Taylor & Francis
Taylor & Francis Group



International Journal of Production Research

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/tprs20

Conserving workforce while temporarily rebalancing assembly lines under demand disruption

Adalberto Sato Michels & Alysson M. Costa

To cite this article: Adalberto Sato Michels & Alysson M. Costa (2022) Conserving workforce while temporarily rebalancing assembly lines under demand disruption, International Journal of Production Research, 60:21, 6616-6636, DOI: 10.1080/00207543.2021.1998694

To link to this article: https://doi.org/10.1080/00207543.2021.1998694

Published online: 10 Nov 2021.

Submit your article to this journal

Article views: 354

View related articles

View Crossmark data

Citing articles: 4 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tprs20

INTERNATIONAL JOURNAL OF PRODUCTION RESEARCH
2022, VOL. 60, NO. 21, 6616–6636
https://doi.org/10.1080/00207543.2021.1998694

Taylor & Francis
Taylor & Francis Group

Check for updates

Conserving workforce while temporarily rebalancing assembly lines under demand disruption

Adalberto Sato Michels and Alysson M. Costa

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

ABSTRACT

In stable circumstances, assembly lines have workers with different capabilities assigned to stations. They perform a set of specialised tasks multiple times daily. Under a situation of high demand disruption (e.g. the COVID-19 pandemic), the overproduction rate would lead inventory levels to soar. An approach to cope with these demand drops and ongoing workforce costs is to dismiss employees and rebalance the line. Nevertheless, this implies social and economic costs related to rehiring and training. Alternatively, agreements can be made to reduce workload with a proportional wage deduction. These decisions are particularly challenging in heterogeneous workforces. We propose a Mixed-Integer Linear Programming (MILP) model to address the Assembly Line Worker Assignment and Rebalancing Problem (ALWARP). Our model aims at preserving jobs while minimising labour costs. We consider scenarios with falling demands and impose regularity metrics on workload reductions. Computational tests on benchmark datasets show that our strategy can distribute social costs among workers, with only slightly higher cumulative labour hours, while avoiding inconveniences associated with future renovations. A real-world case study of a truck cabin assembly line is investigated: the model can easily incorporate many realistic features and decide which workers should have their workload reduced and at which rate.

ARTICLE HISTORY
Received 24 July 2020
Accepted 18 October 2021

KEYWORDS
Assembly line rebalancing; heterogeneous workers; workforce conservation; temporary demand disruption; mixed-integer linear programming

1. Introduction

Assembly lines are manufacturing systems commonly applied to high-volume industries of standardised products. Their layout is frequently based on flow-shops, i.e. product-oriented designs built to fit commodities. They are generally dedicated to producing homogeneous goods, a characteristic that is convenient for mass production. In academic research, assembly lines have originated a combinatorial problem widely discussed in the literature (Battaïa and Dolgui 2013): the Assembly Line Balancing Problem (ALBP).

In its simplest form, several simplification hypotheses are considered (Baybars 1986) to define the so-called Simple Assembly Line Balancing Problem (SALBP). The SALBP consists of assigning a set of tasks to stations, whilst respecting constraints imposed by a precedence graph. Furthermore, these stations must be organised in a straight, serial line producing a single product's unique model. The most studied objectives in a SALBP are the minimisation (i) of the number of stations (SALBP-1) or (ii) of the cycle time (SALBP-2). Scholl and Becker (2006) surveyed important contributions to its literature, which listed various exact algorithmic solution methods and practical applications. Among them, SALOME is present as an effective bidirectional branch-and-bound (B&B) procedure developed by Scholl and Klein (1997). Besides, Bautista and Pereira (2009) proposed a dynamic programming approach, which was followed by a branch, bound, and remember (BBR) algorithm (Sewell and Jacobson 2012), and an enhanced multi-Hoffmann heuristic (Sternatz 2014). Morrison, Sewell, and Jacobson (2014) added a new backtracking rule to save memory in the BBR algorithm and proved optimal solutions for 86% of unsolved problems in a new dataset (Otto, Otto, and Scholl 2013). These and other techniques were gathered in an overview and further improved for the SALBP-1 by Pape (2015). Finally, Kramer, Dell'Amico, and Iori (2017) generalised precedence constraints for the bin packing problem and developed an iterated local search algorithm to solve it. They also demonstrated that their method is a valid approach for the SALBP-1 with generalised precedence graphs.

Industrial assembly lines frequently rely on fixed physical layouts, which can be hard to rearrange.

CONTACT Adalberto Sato Michels a.satomichels@unimelb.edu.au

© 2021 Informa UK Limited, trading as Taylor & Francis Group

Furthermore, it is often the case that the workforce is trained to conduct specialised tasks. Therefore, it is usually unrealistic to implement a (re)balancing solution from scratch without reckoning the active workforce and line configurations (Falkenauer 2005). Since task reassignment of an operating line has several restrictions, managers might face some difficulty changing previously implemented decisions. This adaptability issue gives rise to a natural extension and more generalised version of ALBPs: the Assembly Line Rebalancing Problem (ALRP). Along with a variety of other practical extensions, Boysen, Fliedner, and Scholl (2007) classified rebalancing or reconfiguration problems as challenging open fields to narrow the gap between research and practice. This survey suggested that ALRPs are commonly modelled with either assignment restrictions (Buxey 1974; Scholl, Fliedner, and Boysen 2010) or machinery moving costs (Gamberini, Grassi, and Rimini 2006).

