sharp drops or spikes in demand due to the pandemic itself, as well as resulting government actions (e.g. border closures, quarantines) and customer decisions (e.g. panic buying). The ability to adequately respond to such perturbations and mitigate their impact has an immediate managerial and societal interest. It also motivates research on the development of more flexible manufacturing systems that can switch from "lockdown" to "recovery" outputs. In particular, this paper investigates how this concept can be applied to assembly lines through economically robust balancing strategies, giving rise to the proposed Economically Robust Assembly Line Balancing Problem (ERALBP).

Recently, many businesses have chosen or been forced to change their production rates. When an abrupt output reduction is deemed necessary (e.g. automotive industries), a common approach is to implement cutbacks on the number or duration of shifts. However, if an output increase is desired (e.g. healthcare products), these approaches might be more limited – especially if the factory is already operating at or near its maximum weekly hours. In those cases, increasing the production rate is necessary to achieve the desired higher output. Nonetheless, due to differences in scale and nature, not all businesses can easily incorporate changes in their production rates. Still, there are relevant economic opportunities in switching from "lockdown" to "recovery" outputs for both types in specific contexts. For industries that can handle these alternations, this means a better response to seasonal effects or sudden demand spikes. For those that cannot, this means designing a manufacturing unit planned for lower initial throughput, while also anticipating a probable (or even intended) higher capacity for the future – this may translate into a more efficient transition later on. Thus, companies embracing this strategy might be capable of preventing some negative impacts caused by an eventual similar disruption.

This paper considers a single-model assembly line balancing problem and investigates how these manufacturing systems can be efficiently designed for two (or more) target throughput rates. This economically robustness entails a comprehensive plan to position task execution and machinery along the line in a manner that allows efficient line partitions under optimized number of stations for each cycle time. This allows the operation to occur under two (or more) alternating target production rate without disturbing the settled configuration's order. The remainder of this paper is organized as follows. Section 2 extensively reviews the relevant literature, discusses its shortcomings regarding the present question, and justifies this work. Section 3 defines the studied problem and presents a Mixed-Integer Linear Programming (MILP) model to represent the proposed flexibility. Section 4 introduces a heuristic method to achieve good solutions for the proposed problem. The MILP model and the heuristic method are both applied to well-known benchmark datasets and their computational experiments' results are reported in Section 5. Furthermore, Section 6 presents an industrial case study, which applies this paper's techniques to practical data of a gearbox assembly line. Finally, key conclusions are summarized in Section 7.

2. Related Works

Assembly line balancing is a classical optimization problem tied to their specific manufacturing context (Scholl, 1999). Its simplest version, the Simple Assembly Line Balancing Problem (SALBP), was first formalized by Baybars (1986) and consists in assigning tasks with deterministic durations to (work)stations, subject to precedence relations between tasks and cycle time requirements for each station. Line balancing problems are usually NP-hard as they subsume bin packing as a particular case (Álvarez-Miranda & Pereira, 2019). Previous works have incorporated relevant practical considerations to line balancing problems, such as ergonomic risks (Bortolini et al., 2017), ecological considerations (Liu et al., 2020), space constraints (Zhang et al., 2020), workstation planning (Defersha & Mohebalizadehgashti, 2018), worker variability (Öner-közen et al., 2017), and internal storage (Lopes et al., 2021).

Most of the relevant literature on robust or stochastic assembly line balancing focus on task durations in terms of parameter uncertainty. A classification of assembly line balancing problem in the literature, proposed by Boysen et al. (2008), considers only two elements related to uncertainty: the processing times can be stochastic or the cycle time restriction must be valid for a given probability. The former is usually used as a restriction for uncertain processing times, taking the form of a chance constraint (for instance, the cycle time restriction must be valid in 95% of the possible realizations (Kao, 1976)). More recent general reviews on assembly line balancing (Battaïa & Dolgui, 2013; Eghtesadifard et al., 2020) notice an increase in the number of contributions dealing with uncertainty. However, the classification of Battaïa & Dolgui (2013) only extends the uncertain processing times classification by considering the subcategories of stochastic, fuzzy, and interval processing times. Even specific surveys on stochastic (Bentaha et al., 2015) and robust (Hazir et al., 2019) approaches on assembly line balancing limit their scope to uncertainty in the processing times.

Nevertheless, there are other sources of uncertainty in the industrial production that are less explored in the literature. This manuscript focuses on product demand as uncertainty drive and how to hedge against this variation. The demand for products naturally changes as they are introduced to the market, mature, and are discontinued. The demand levels are also affected by marketing efforts. They are prone to market changes such as economic crises or global pandemics (Ivanov & Dolgui, 2020).

There are two observable trends in the literature related to demand changes. The first one considers the problem of re-optimizing an existing system for another production situation. In contrast, the second trend approaches the design of the assembly line foreseeing changes in the demand, so that the system is either robust or adapt well to variation. The former approach is called re-balancing (Battaïa & Dolgui, 2013) and considers the invested capital on an assembly line's stations and equipment. Moving heavy machinery and retraining workers impose costs, so that re-balancing approaches usually restrict the amount or costs of the reallocations. Examples are Gamberini et al. (2006), Makssoud et al. (2015), and Sikora et al. (2017). They respectively consider the maximization of a similarity index among multiple objectives, the minimization of the number of changes, or costly reassignments as fixed.

The literature that prepares the design of an assembly system coping with demand variation is said to be flexible (Simaria et al., 2009), robust (Chica et al., 2016), or prepared for a demand variation environment (Li & Gao, 2014). The works from Simaria et al. (2009), Li & Gao (2014), Yang & Gao (2016), Chica et al. (2016), Chica et al. (2019), and Sikora (2021) all consider the balancing problem of an assembly line considering that demand can be modeled as a finite set of demand scenarios.