Other impractical simplification hypotheses are assuming that (i) any task can be processed at any station and (ii) task processing times are independent of the station to which it is assigned. In essence, this assumes that identical workers or machines perform all tasks. When task processing times are different depending on who performs the task, the SALBP is generalised into the Assembly Line Worker Assignment and Balancing Problem (ALWABP), as proposed by Miralles et al. (2008). Previous goal functions of minimising resources (workers/stations) or cycle time are still valid here. This problem has been widely applied to sheltered work centres (Miralles et al. 2008), in which workers with disabilities are unable to perform some of the tasks. Meanwhile, for the ones that they are capable of performing, their task processing times vary according to their abilities. We highlight that even if the ALWABP was proposed for sheltered work centres, heterogeneous/specialised workforces are present in other industrial settings, including, for example, robotic lines with different available technologies.

These problem variants were addressed in stable and mostly predictable conditions. In late 2019, however, the COVID-19 has been identified in Wuhan, a city in China (Hui et al. 2020). A few months later, in early 2020, it became a global pandemic, leading many countries to adopt strict social measures to slow down the virus propagation and prevent infections from spreading. The imposed restrictions naturally impacted the demand of some manufactured products negatively, pushing industries to apply collective leaves for a couple of weeks. After this period, nevertheless, production levels could not be maintained at the usual rate due to inventory capacities and raw material scarcity. In order to avoid massive unemployment rates, governments allowed companies to make agreements with their workers by reducing labour hours with a proportional deduction in salaries. For instance, governments in Australia, United Kingdom (UK), United States (US), and Brazil announced financial support programs to businesses and workers. On March 25, 2020, the Coronavirus Economic Response Package Omnibus Act (Australian Government 2020a) was deferred by the Australian government, followed by its payment and benefit rules (Australian Government 2020b) a month later. This program became known as JobKeeper and provides payments to assist households and businesses. It temporarily helps industries significantly impacted by the outbreak to cover their employees' costs and manage short-term cash flow challenges. More citizens can retain their jobs, carry on earning an income, and ensure a continued flow of credit in the economy. On the same date, government programs in the UK began to promote financial support for workers on furlough. Employers can apply for staff to get up to 80% of their payroll (Parliament of the United Kingdom 2020). Loans, tax relief, and cash grants are also available to support small businesses. Established two days later, on March 27, 2020, the Coronavirus Aid, Relief, and Economic Security (CARES) Act bill (H.R. 748) in the US started to provide assistance to workers through income additions and tax reliefs, small businesses with resources for maintaining payroll and hire back employee, and industries by preserving jobs adversely impacted by the outbreak via employee retention credits and payroll tax deferrals (United States Congress 2020). Finally, on April 1, 2020, the Brazilian government passed temporary legislation (the MPV 936/2020) to permit workload reductions of 25%, 50%, 70%, or even more depending on the situation (Brazilian Ministry of Economy 2020). Financial help has also been approved for individuals.

Recognising that this is a temporary circumstance is an important assumption. Usually, a sinking market demand would be followed by enterprises announcements on downsizing their workforces by means of mass lay-off and involuntary furloughs. Such kind of reaction has certainly been triggered by the recent COVID-19 outbreak as well. Still, in the context of manufacturing industries, these drastic measures are being carefully evaluated. Since governments permitted companies and employees to accommodate workload reductions with a proportional wage deduction in mutual agreements, these industries have ample incentives to conserve their workforce during this upheaval.

The most straightforward approach would be simply dismissing workers up to the point that the remaining ones would be enough to keep up with the shrunken demand, which implies in high dismissal costs and even higher rehiring, training, and loss of product

quality standards in the near future. Alternatively, the well-trained workforce can be maintained for further production re-establishment with reduced workloads. This latter approach also has a positive social impact: none of the employees would lose their jobs in the midst of an economic crisis. As the current circumstance is predicted to be a temporary setback, these financial aids allow a trade-off that is concerned with for both sides. With industries because rehiring a well-trained, heterogeneous workforce may be troublesome and costly in terms of product quality standards and retraining. Also with employees because they are able to keep their jobs whilst partially receiving a monetary compensation on the financial default they would experience (e.g. from government supports).

Motivated by these facts, the Assembly Line Worker Assignment and Rebalancing Problem (ALWARP) is proposed and investigated in this paper. It focuses on integrating ALWABP and ALRP under scenarios with an acute decline in demand, considering conservation of a heterogeneous workforce, workload reduction regularity, and social distancing recommendations. As in the original SALBP, we consider a serial line that produces a unique single product model, although this work can be promptly generalised for other layouts or mixed-model production situations.

The remainder of this work is organised as follows. In Section 2, the relevant literature for both ALWABP and ALRP is introduced, emphasising particular objective functions, extensions, and methodologies used by each author. Section 3 is meant to precisely define and explain the characteristics of an ALWARP under temporary demand disruption conditions, demonstrating advantages associated with workforce conservation. Section 4 presents Mixed-Integer Linear Programming (MILP) models considering different assumptions regarding the workforce reduction regularity. Moreover, practical extensions are discussed and further modelling decisions in the case of the studied truck cabins assembly line are presented. Computational experiments use benchmark datasets and are discussed in Section 5. The full industrial case study problem is explained, modelled, and solved in Section 6. Lastly, Section 7 ends this paper with concluding remarks and a discussion on further research directions.

2. Literature review

This section focuses on reviewing the main literature contributions for the ALWABP and the ALRP. Section 2.1 goes through the historical evolution of the ALWABP in the literature, pointing out modelling decisions and proposed solution methods. Section 2.2 starts describing the ALRP, indicates how it inspired the development of this current work, and connects the most relevant literature to its time-line. Finally, Section 2.3 compares the current work with the most similar references and justifies its position in the literature.

2.1. The assembly line worker assignment and balancing problem

The main motivation for Miralles et al. (2008) to firstly bring the issue of worker heterogeneity into assembly lines was to address balancing problems in sheltered work centres that employ workers with disabilities. The Assembly Line Worker Assignment and Balancing Problem (ALWALBP) has been defined and B&B procedures to solve it were developed. Since then, the problem has received a fair amount of attention in the literature. For instance, Costa and Miralles (2009) soon considered an ALWABP with job rotations, proposing an MILP formulation and a mathematical programming based heuristic decomposition method.

Thereafter, various approximate and exact methods to solve the original ALWABP have been proposed. Some of the main heuristic contributions have been a constructive heuristic based on priority rules (Moreira et al. 2012), an iterative Genetic Algorithm (GA) (Mutlu, Polat, and Supciller 2013), and a two-phase Variable Neighbourhood Search (VNS) algorithm (Polat et al. 2016). On terms of exact methods, both a task-oriented branch-and-bound (TBB) algorithm (Borba and Ritt 2014) and a station-oriented branch, bound, and remember (SBBR) algorithm (Vilà and Pereira 2014) have been proposed.

Moreira et al. (2012) assign tasks and workers to stations depending on proposed priority rules, followed by Mutlu, Polat, and Supciller (2013) obtaining new best solutions with a reduced computational processing time. Polat et al. (2016) also compared the results found on a benchmark dataset with the former heuristics. Additionally, they investigated an application of the method on an LCD TV industrial case study. Borba and Ritt (2014) and Vilà and Pereira (2014) proposed their exact methods simultaneously: the SBBR from the latter authors outperforms the TBB from the former in the majority of instances. However, it can be stated that the algorithms are complementary due to the fact that TBB concentrates on getting better lower bounds. In contrast, SBBR mainly relies on memory-based dominance rules. Thus, TBB can optimally solve numerous instances for which SBBR runs out of memory and outperforms it on instances with high order strength, whilst SBBR proves optimality for substantially more instances with low order strength and short task execution times.

Some variants of the initial problem were equally explored. Models for cases with parallel stations, collaboration between heterogeneous workers, and parallel lines in the context of sheltered work centres were studied by Araújo, Costa, and Miralles (2012, 2015). A weighted average had to be pondered to evaluate cycle times, as stations and lines in parallel might deliver distinct production rates. Collaboration between workers assigned to the same station is allowed in order to overcome individual inefficiencies. Each variant was also dealt with heuristic approaches.

The ALWABP was considered for mixed-model assembly lines in Cortez and Costa (2015) and Ramezanian and Ezzatpanah (2015). Ramezanian and Ezzatpanah (2015) minimises cycle time and operating costs and proposes an evolutionary, imperialist competitive algorithm (ICA) to solve the problem. Cortez and Costa (2015), in turn, model the sequencing problem with the goal of reducing the use of utility workers and propose local search and GRASP heuristics based on a surrogate objective function modelled as a linear program.

Sungur and Yavuz (2015) proposed a variant named Assembly Line Balancing with Hierarchical Worker Assignment (ALBHW) problem, which occurs when tasks and workers have different qualification requirements and levels. An ALBHW problem's objective is total cost minimisation, as it assigns workers and tasks to the stations based on their hierarchical ranks. Naturally, well-qualified workers are more costly, but they can perform a higher variety of tasks and replace any lower-qualified worker. Sungur and Yavuz (2015) formulated an MILP model to solve the ALBHW problem, tested it on several instances, and analysed its computational results.

The ALWABP was also examined in the context of U-shaped assembly lines by Oksuz, Buyukozkan, and Satoglu (2017). A Mixed-Integer Non-Linear Programming (MINLP) model for the type-E problem was developed. The objective is to maximise efficiency or, in other words, minimise the cycle time and the number of stations at the same time. Two meta-heuristics were developed alongside the model to achieve results as efficient as the MINLP formulation in less computational time: an Artificial Bee Colony (ABC) algorithm and a GA.

Finally, Moreira, Miralles, and Costa (2015) and Moreira et al. (2017) studied integrating disable workers into regular assembly lines, but would still consider all 'conventional' workers as identical. Ultimately, most of the workforce heterogeneity research has focussed on assembly lines found in sheltered centres. That is understandable because the inspiration for the field had been set in those environments. Although these studies may help companies offer more jobs for people with disabilities by considering their specific limitations, they have not taken into account job specialisation and heterogeneity for 'conventional' workers if a line rebalancing was required.

2.2. The assembly line rebalancing problem

The intense necessity of assembly line rebalancing in industrial environments led to the publication of multiple studies regarding the ALRP in the academic literature. Corominas, Pastor, and Plans (2008) were among the first to study the problem by tackling the rebalancing of a motorcycle assembly line plant using deterministic data. Hiring temporary, unskilled staff to work among permanent, skilled workers is required due to a seasonal increase in demand. These unskilled workers either need more time to carry out their tasks or are incapable of performing them. Concomitantly, multi-objective heuristic methods to address a stochastic version of the problem were proposed by Gamberini, Grassi, and Rimini (2006) and Gamberini et al. (2009). They were based on single-pass heuristic algorithms to conduct task reassignment in assembly lines that were going through product design changes and, consequently, in precedence relations, task processing times, and cycle time. Their algorithms contemplate initial and final task assignment similarities, thus attempting to maximise final rebalanced configuration similarity, as well as to minimise expected assembly costs and task reassignment.