Simaria et al. (2009) consider the design of an assembly line that must operate well with different demand levels. The authors deal with a multi-product assembly line, however, only one product is produced per scenario. The multiple scenarios determine which and how much must be produced. The proposed solution procedure is based on a hierarchical approach: first, the line is balanced for the most loaded demand scenario; in the second level, workers are assigned to tasks and stations. In the first problem, the assignments of tasks to stations are fixed, representing the installation of equipment and machines. At the second level, tasks are assigned to workers, who can perform tasks in multiple stations in a U-shaped assembly line. For each demand of each scenario, the number of workers used is minimized. The procedure relies on a hierarchical approach in which the assignments found in the first stage are fixed for the second stage. There is no iteration between the stages, and both steps are solved with metaheuristics.

Li & Gao (2014) use overtime as a measure to deal with demand variability. Their system consists of a mixed-model assembly line producing under a collection of demand scenarios. The authors minimize the system's running costs composed of regular (linked to the number of stations) and overtime costs. The assignment of tasks to stations represents the allocation of equipment, which must be equal for all demand scenarios, along with the number of stations and workers. For each demand, however, the cycle time and the amount of overtime can be specified.

The consideration of local reassignments of tasks is modeled in Yang & Gao (2016). The authors consider the training of workers as the bottleneck for reassigning tasks in the assembly line. The approach is based on cross-training, that is, workers are trained for the tasks in their station plus the tasks of an adjacent station. The task allocation is then assigned to skill zones instead of directly to stations. In each demand scenario, tasks within each skill zone can be reassigned to workers with a specific ability. The method is used to minimize the number of skill zones, which translates to minimizing workstations.

Chica et al. (2016) and Chica et al. (2019) focus on the robustness of the task allocations with respect to processing time and area restrictions. The modeled assembly line works with multiple products, for which one assignment of tasks to stations is selected. In Chica et al. (2016), a finite set of scenarios contains possible demand values for each product. The problem is solved in a multi-objective framework that minimizes the number of stations and their length (related to the required area), while maximizing several robustness criteria. In Chica et al. (2019), the authors extend the approach using simulation to create representative demand scenarios.

In Sikora (2021), the balancing of a mixed-model assembly line is combined with the products' sequencing for a set of demand scenarios. The approach considers the assignment of tasks to stations to be fixed for all scenarios, while the sequencing can be solved independently per scenario.

From the presented literature on demand variation, two groups of uncertainty can be identified – the first deals with variations of relative demands in multiple model production. The number of stations (and workers) in Li & Gao (2014), Yang & Gao (2016), Chica et al. (2016), Chica et al. (2019), and Sikora (2021) remains constant for all scenarios. Different demands affect the average processing times (and area requirements) in the stations, but the total production levels are roughly constant. Alternatively, Simaria et al. (2009) use a different workforce size for each scenario, so that different absolute demand values strongly vary. Some flexibility for total demand variation is also encountered in Li & Gao (2014). They deal with the issue by employing overtime.

To the best of the author's knowledge, Simaria et al. (2009) is the only reference that considers the number of workers as a reaction to total demand variation. Nonetheless, their proposed method is based on a hierarchical approach without iteration between the assignment of tasks to stations and the scenario-specific assignments of workers to stations. Therefore, there is a gap in the literature to examine the integrated problem of balancing assembly lines and workforce planning for uncertain total demand levels. In this paper, we address such gap by extending the SALBP to consider multiple demand scenarios at the line design stage. The line layout is optimized to cope with different throughput requirements while not incurring in any re-balancing costs. Hence, we formally define the proposed ERALBP in Section 3 and develop a specialized heuristic procedure to efficiently solve it in Section 4.

3. Problem Description

The studied ERALBP takes more than one economic scenario (target cycle time) into account. It considers the standard SALBP definitions and its simplification hypotheses (Baybars, 1986). Tasks ($t \in T$) have known durations (D_t), are indivisible, subject to precedence relations ($(t_1, t_2) \in R$), and must each be assigned to a station ($s \in S$). Stations are equally equipped and the task-station assignment does not affect the task's duration. Finally, for the SALBP type-1, the demand is known and the sum of processing times in each station is bounded by the line's cycle time (C).

Here, an observation is necessary: stations are only considered equally equipped in a design stage. However, upon implementation of the chosen design, equipment is distributed to stations as a function of the tasks they perform. This distinction is relevant because some of the necessary equipment can be heavy or effectively immovable. Thus, by relaxing this simplification hypothesis, we take a step to further extend the classical SALBP into the proposed ERALBP.

Additionally, it is assumed that the line operates with a continuous conveyor and that tasks are performed on the product while it is moving. This continuously paced line control is usually considered the standard assumption of the relevant literature (Boysen et al., 2008). Assuming that the line operates at a fixed speed, physical positions in the line can be measured in temporal units from its start. Its continuous nature also implies that equipment is continuously distributed to reflect the sequence in which tasks are performed.

The ERALBP also evaluates multiple cycle times. While the model described herein can represent any number of demand scenarios, this paper focuses on only two. The first one represents the higher demand configuration, and is assumed to be the one with a lower cycle time. The high and low demands are modeled by considering two scenarios ($k \in K$), each with its own target cycle time (C_k).