The pioneer studies in Corominas, Pastor, and Plans (2008), Gamberini, Grassi, and Rimini (2006), and Gamberini et al. (2009) have been followed by Oliveira et al. (2012), who solve a rebalancing case in an automotive mixed-model assembly line. In their case, inefficiencies had to be corrected and performance indicators (such as the number of stations opened, average workload, and unbalancing level) were considered alongside practical zoning constraints. In the sequence, a novel multi-objective GA was developed by Yang, Gao, and Sun (2013) to solve mixed-model ALRPs with seasonal demand fluctuations. Under a cycle time constraint, vertical and horizontal balancing strategies were applied, as well as a third goal of minimising worker retraining costs.

Makssoud et al. (2015) proposed a new mathematical model with linearised objective function and constraints for counting the total number of performed task reassignments. Given the production requirement to meet the new demand, an exact solution method to solve an ALRP that aimed at the minimisation of tasks to be reassigned and the cost of opening extra stations was also developed. Later, Li (2017) has solved the cycle time minimisation version of a robotic ALRP with stochastic task learning. Leveraged on the classical SALOME (Scholl and

Klein 1997), an efficient non-constant task time rebalancing (ENCORE) algorithm is proposed and tested. Sancı and Azizoğlu (2017) considered the need to rebalance the line due to disruptions caused by station breakdowns or shutdowns. Trade-offs between cycle time (efficiency measure) and task reassignment (stability measure) were considered in an MILP-based algorithm. In addition, a B&B algorithm was implemented, with experimental results showing its superiority in terms of solution quality and computational processing time.

More recently, Sikora, Lopes, and Magatão (2017) formulated and applied an MILP model to solve the Travelling Worker Assembly Line Balancing Problem (TWALBP) in a situation that a rebalancing process was required to correct inefficiencies. By considering several practical extensions from real case scenarios such as human/robotic workers, assignment restrictions, zoning constraints, and automatic/common tasks, the model decides on task assignments and routing for every worker on a given layout. These (travelling) workers were allowed to perform or assist with tasks at more than one station. Trade-offs between assignment flexibility and movement times were analysed. Conceptually, this type of rebalancing process is evaluated in this paper with the addition of the highly complicating aspects of workforce heterogeneity. One of these real case scenarios, in which the assembly line under analysis employs more stations than workers, is herein explored in Section 6 as part of the case study. Finally, Belassiria et al. (2018) developed a GA hybridised with a priority rule-based heuristic procedure to tackle the ALRP. Their algorithm had the maximisation of line efficiency and workload balance as well as the minimisation of total cost as objectives. Nevertheless, none of these works took into account a temporary demand disruption scenario in which workforce had to be conserved during such period.

2.3. Integrating ALWABP and ALRP

Based on this literature review, to the best of our knowledge, no past research focussed on the problem of rebalancing assembly lines while considering full heterogeneity in task execution times. The closest study in the ALRP literature that considered heterogeneity was developed by Corominas, Pastor, and Plans (2008). However, workers' heterogeneity was limited to temporary unskilled workers and speed levels: a worker was always faster or slower than another. In this paper, we address this gap by proposing the ALWARP, in which task execution times are defined per worker.

Another difference between the ALWARP and the problem studied by Corominas, Pastor, and Plans (2008) is that they faced their rebalancing problem focussing on the supply of a high-season demand. We, on the other hand, consider situations of plummeting demand and focus on the maintenance of the current workforce at the lowest expense. Moreover, our model also incorporates the eventual need for social distancing measures in the face of restrictions imposed by the COVID-19 pandemic. In the following section, we explicit this problem description.

3. Problem statement

In order to illustrate the characteristics and the elements of an ALWARP under demand disruption scenarios, a sample instance is used in Figure 1.

A set of tasks (N) must be performed in order to assemble a product. Tasks are indivisible and must respect precedence relations in their execution order. A precedence graph containing task indexes (nodes from 1 to 11), as well as a SALBP solution with a defined cycle time that meets the initial demand (C^*) are presented at the top of Figure 1. In this scenario, the company uses an optimal SALBP-1 configuration (with five stations/workers, $m^* = 5$) for a known stable demand: the sum of these task processing times assigned to the same station/worker must not exceed the defined cycle time (C^*).

At the moment that this assembly line was implemented, the workforce could be considered homogeneous. Nonetheless, due to initial training and intensive repetition, these workers specialised in a subgroup of this task set. Changing their worker-task assignment at this point would slow down the line's pace. Therefore, depending on the heterogeneous worker w, each task i has a different deterministic processing time (t_{wi}) to be completed. Job rotation schedules alleviate this effect, but they are routinely conducted between neighbouring stations, thus the difficulty often remains if workers have to perform tasks originally assigned to stations far from their own. For instance, in Figure 1, worker 'a' in the first station is specialised on tasks 1 and 2, whereas more familiar with tasks 3, 4, and 5 (performed just beside it) than with task 11 (performed in the last station).

Under the demand disruption scenario caused by a pandemic, the initially defined cycle time (C^*) must be proportionally increased to cope with the necessary production rate. For example, if the initial cycle time to meet the demand had to be $C^* = 10$ and now the demand has fallen 50%, then the current cycle time limit will be $\overline{C} = 20$. Similarly, if the demand drops 90%, the new cycle time limit will be $\overline{C} = 100$.

Additionally, we consider social distancing rules to be applied to the line in order to follow health organisation recommendations, i.e. allowing extra physical space

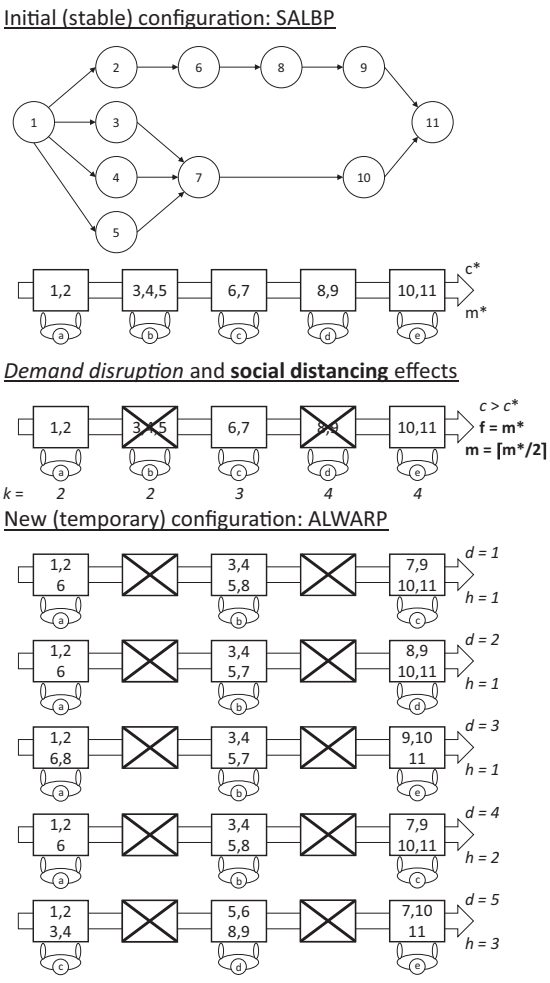


Figure 1. Illustrative example solution for the ALWARP. Figure 1 Alt Text: Illustrative example solution for the ALWARP. Workers are represented by letters *a* to *e*. Each configuration has an associated day of the week, $d = 1, \ldots, 5$ and shift type, $h = 1 \ldots 3$. Demand disruption and social distancing effects are highlighted in *italics* and **bold**, respectively.

Note: Figure 1 Long Description: An illustrative solution for the ALWARP is shown. The initial (stable) configuration is a SALBP solution. At the top, a precedence graph with eleven nodes is depicted along with its SALBP-1 solution below it. Each worker is represented by letters from a to e in a balanced solution for the given cycle time. Demand disruption and social distancing effects are highlighted in italics and bold, respectively. At the bottom, the new (temporary) configuration is an ALWARP solution. A rebalanced solution for each day of the work week is presented, with three workers along the line instead of five. The remaining stations between them are crossed to follow social distancing measures.

between workers. A possible approach to reduce contagion risk is creating clusters of employees. This approach has been adopted in practice to avoid infection spreading between worker groups. Still, it comes with operational limitations if the initial number of workers and their individual capabilities imply that some cluster divisions might operate with a further reduced workforce or may not form enough feasible groups.

We adopt an alternative approach by keeping the full assembly line structure while blocking out intermediary stations, allowing a physical gap between operating stations. In the case of an odd number of stations, as in Figure 1, this is immediately carried out by eliminating even-numbered stations. If the initial number of stations were an even number, an additional station would eventually be blocked to avoid any two workers operating in adjacent stations.

In order to comply with the social and economic (laying-off and hiring costs) issues described in Section 1, each worker must be hired in at least one period d of the considered time-horizon. For each period, a shift option h must be chosen to define its working hours' duration. In Figure 1, a work-week with five business days and three shift options (8, 6, and 4 h) were considered. Notice that all workers (from 'a' to 'e') are hired at least once a week with different shift durations (h from 1 to 3) in the example given by Figure 1. Furthermore, different workload reduction rates k were applied to each worker, which explains the more vigorous use of some workers over the others. Suppose a regular work-week containing 40 h (8 h a day) and four reduction possibilities: 1 meaning no reduction and 2, 3, and 4 respectively meaning 25%, 50%, and 70% reduction in workload. In this example, workers 'a' and 'b' had only 25% of their workload reduced, 'c' had 50%, whereas 'd' and 'e' had the most considerable 70% reduction. None of them had to be dismissed.

Nonetheless, these social advantages come with two drawbacks: (i) the possibility of a slenderly increased cost in salaries and (ii) the computational burden in solving the problem. For the studied problem, it is assumed that the balancing decision is a temporary short-term plan due to the uncertainty associated with the context. Workers may be heterogeneous (Miralles et al. 2008), but qualification levels do not hierarchically rank them to receive distinct wages (Sungur and Yavuz 2015). Thus, it is also considered that the cost per hour of any worker is identical. Hence, the mathematical model's primary objective (see Section 4.1) is to minimise the required working hours among employees. Even though workload reductions and shift options with different durations can be selected, the exigence of hiring each worker at least once a week must be satisfied. Therefore, the total contracted hours may not be used for all workers, with conceivable payments for non-worked hours, leading to minor inefficiencies compared to the more simplistic approach. For the instance presented in Figure 1, for example, the total capacity for labour hours among workers is 104 h, but the shift selection and the number of available stations only allow them to work a maximum amount of 102 h, thus giving rise to two slack hours for worker 'c'. Secondly, whereas it is relatively easy to resolve a SALBP case with SALOME to contemplate the enlarged cycle time, it might be much harder to

simultaneously decide the assignment of tasks to workers, workers to stations and periods, as well as the selection of workload reduction and shift options to workers and periods.

As aforementioned, stations from assembly lines examined in this study are dedicated to the mass production of a single model of a unique product. Such stations are positioned sequentially in a straight, serial line. Only one work-piece can be processed at a given time in each station. These pieces move forward between stations within a cycle time to be known and fixed by a limit ($\overline{C}$ under demand disruption scenarios), whilst their transportation times between stations are unheeded. For this study, costs related to moving tasks between stations are considered negligible compared to worker wages. As the line produces a single product, its pace and, consequently, its production rate are exclusively determined by the most loaded station. Moreover, depending on the shift option selected for a given period, the cycle time must be pondered by the working hours duration to always respect the daily production requirements.