The multiple scenario extension requires assignment decisions to be explicitly taken for each k. Thus, binary variables to control task-station assignments for each scenario k are created: $x_{t,s,k} = 1$ if task t is performed at station s. Naturally, each scenario may need a different number of stations, so we need station opening variables for each scenario: $y_{s,k} = 1$ if station s is active in scenario k. However, as the switching process between output rates cannot incur in any re-balancing costs, there can only be a single line layout – which in this case translates into a sequence of tasks. This procedure is controlled with a binary variable for each pair of different tasks t_1 and t_2, where $z_{t_1,t_2} = 1$ if t_1 is performed after t_2. While scenarios differ in number of stations and cycle time, task durations (D_t), task sequence (z_{t_1,t_2}), and precedence relations (R) must be the same for all scenarios. A MILP model to represent the ERALBP is given by Expressions (1)–(10).

$$\text{Minimize} \quad \sum_{k\,\in\,K} w_k \cdot \sum_{s\,\in\,S} y_{s,k} \tag{1}$$

subject to:

$$\sum_{s\,\in\,S} x_{t,s,k} = 1 \qquad \forall\, t\,\in\,T,\; k\,\in\,K \tag{2}$$

$$\sum_{s\,\in\,S} s \cdot x_{t_1,s,k} \leq \sum_{s\,\in\,S} s \cdot x_{t_2,s,k} \qquad \forall\,(t_1,t_2)\,\in\,R,\; k\,\in\,K \tag{3}$$

$$\sum_{t\,\in\,T} D_t \cdot x_{t,s,k} \leq C_k \qquad \forall\, s\,\in\,S,\; k\,\in\,K \tag{4}$$

$$y_{s,k} \geq x_{t,s,k} \qquad \forall\, t\,\in\,T,\; s\in\,S,\; k\,\in\,K \tag{5}$$

$$y_{s,k} \leq y_{s-1,k} \qquad \forall\, s\in\,S:\; s>1,\; k\,\in\,K \tag{6}$$

$$x_{t,s} = 0 \qquad \forall\, t\in\,T,\; k\in K,\; s\notin[E_{t,k},...,L_{t,k}] \tag{7}$$

$$z_{t_1,t_2} \cdot S_{max_k} \geq \sum_{s\,\in\,S} s \cdot (x_{t_2,s,k} - x_{t_1,s,k}) \qquad \forall\, t_1,t_2\,\in\,T:\; t_2>t_1,\; k\,\in\,K \tag{8}$$

$$(1 - z_{t_1,t_2}) \cdot S_{max_k} \geq \sum_{s\,\in\,S} s \cdot (x_{t_1,s,k} - x_{t_2,s,k}) \qquad \forall\, t_1,t_2\,\in\,T:\; t_2>t_1,\; k\,\in\,K \tag{9}$$

and:

$$x_{t_1,s,k},\; y_{s,k},\; z_{t_1,t_2} \;\in\; \{0,\;1\} \qquad \forall\, t_1,t_2\,\in\,T,\; s\,\in\,S,\; k\,\in\,K \tag{10}$$

Expression (1) presents the goal function, i.e. minimize the weighted sum of the number of required stations (w_k states the relative importance of scenario k). For all scenarios, the basic line balancing constraints apply: all tasks must be assigned (Equation (2)), precedence relations respected (Inequality (3)) as well as the cycle time (Inequality (4)). Inequality (5) states that the balancing decisions (x) are contingent on the station being active (y). Inequality (6) states a simple symmetry break. Expression 7 filters the problem's search space by discarding task-station allocations that necessarily imply on cycle time violations. This is done by using the earliest ($E_{t,k}$) and latest ($L_{t,k}$) stations concept (Scholl & Becker, 2006) for task t and each cycle time C_k. Constraints (8) and (9) demand that either t_1 is performed after t_2 or the reverse, imposing a single task sequence for all scenarios. In these constraints, S_{max_k} states an upper bound on the number of stations for each scenario k. Lastly, Expression 10 states the binary requirements for variables.

Given that the SALBP is NP-hard, hence, so is ERALBP: it reduces to SALBP in the case of a single scenario or two equal ones. Therefore, one can anticipate the herein presented MILP model (1)–(10) to perform poorly when attempting to solve large instances. Due to the expected combinatorial complexity of the proposed ERALBP, we introduce a heuristic procedure in Section 4 to quickly produce high-quality solutions for large instances of the problem. Finally, in order to demonstrate the necessity of a specialized method for the proposed problem, computational experiments are conducted in Section 5 to assess the performance of the MILP model and the heuristic procedure.

While the proposed method and the experiments in this paper focus on two cycle times, Appendix A presents a generalized version of the method for more cycle times.

4. Heuristic Procedure

The proposed approximate solution procedure consists in solving a SALBP instance with the first cycle time (C_1) and post-processing its solution to define a valid task sequence for the instance. Task assignments are fixed for the C_1 stations and task sequence within each C_1 station is optimized for C_2 subject to the problems' original precedence constraints. This ensures that the number of stations required for C_1 is not altered. Given an ERALBP instance with two cycle times C_1 and C_2 such that $C_1 < C_2$, an associated SALBP problem can be defined by setting its cycle time as the smallest value, i.e. C_1. For practical sizes, this instance can typically be efficiently solved using methods such as SALOME (Scholl & Becker, 2006) or Branch, Bound, and Remember (Sewell & Jacobson, 2012). The procedure is not guaranteed to reach the global optimum number of stations for C_2 because the SALBP solution imposes additional precedence constraints for C_2: all tasks assigned to the s^{th} C_1 station must be performed before those assigned to the $(s+1)^{\text{th}}$ one. Naturally, the problem's original precedence relations still impose partial ordering to the tasks within each station.

Given these additional precedence requirements, an optimized task sequence for the second cycle time can be obtained by solving a set of small precedence-constrained knapsack problems. This is achieved constructively by post-processing the SALBP's solution, as presented by Algorithm 1. In it, NS_1 states the number of stations for C_1 in the given solution.

Algorithm 1 Post-processing given SALBP solution

$NS_2 \leftarrow 1$ ▷ Open the first station for C_2
$A \leftarrow C_2$ ▷ Currently available time at opened station for C_2
$T_s \leftarrow$ set of tasks assigned to s^{th} station of SALBP's solution
TaskSequence $\leftarrow$ Empty List of Integers
for $s = 1$ to NS_1 **do** ▷ for each station in SALBP's solution
  if $\sum_{t\,\in\,T_i} D_t \leq A$ **then** ▷ All tasks in T_i fit currently opened station
    $A \leftarrow A - \sum_{t \in T_i} D_t$ ▷ Decrease available time at current C_2 station
    TaskSequence.Add(T_i) ▷ at the end, in any topological order
  else
    FitT $\leftarrow$ PrecKnapsack(A, T_i, D, R) ▷ Tasks that fit in the currently opened station for C_2
    RemT $\leftarrow$ T_i−FitT ▷ Remaining tasks that will be assigned in the next station for C_2
    TaskSequence.Add(FitT) ▷ at the end, in any topological order
    $NS_2 \leftarrow NS_2 + 1$ ▷ Close current C_2 station and open new one
    $A \leftarrow C_2 - \sum_{t \in \text{RemT}} D_t$ ▷ Open next station with appropriate available time
    TaskSequence.Add(RemT) ▷ at the end, in any topological order
  end if
end for
return (NS_2, TaskSequence)

In essence, this post-processing occurs by incrementing the task sequence with tasks assigned to SALBP's solution one station at a time. To do so, the number of stations used for C_2 (NS_2), and the remaining available time at the currently open station (A) are tracked. Because $C_1 < C_2$, for each station in SALBP's solution, one of two things can occur: either the sum of tasks is assigned to the C_1 station is inferior to the available time A, or it is bounded by $A + C_2$. In the former case, the order of this subset of tasks (T_s) is irrelevant for the C_2 solution and tasks are added topologically to the end of the sequence. In the latter case, tasks are split into those that fit the currently opened station (FitT) and those that do not (RemT). These sets depend on the line's task sequence. To determine one of the best such sequences, the proposed approach solves a Precedence-Constrained Knapsack Problem (PCKP) in line 10. The parameters for the associated PCKP instance are: the knapsack capacity (A), the set of items (T_s), their storage cost and item value (in this case, these are equal and given by D), and the set of precedence relations between items (R). Because the number of tasks in each station is typically not large, these knapsack problems tend to remain tractable.

4.1. Illustrative example

Consider an assembly line balancing instance defined by the precedence diagram and task durations described by Gunther (1983) and illustrated by Figure 1. Let $C_1 = 50$ and $C_2 = 70$ define the cycle times for high and low outputs scenarios, respectively. Using SALOME, the optimal SALBP solutions for each cycle time have, respectively, ten and eight stations.

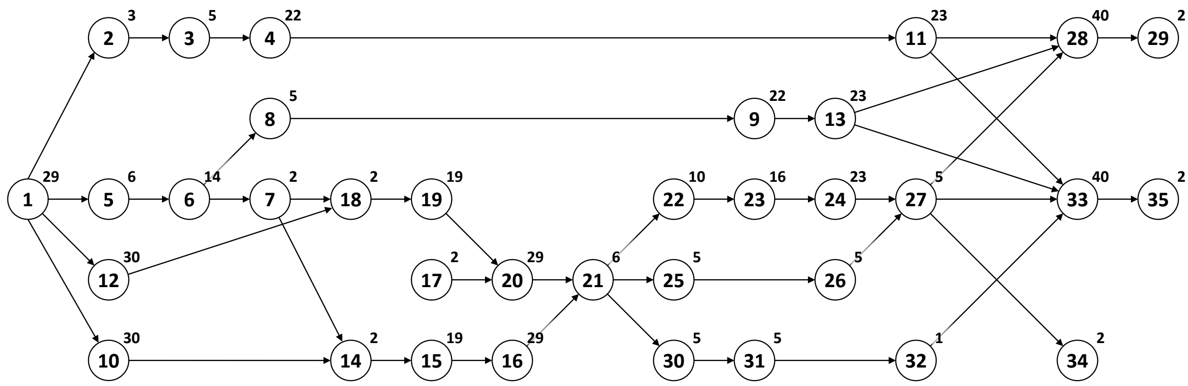

Figure 1: Gunther (1983)'s precedence diagram: task indexes (t) and durations (D_t) are presented within and next to circles, respectively

Figure 2 illustrates the solution obtained by the proposed heuristic for this particular instance. In this case, the heuristic found a guaranteed optimal answer as the numbers of stations match the SALBP optimal for both cycle times simultaneously. Its center row illustrates the task sequence, in which task indexes are labeled. Adjacent to it, stations or partitions for each cycle time are depicted.

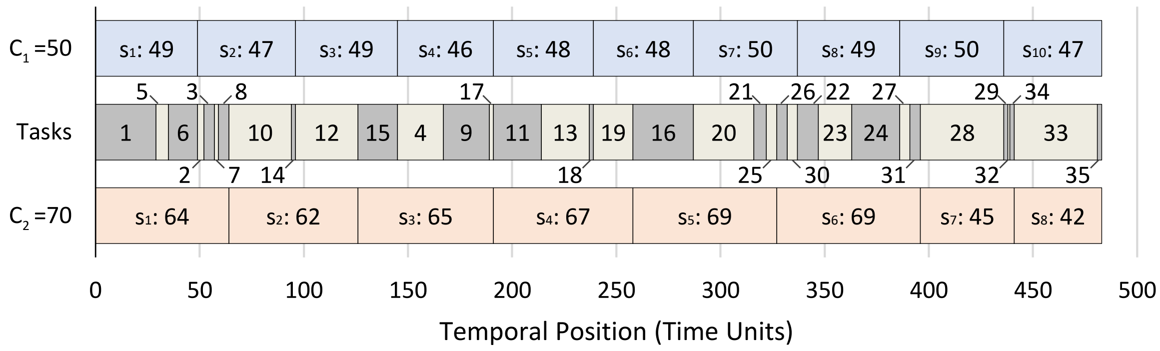

Figure 2: Optimal solution representation for the illustrative example

5. Computational Experiments

The MILP model presented in Section 3 and the heuristic method described in Section 4 were applied to instances generated using precedence diagrams from a well-known benchmark (Scholl, 1999). All experiments were conducted on a i7-8700 CPU (3.2 GHz, 6 CPUs, 12 threads) with 32GB RAM and a 3600 second time limit, Gurobi 9.0 was employed to solve the MILP models, and the SALBP solutions employed by the proposed heuristic were provided by using the SALOME method (Scholl & Becker, 2006). Problem data and the best solutions found for each instance are made available in the paper's supplementary material.