Ultimately, the problem herein proposed also has a social aspect integrated into it. Thus, letting some workers abide a 70% workload reduction while others only have a 25% reduction does not seem to be fair, and may indeed not be the ideal solution from a financial equity perspective. Hence, in order to avoid such discrepancies and, potentially, some internal conflict among employees, a modification of the proposed mathematical model is presented in Section 4.2. This new model still prioritises the minimisation of worked hours in the objective function, but it is now accompanied by a secondary goal of reducing overall workload irregularities.

4. Mathematical formulation

In this section, the developed MILP models to represent the ALWARP are described. Sections 4.1 outlines the main model to lay out the problem, whilst Section 4.2 shows the necessary changes in the objective function and implemented constraints to take workload reduction regularity into account. Section 4.3 describes post-optimisation procedures to improve the obtained primal bounds' quality. Finally, Section 4.4 discusses the proposed modelling approach's flexibility and how it can be adapted to cope with practical characteristics found in real-world assembly lines, such as that described in the industrial case study. The formulations' notation for parameter and sets are detailed in Table 1 and can be generalised for most assembly line layouts. Table 2 does the same for the variables, which, in turn, are generated by the model depending on the tuples defined in Table 1.

Table 1. Parameters and sets notation.

Parameter	Meaning
m	Maximum number of stations
f	Number of workers
p	Number of periods
q	Number of available shifts per period
g	Number of possible workload reductions
n	Number of tasks
$\overline{C}$	Cycle time limit
$\overline{L}$	Regular workload
l_h	Workload for shift h
r_k	Level of workload reduction k
t_{wi}	Task processing time of worker w to execute task i
Set	**Meaning**
S	Set of stations s; $S = \{1, \ldots, m\}$
W	Set of workers w; $W = \{1, \ldots, f\}$
D	Set of periods d; $D = \{1, \ldots, p\}$
H	Set of shifts h; $H = \{1, \ldots, q\}$
K	Set of possible working hours reductions k; $K = \{1, \ldots, g\}$
N	Set of tasks i; $N = \{1, \ldots, n\}$
R	Set of precedence relations between two tasks i and j: (i, j)
SW	Tuple of feasible Station-Worker elements
SD	Tuple of feasible Station-Period elements
SN	Tuple of feasible Station-Task elements
WD	Tuple of feasible Worker-Period elements
WK	Tuple of feasible Worker-Reduction elements
WN	Tuple of feasible Worker-Task elements
DH	Tuple of feasible Period-Shift elements
SWD	Tuple of feasible Station-Worker-Period elements
WDH	Tuple of feasible Station-Period-Task elements
$SWDN$	Tuple of feasible Station-Worker-Period-Task elements

Table 2. Variables notation.

Variable	Meaning
x_{swdi}	Station-worker-period-task assignment: set to 1 if task i is performed in station s by worker w on period d
y_{swd}	Station-worker-period assignment: set to 1 if station s is occupied by worker w on period d
a_{wdh}	Worker-period-shift assignment: set to 1 if worker w is employed on period d with shift h
b_{dh}	Period-shift selection: set to 1 if shift h is chosen on period d
z_{wk}	Working hours reduction options: set to 1 if reduction k is applied to worker w

4.1. MILP model

Using the above-introduced notation, we propose the following problem formulation:

$$\text{Minimise:} \quad \sum_{(w,k)\in WK} r_k \cdot z_{wk}, \tag{1}$$

subject to:

$$\sum_{(s,w)\in SW} x_{swdi} = 1, \quad \forall\, d \in D,\ i \in N, \tag{2}$$

$$\sum_{s\in S} y_{swd} \leq 1, \quad \forall\, (w,d) \in WD, \tag{3}$$

$$\sum_{w\in W} y_{swd} \leq 1, \quad \forall\, (s,d) \in SD, \tag{4}$$

$$\sum_{(s,d)\in SD} y_{swd} \geq 1, \quad \forall\, w \in W, \tag{5}$$

$$\sum_{h\in H} b_{dh} = 1, \quad \forall\, d \in D, \tag{6}$$

$$\sum_{k\in K} z_{wk} = 1, \quad \forall\, w \in W, \tag{7}$$

$$\sum_{(s,w)\in SW} s \cdot x_{swdi} \leq \sum_{(s,w)\in SW} s \cdot x_{swdj}, \quad \forall\, (i,j) \in R,\ d \in D, \tag{8}$$

$$\sum_{(w,i)\in WN} t_{wi} \cdot x_{swdi} \leq \overline{C} \cdot \sum_{h\in H} \frac{l_h}{\overline{L}/p} \cdot b_{dh}, \quad \forall\, (s,d) \in SD, \tag{9}$$

$$\sum_{i\in N} x_{swdi} \leq n \cdot y_{swd}, \quad \forall\, (s,w,d) \in SWD, \tag{10}$$

$$a_{wdh} \geq \sum_{s\in S} y_{swd} + b_{dh} - 1, \quad \forall\, (w,d,h) \in WDH, \tag{11}$$

$$\sum_{(d,h)\in DH} l_h \cdot a_{wdh} \leq \overline{L} \cdot \sum_{k\in K} r_k \cdot z_{wk}, \quad \forall\, w \in W, \tag{12}$$