The cycle time values (C_1 and C_2) were defined based on SALBP benchmark values on the high, mid, and low ranges. Instances were produced by combining high-mid, mid-low and high-low values of cycle time for each precedence diagram. Furthermore, they are ordered by size (measured in number of tasks), grouped in small (less than 50 tasks), medium (between 50 and 100 tasks) and large instances (more than 100 tasks). Table 1 reports the tested instance parameters, columns C_1 and C_2 present the two cycle times of each instance, while columns L_1 and L_2 present the optimal SALBP number of stations for each cycle time. The combination of cycle times led to two instances: one considering the high demand scenario as the most relevant ($w_1 = 2$ and $w_2 = 1$), and the other with the opposite consideration ($w_1 = 1$ and $w_2 = 2$). Table 1 also reports the results in terms of the answer obtained by using only the proposed heuristic (UB_h), only the MILP model (UB_m) and of using the model warm-started by the heuristic's solution (UB_B). Lastly, the LB column reports the best lower bound for the instance, as informed by the MILP models, and the Gap columns inform the integer gap of the upper bound found by the heuristic and the MILP model ($(UB - LB)/UB$).

The heuristic is very fast for all instances, requiring at most 10s, including the solution of the related SALBP-1 instance. Nonetheless, it produced multiple optimal solutions, and the result is rather close to the best solution, when it did not. The MILP models are particularly useful for the small instances, which could all be solved to optimality. However, for the larger ones, it often failed to even reach a feasible solution (indicated by N/A). Combined, the proposed heuristic and the MILP model are able to reach solutions

Table 1: Computational experiment's results

Instance information			Results for $w_1 = 2$ and $w_2 = 1$						Results for $w_1 = 1$ and $w_2 = 2$					
Name (size)	C_1	C_2	UB_h	UB_m	UB_b	LB	Gap_h	Gap_m	UB_h	UB_m	UB_b	LB	Gap_h	Gap_m
Mitchel (21)	14	21	22	22	22	22	0%	0%	20	20	20	20	0%	0%
	21	35	14	14	14	14	0%	0%	13	12	12	12	7.7%	0%
	14	35	20	19	19	19	5%	0%	16	14	14	14	12.5%	0%
Gunther (35)	44	50	36	34	34	34	5.6%	0%	36	32	32	32	11.1%	0%
	50	70	28	28	28	28	0%	0%	26	26	26	26	0%	0%
	44	70	32	N/A	32	32	0%	100%	28	28	28	28	0%	0%
Kilbridge (45)	57	79	28	27	27	27	3.6%	0%	26	24	24	24	7.7%	0%
	79	184	18	17	17	17	5.6%	0%	15	13	13	13	13.3%	0%
	57	184	24	23	23	23	4.2%	0%	18	16	16	16	11.1%	0%
Hahn (53)	2004	2806	22	22	22	22	0%	0%	20	20	20	20	0%	0%
	2806	4676	16	16	16	16	0%	0%	14	14	14	14	0%	0%
	2004	4676	20	20	20	20	0%	0%	16	16	16	16	0%	0%
Tonge (70)	160	251	62	N/A	62	60	3.2%	100%	55	N/A	55	51	7.3%	100%
	251	527	35	N/A	35	35	0%	100%	28	30	28	28	0%	6.7%
	160	527	54	N/A	54	53	1.9%	100%	39	N/A	39	37	5.1%	100%
Arcus (83)	3786	5824	56	N/A	56	56	0%	100%	49	N/A	49	49	0%	100%
	5824	10816	36	N/A	36	36	0%	100%	30	N/A	30	30	0%	100%
	3786	10816	50	N/A	50	50	0%	100%	37	N/A	37	37	0%	100%
Arcus2 (111)	5755	7916	76	N/A	76	74	2.6%	100%	71	N/A	71	67	5.6%	100%
	7916	17067	50	N/A	50	49	2%	100%	40	N/A	40	38	5%	100%
	5755	17067	63	N/A	63	63	0%	100%	45	N/A	45	45	0%	100%
Barthold (148)	403	513	41	40	41	39	4.9%	2.5%	40	38	40	36	10%	5.3%
	513	805	30	N/A	29	29	3.3%	100%	27	N/A	25	25	7.4%	100%
	403	805	36	36	35	35	2.8%	2.8%	30	N/A	28	28	6.7%	100%
Scholl (297)	1422	1883	139	N/A	139	137	1.4%	100%	128	N/A	128	124	3.1%	100%
	1883	2787	100	N/A	100	99	1%	100%	89	N/A	89	87	2.2%	100%
	1422	2787	127	N/A	127	125	1.6%	100%	104	N/A	104	100	3.8%	100%

with low integer gaps even for the large instances (2% on average).

The proposed heuristic tends to perform better for the instances in which the high demand scenario is considered the most relevant. That occurs because it employs an optimal SALBP-1's solution of the lower cycle time to produce the ERALBP's solution. This process ensures that the resulting solution will have the optimal number of stations for that cycle time. However, in many instances, this led to solutions that also have the optimal number of stations for the higher cycle time, meaning these are optimal regardless of the relevance weights (w_1 and w_2) for the scenarios. Consequently, the relevance weights attributed to each scenario did not affect the best solutions found by the heuristic and the MILP models. However, one exception occurred: The instance Mitchel ($C_1 = 21$, $C_2 = 35$) had optimal solutions with 5-4 or 6-3 stations, respectively, depending on the relevance weights w_1 and w_2.

5.1. Instance Parameter Influence on large instances

The proposed heuristic is applied to Otto et al. (2013)'s dataset in order to test its behavior in general for large instances. While the MILP model described in Section 3 is also tested, it failed to reach solutions for all instances within the time limit. Therefore, this section reports only the heuristic's results. Due to the number of instances, the MILP time limit is set to 1800 seconds for this section's experiments. Furthermore, the SALBP initial solutions used by the proposed heuristic in this section were provided by the Branch, Bound, and Remember method (Sewell & Jacobson, 2012) using a time limit of 30s.

The proposed method is applied to the 525 large instances, all of which have 100 tasks, but vary regarding the distribution of task durations and the structure of the precedence graph. In these experiments, the cycle time values of $C_1 = 1000$ and $C_2 = 1500$ are used. Table 2 reports the average solution Gap for each instance type and for each relative relevance weight (w_1 and w_2) as well as the average time required by the heuristic procedure and the number of optimal solutions found (OPT column) in relation to the number of tested instances. The Ordering Strength parameter informs how restricted the precedence diagram is, with higher values meaning more restrictive instances. The Task Durations parameter informs how these compare to the instances' original cycle time (1000 T.U.): these are bimodal (BM), peak in the bottom (PB), and peak in the middle (PM). Further descriptions of these parameters are given by Otto et al. (2013).

Table 2: Experiment's results for the Otto et al. (2013)'s dataset

Instance Parameters		Solution Gap $w_1 = 2 \cdot w_2$	$w_1 = w_2$	$2 \cdot w_1 = w_2$	Time (s)	OPT
Order Strength	0.2	2.0%	2.9%	3.9%	1.1	83/225
	0.6	2.1%	3.0%	4.0%	0.75	86/225
	0.9	1.5%	2.4%	3.4%	0.49	27/75
Task Durations	BM	1.3%	2.0%	2.8%	0.33	49/175
	PB	0.4%	0.7%	0.9%	0.32	147/175
	PM	4.1%	5.8%	7.7%	9.08	0/175
Average		1.9%	2.8%	3.8%	3.25	196/525
Maximum		8.6%	10.1%	12.1%	35.0	

Table 2 allows an analysis of the impact of each type of instance parameters for the proposed method's performance. A parameter with rather direct impact is the relative relevance of each cycle time (w_1 and w_2): when w_2 is higher, solution gaps are higher. This result reflects the fact that the method post-processes a SALBP solution for C_1. Table 2 also suggests that Order Strength is a less relevant factor for the method's performance. Nonetheless, more restrictive instances (0.9) displayed lower solution gaps, meaning they tend to be easier, as expected. The task duration parameter appears to have greater relevance for the algorithm performance: when task times are short relative to cycle time (PB), instances tend to be easier, as reflected by a less than 1% average solution gap; the bimodal duration distribution (BM) displayed intermediary behavior, but the hardest distribution, by far is the "Peak in the Middle" distribution (PM). This outcome is expected as these distributions are also tied to harder, however less common in practice, SALBP instances (Otto et al., 2013). The number and percentage of instances for which the proposed method reached guaranteed optimal solutions (minimum number of stations for both cycle times) also provides similar insights into instance difficulty, and mimics the behavior displayed by the Solution Gaps.

6. Industrial Data Case Study

This section reports the results of applying the proposed method to practical data of a gearbox assembly line, which is also made available in this paper's supplementary material for reproducibility purposes. This line had been one of the subjects of a re-balancing study (Sikora et al., 2017). It is dedicated to the manufacture of a single product model, and it contains manual workers as well as two specialized robots. In this line, robots act as single-purposes stations that must (and can only) perform specific tasks, therefore "splitting" the line in three segments. Sikora et al. (2017) describe that the line was initially set to an "as-is" layout of 20 total stations (18 workers and 2 robots) operating at a cycle time of 1541 time units. This initial layout resulted from a previous adaptation effort caused by a drop in demand. It also fixed some tasks to specific positions, because of heavy machinery whose re-locations are considered too costly. After re-balancing, Sikora et al. (2017) reach a minimum cycle time of 1345 time units in a solution with 17 total stations.

This case study compares that approach to what can occur if the line is designed to anticipate such demand fluctuations. Two scenarios are considered: a "high demand" one, in which it is desired that the line operates at maximum throughput capacity, and a "low demand" which is set as the "as-is" configuration of 1541 time units. Based on the instance's task durations (Scholl & Becker, 2006), the line's lowest possible cycle time is 1200 time units. Using standard literature SALBP-1 methods[1], it has been determined that the optimal number of total stations for these scenarios is 20 for the high demand one, and 16 for the low demand one. An ERALBP solution has been produced by applying the proposed heuristic to each line "segment" between the robotic workstations of the "high-demand" scenarios. The resulting solution can operate at both cycle times at their optimal number of total stations. The solution is illustrated in Figure 3: each w_i box represents a worker, and R_1 and R_2 represent the dedicated robots.