$$x_{swdi} \in \{0,1\}, \quad \forall\, (s,w,d,i) \in SWDN, \tag{13}$$

$$y_{swd} \in \{0,1\}, \quad \forall\, (s,w,d) \in SWD, \tag{14}$$

$$a_{wdh} \in \{0,1\}, \quad \forall\, (w,d,h) \in WDH, \tag{15}$$

$$b_{dh} \in \{0,1\}, \quad \forall\, (d,h) \in DH, \tag{16}$$

$$z_{wk} \in \{0,1\}, \quad \forall\, (w,k) \in WK. \tag{17}$$

The objective function is given by Expression (1): the goal is to minimise the required hours for workers employed in the assembly line. The objective function value represents the cumulative wages required to implement the solution in terms of equivalent salaries paid for full-time employees. The model is subjected to Constraints (2)–(12). Task assignments are enforced by Constraints (2), ensuring that each task is assigned to a station and performed by a worker exactly once in each period. Constraints (3) and (4) are complementary: for each period, they respectively ensure that each worker is hired at most in one station and that each station is occupied by at most one worker. Worker assignment is finally conducted by Constraints (5), which guarantee that all workers must be hired at least once a week (or any time-frame). Similarly to the assignment of tasks, Constraints (6) state that only one shift option can be selected for each period, whereas Constraints (7) state that only one workload reduction option can be applied to each worker. Constraints (8) guarantee that the precedence relations are respected. As the actual processing time of a task depends on the worker who is performing it (x_{swdi}) and their ability (t_{wi}) to do so in the rebalancing process, Constraints (9) limit the total processing time of tasks for each station (and consequently worker) to be at most the cycle time limit ($\overline{C}$), weighted by the shift option chosen for that given period. Constraints (10) indicate that a task can only be assigned to a worker in a given station and period if such worker is also assigned to the referred station and period. For these last constraints, a bound on the number of tasks that can be executed by the worker in question is needed. The number of tasks in the line (n) is clearly a valid bound. Equations (11) attribute appropriate values to a_{wdh} variables: if a worker w is hired on a period d, and a shift h is elected on this same period d, then it is possible to infer that worker w is working on the shift option h for that period d. Equations (12) state that workers can only perform tasks up to their hired paid hours. In conjunction with the objective function, these constraints keep the total paid hours among workers in the line at a minimum. Finally, Constraints (18) below are symmetry-breaking valid inequalities and can be added to the model in order to reduce the size of the tree explored by branch-and-cut based algorithms.

$$\sum_{w\in W} y_{swd} \leq \sum_{w\in W} y_{(s-1)wd}, \quad \forall\, (s,d) \in SD \,|\, s > 1. \tag{18}$$

4.2. Workload reduction regularity

The main model (1)–(18) represents the ALWARP. Nonetheless, seeking a workload reduction regularity is, from a social perspective, a thoughtful addition. Table 3 displays the necessary extra variables to assess workload irregularities between workers with different approaches.

The objective function (1) is modified to consider the overall irregularity (Δ) as a secondary goal to be minimised as follows:

$$\text{Minimise}: \sum_{(w,k)\in WK} r_k \cdot z_{wk} + \Delta. \tag{19}$$

Table 3. Variables notation for workload reduction regularity.

Variable	Set	Domain	Meaning
α	–	$\mathbb{R}_+$	Average workload reduction
δ	–	$\mathbb{R}_+$	Absolute value of the maximal workload reduction difference between the most/least loaded worker and the average (α)
δ_w	$w \in W$	$\mathbb{R}_+$	Absolute value of the workload reduction difference between worker w and the average (α)
δ^+/δ^-	–	$\mathbb{R}_+$	Highest/lowest workload reduction among all workers
Δ	–	$\mathbb{R}_+$	Overall detected irregularity

We consider three different irregularity criteria: in Sections 4.2.1 and 4.2.2, we minimise the maximal workload reduction deviation from average and the sum of workload reduction deviations from average, respectively, with the average α computed in Constraint (20).

$$\alpha = \frac{\sum_{(w,k)\in WK} r_k \cdot z_{wk}}{f}. \tag{20}$$

In Section 4.2.3, in turn, we consider the maximal deviation on workload reduction between any two of the workers. Details for the three criteria are presented in the following.

4.2.1. Regularity 1: maximal deviation from average

Constraints (21)–(23) are added to the main model along with the new objective function to assess and minimise the highest deviation from the average workload reduction. For instance, if the average workload reduction and the maximal deviation from it were evaluated for the illustrative example presented in Figure 1, the values of $\alpha = \frac{0.75+0.75+0.5+0.3+0.3}{5} = 0.52$ and $\delta = \max\{|0.75 - \alpha|,\ |0.75 - \alpha|,\ |0.5 - \alpha|,\ |0.3 - \alpha|,\ |0.3 - \alpha|\} = \max\{0.23, 0.23, 0.02, 0.22, 0.22\} = 0.23$ would be found. Naturally, absolute-value and maximum functions must be linearised to be included in the MILP framework: Constraints (21) and (22) guarantee that δ receives the absolute value of the maximal deviation from average. Constraint (23) simply states that Δ equals δ for this workload reduction regularity option.

$$\delta \geq \sum_{(w,k)\in WK} r_k \cdot z_{wk} - \alpha, \quad \forall\, w \in W, \tag{21}$$

$$\delta \geq -\sum_{(w,k)\in WK} r_k \cdot z_{wk} + \alpha, \quad \forall\, w \in W, \tag{22}$$

$$\Delta = \delta. \tag{23}$$

4.2.2. Regularity 2: sum of deviations from average

For the second option, Constraints (24)–(26) are appended to the main model along with the new objective function to assess and minimise the sum of all deviations from the average workload reduction. For example, the measured deviations from the average workload reduction (same previous α) for each worker in the illustrative instance presented by Figure 1 is $\delta_w = \{0.23, 0.23, 0.02, 0.22, 0.22\}$. Once again, notice that the absolute-value function was linearised for each worker with Constraints (24) and (25). Consequently Δ assumes the sum of all deviations from the average ($0.23 + 0.23 + 0.02 + 0.22 + 0.22 = 0.92$) normalised by the number of employees in the workforce ($f = 5$) in Constraint (26), resulting in a value of $\Delta = 0.184$.