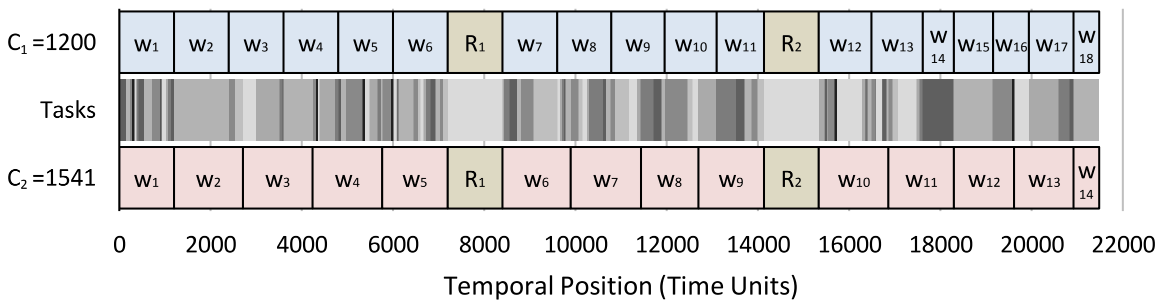

Figure 3: Optimal solution representation for the case study

Compared to the literature's re-balancing approach, this solution presents two advantages: First, it allows a 10% higher production capacity in the "high-demand" scenario than what was possible by re-balancing the line. Second, all tasks are assigned to the same position in the line for both scenarios, allowing straightforward alternations between throughput capacities. The first advantage can be partly attributed to the fact that the

[1]In order to model the robots and their specific relationship to the tasks that they must perform, these tasks had their processing times set to the cycle time of each demand scenario (1541 and 1200 time units) in order to produce a SALBP-1 instance equivalent to the studied problem. SALOME was employed to solve the SALBP instances.

re-balancing approach is more limited in its task-station assignments by the impractical re-positioning costs of a few tasks, leading to a higher cycle time. The second one highlights how the proposed method can meet the challenges of demand fluctuations tied to predictable patterns (seasonality, future production goals) and unexpected events (such as pandemics).

7. Discussions and Conclusions

This paper discussed the design and balancing of an assembly line that can efficiently switch between two output levels. This throughput difference is modeled as two different cycle times. A mixed-integer linear programming (MILP) model and an efficient heuristic procedure are presented. The latter consists in post-processing a line balancing solution for one of the problem's cycle times. The heuristic is capable of quickly producing good-quality solutions even for large instances of the problem. Combined with the MILP model, it reached solutions with an average integer gap of 2% for the large instances, including some optimal solutions. Tests with a 525-instance dataset confirm the heuristic's solid performance for larger problems, finding 196 optimal solutions in total. Furthermore, these tests show that, similarly to SALBP, the studied problem's difficulty is affected by the strength of precedence relations and the distribution of task durations.

For businesses that can handle sudden changes in production rate, this fast switching between output rates means better benefiting from economic opportunities tied to increased seasonal effects or unexpected demand spikes. In some sectors, sudden changes to production rates are not as trivial. Nonetheless, the design of assembly lines that can handle such output switch represents a relevant economic opportunity: namely designing lines that are optimized for an initial production rate (cycle time), while also anticipating a planned higher throughput in the future. This flexibility can reduce costs tied to re-balancing the line.

A case study with industrial data further illustrates these economic opportunities: By designing the line in a manner that anticipates demand fluctuations, it is possible to reach better high-demand performance than what was achieved by previous efforts in re-balancing the line. Furthermore, the alternation between the two cycle time scenarios is less costly than the re-balancing alternative, as all tasks maintain their physical positions in both scenarios.

Further works should extend the proposed formulation to incorporate other problem features such as mixed-model lines, parallel stations, or treating subsets of tasks as flexible. Furthermore, other exact methods, such as Combinatorial Benders' Decomposition and Branch-and-Bound techniques, can also be applied to the problem.

Appendix A. Generalization for more cycle times

Algorithm 2 presents a generalization of the heuristic method presented in Section 4. It employs Algorithm 1 iteratively, revising the set of precedence relations after each iteration. By assumption, the cycle times are ordered such that $C_j < C_{j+1}$, and the given SALBP solution is optimized for C_1. In each iteration, the partition for cycle time C_j imposes additional partial ordering constraints for the following cycle times: tasks assigned to one station in that partition have effective precedence over tasks assigned to subsequent stations. This information is carried over to the following iterations by adding the new partial ordering requirements to the precedence relations set R, meaning task sequences in subsequent iterations necessarily retain the partition quality of previous cycle times.

Algorithm 2 Solution method for multiple cycle times

$T_s(C_1) \leftarrow$ set of tasks assigned to s^{th} station ▷ Original SALBP solution
for each C_j: $j > 1$ **do** ▷ for each cycle time
    $(NS_j, \text{TaskSequence}) \leftarrow$ **Algorithm 1**$(T_s(C_1), C_j, D, R)$
    $T_s(C_j) \leftarrow$ set of tasks assigned to s^{th} for C_j given TaskSequence
    $R_{new} \leftarrow \{(t_1, t_2) | t_1 \in T_s(C_j),\ t_2 \in T_{s+1}(C_j)\}$ ▷ New implied precedence relations
    $R \leftarrow R \cup R_{new}$ ▷ Combine new precedences to original ones
end for
return $(NS_J, \text{TaskSequence})$ ▷ Number of stations for each C_j and final task sequence

References

Álvarez-Miranda, E., & Pereira, J. (2019). On the complexity of assembly line balancing problems. *Computers and Operations Research*, *108*, 182–186. doi:10.1016/j.cor.2019.04.005.

Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics*, *142*, 259–277. doi:10.1016/j.ijpe.2012.10.020.

Baybars, I. (1986). Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem. *Management Science*, *32*, 909–932.

Ben-Tal, A., Ghaoui, L. E., & Nemirovski, A. (2009). *Robust Optimization*. (1st ed.). Princeton: Princeton University Press. doi:10.1007/s10957-013-0421-6. arXiv:arXiv:1011.1669v3.

Bentaha, M. L., Dolgui, A., & Battaïa, O. (2015). A bibliographic review of production line design and balancing under uncertainty. *IFAC-PapersOnLine*, *28*, 70–75. doi:10.1016/j.ifacol.2015.06.060.

Birge, J. R., & Louveaux, F. (2011). *Introduction to Stochastic Programming*. (2nd ed.). New York: Springer-Verlag. doi:10.1007/978-1-4614-0237-4.

Bortolini, M., Faccio, M., Gamberi, M., & Pilati, F. (2017). Multi-objective assembly line balancing considering component picking and ergonomic risk. *Computers and Industrial Engineering*, *112*, 348–367. doi:10.1016/j.cie.2017.08.029.

Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics*, *111*, 509–528. doi:10.1016/j.ijpe.2007.02.026.

Chica, M., Bautista, J., & de Armas, J. (2019). Benefits of robust multiobjective optimization for flexible automotive assembly line balancing. *Flexible Services and Manufacturing Journal*, *31*, 75–103. doi:10.1007/s10696-018-9309-y.

Chica, M., Bautista, J., Cordón, Ó., & Damas, S. (2016). A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand. *Omega*, *58*, 55–68. doi:10.1016/j.omega.2015.04.003.

Defersha, F. M., & Mohebalizadehgashti, F. (2018). Simultaneous balancing, sequencing, and workstation planning for a mixed model manual assembly line using hybrid genetic algorithm. *Computers and Industrial Engineering*, *119*, 370–387. doi:10.1016/j.cie.2018.04.014.

Eghtesadifard, M., Khalifeh, M., & Khorram, M. (2020). A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990–2017. *Computers & Industrial Engineering*, *139*, 106182. doi:https://doi.org/10.1016/j.cie.2019.106182.

Gamberini, R., Grassi, A., & Rimini, B. (2006). A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem. *International Journal of Production Economics*, *102*, 226–243. doi:10.1016/j.ijpe.2005.02.013.

Gunther, R. (1983). Currently practiced formulations for the assembly line balance problem. *Journal of Operations Management*, *3*, 209–221. doi:10.1016/0272-6963(83)90005-0.

Hazir, Ö., Dolgui, A., Hazır, Ö., Dolgui, A., Hazir, Ö., & Dolgui, A. (2019). A Review on Robust Assembly Line Balancing Approaches. *IFAC-PapersOnLine*, *52*, 987–991. doi:https://doi.org/10.1016/j.ifacol.2019.11.323.

Hui, D. S., I Azhar, E., Madani, T. A., Ntoumi, F., Kock, R., Dar, O., Ippolito, G., Mchugh, T. D., Memish, Z. A., Drosten, C., Zumla, A., & Petersen, E. (2020). The continuing 2019-nCoV epidemic threat of novel coronaviruses to global health — The latest 2019 novel coronavirus outbreak in Wuhan, China. *International Journal of Infectious Diseases*, *91*, 264–266. doi:10.1016/j.ijid.2020.01.009.

Ivanov, D., & Dolgui, A. (2020). Viability of intertwined supply networks: extending the supply chain resilience angles towards survivability. A position paper motivated by COVID-19 outbreak. *International Journal of Production Research*, *58*, 2904–2915. doi:10.1080/00207543.2020.1750727.

Kao, E. P. C. (1976). A Preference Order Dynamic Program for Stochastic Assembly Line Balancing. *Management Science*, *22*, 1097–1104. doi:10.1287/opre.26.6.1033.

Li, J., & Gao, J. (2014). Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research*, *52*, 3552–3567. doi:10.1080/00207543.2013.874603.

Liu, R., Liu, M., Chu, F., Zheng, F., & Chu, C. (2020). Eco-friendly mulit-skilled worker assignment and assembly line balancing problem. *Computers and Industrial Engineering*, *150*, 106944. doi:10.1016/j.cie.2020.106944.

Lopes, T. C., Brauner, N., & Magatão, L. (2021). Assembly Line Balancing with Fractional Task Allocations. *International Journal of Production Research*, *in press*. doi:10.1080/00207543.2020.1866224.

Makssoud, F., Battaïa, O., Dolgui, A., Mpofu, K., & Olabanji, O. (2015). Re-balancing problem for assembly lines: new mathematical model and exact solution method. *Assembly Automation*, *35*, 16–21. doi:10.1108/AA-07-2014-061.

Öner-közen, M., Minner, S., & Steinthaler, F. (2017). Efficiency of Paced and Unpaced Assembly Lines under Consideration of Worker Variability - A Simulation Study. *Computers and Industrial Engineering*, *111*, 516–526. doi:10.1016/j.cie.2017.03.030.

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. *European Journal of Operational Research*, *228*, 33–45. doi:10.1016/j.ejor.2012.12.029.

Scholl, A. (1999). *Balancing and Sequencing of Assembly Lines*. (2nd ed.). Heidelberg: Physica.

Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research*, *168*, 666–693. doi:10.1016/j.ejor.2004.07.022.

Sewell, E. C., & Jacobson, S. H. (2012). A Branch, Bound, and Remember Algorithm for the Simple Assembly Line Balancing Problem. *INFORMS Journal on Computing*, *24*, 433–442. doi:10.1287/ijoc.1110.0462.

Sikora, C. G. S. (2021). Benders' decomposition for the balancing of assembly lines with stochastic demand. *European Journal of Operational Research*, *292*, 108–124. doi:10.1016/j.ejor.2020.10.019.

Sikora, C. G. S., Lopes, T. C., & Magatão, L. (2017). Traveling worker assembly line (re)balancing problem: Model, reduction techniques, and real case studies. *European Journal of Operational Research*, *259*, 949–971. doi:10.1016/j.ejor.2016.11.027.

Simaria, A. S., Zanella De Sá, M., & Vilarinho, P. M. (2009). Meeting demand variation using flexible U-shaped assembly lines. *International Journal of Production Research*, *47*, 3937–3955. doi:10.1080/00207540701871044.

Yang, C., & Gao, J. (2016). Balancing mixed-model assembly lines using adjacent cross-training in a demand variation environment. *Computers & Operations Research*, *65*, 139–148. doi:10.1016/j.cor.2015.07.007.

Zhang, Z., Tang, Q., & Chica, M. (2020). Multi-manned assembly line balancing with time and space constraints: A MILP model and memetic ant colony system. *Computers and Industrial Engineering*, *150*, 106862. doi:10.1016/j.cie.2020.106862.