$$\delta_w \geq \sum_{(w,k)\in WK} r_k \cdot z_{wk} - \alpha, \quad \forall\, w \in W, \tag{24}$$

$$\delta_w \geq -\sum_{(w,k)\in WK} r_k \cdot z_{wk} + \alpha, \quad \forall\, w \in W, \tag{25}$$

$$\Delta = \frac{\sum_{w\in W} \delta_w}{f}. \tag{26}$$

4.2.3. Regularity 3: maximal deviation between workers

The last option adds Constraints (27)–(29) to the main model along with the new objective function to calculate and minimise the maximal deviations of workload reductions among all workers. Variables δ^+ and δ^- assess the highest and lowest weekly workload reduction within the workforce, respectively. For instance, the highest and lowest workload reduction values for the solution presented in Figure 1 would be $\delta^+ = 0.75$ and $\delta^- = 0.3$, which is found by Constraints (27) and (28). Therefore, the maximal workload deviation between workers, represented by Δ in Constraint (29), equals 0.45 for this instance.

$$\delta^+ \geq \sum_{(w,k)\in WK} r_k \cdot z_{wk}, \quad \forall\, w \in W, \tag{27}$$

$$\delta^- \leq \sum_{(w,k)\in WK} r_k \cdot z_{wk}, \quad \forall\, w \in W, \tag{28}$$

$$\Delta = \delta^+ - \delta^-. \tag{29}$$

4.3. Post-optimisation enhancement

Once a feasible solution is found with the original models, two enhancements are proposed to improve the quality of the obtained bounds, namely (i) a lower bounding procedure is presented and (ii) variable reduction techniques are implemented. The former uses a lower bound based on homogeneous workers in a SALBP configuration line for a single day, whereas the latter relies on the variable elimination concept introduced by Patterson and Albracht (1975) (i.e. earliest/latest station) and fewer workload reductions or shift options. Table 4 describes

Table 4. Parameters and sets notation for the post-optimisation phase.

Parameter	Meaning
t_i	Minimum task processing time of task i: $t_i = \min_{w\in W}\{t_{wi}\}$
E_i	Earliest station at which task i can occur
L_i	Latest station at which task i can occur
Set	**Meaning**
S_i	Set of all direct and indirect successor of task i
P_i	Set of all direct and indirect predecessor of task i

the required additional parameters and sets to provide efficient lower bounds and assignment restrictions for the problem.

Firstly, we can create a strong lower bound for the problem based on a SALBP perspective, in which a homogeneous worker is conceptualised. For each task, this worker gets the lowest task processing time (t_i) amongst the heterogeneous workers (t_{wi}). Next, we execute this simplified version of the problem to assess the number of full-time workers that would be required in the most optimistic case, where every task is assigned to the fastest worker. This lower bound value is fed to the main model, thus automatically excluding all impossibly promising nodes in the B&B procedure.

Secondly, we can use the problem's knowledge learnt from the main model's result to offer a decreased set of alternatives regarding workload reductions and shift options. Moreover, Patterson and Albracht (1975) proposed the implementation of the earliest and latest stations (E_i and L_i, respectively), which reduces the number of variables by limiting the task assignment to occur in a lower range of stations. Note that this process was proposed for the SALBP, thus Expressions (30) and (31) adapt this idea to the ALWARP by defining the earliest and latest possible stations for each task. Therefore, in absolute terms, this is five times more effective in eliminating variables for the problem at hand, since we are considering weekdays. In combination with the lower bounding procedure, these techniques help the main model to converge towards an optimal solution and prove it, as further studied in Section 5.5.

$$E_i = \left\lceil \frac{(t_i + \sum_{j \in P_i} t_j)}{\overline{C}} \right\rceil, \quad \forall\, i \in N, \tag{30}$$

$$L_i = m + 1 - \left\lceil \frac{(t_i + \sum_{j \in S_i} t_j)}{\overline{C}} \right\rceil, \quad \forall\, i \in N. \tag{31}$$

4.4. Practical extensions

Manufacturing industries facing demand disruptions may be willing to adapt to such a temporary situation without carrying out lay-offs. Given the diminished requirements on production rates, managers may aim to achieve the best possible cost reduction while conserving the entire workforce. This combination of problems is a notable realistic feature that could be widely employed under particular circumstances. Therefore, in order to contemplate more industrial scenarios, incorporating more practical extensions is a desirable modification.

The models herein presented focus on the core characteristics of assembly lines, but they are flexible enough to easily incorporate particular assembly line settings found in practice. Indeed, as the herein proposed MILP formulations' sets were modelled as tuples (e.g. *SN*, *SW*, *WN*, *WK*, and *WD*), several practical limitations can be readily incorporated: machine positioning or task zoning features could be restricted by making the tuple *SN* sparse. We can usually find this type of assignment restrictions in the ALRP literature (Boysen, Fliedner, and Scholl 2007; Scholl, Fliedner, and Boysen 2010; Sikora, Lopes, and Magatão 2017). For instance, if specialised equipment located at specific stations is necessary to perform a subset of tasks, then the models only create the valid task assignment variables for those stations. In such cases, Constraints (18) might over-restrict the problem and prune the optimal solution. Thus, they could not be used for this application.

During the pandemic scenario, reducing contagion risks can also be an important measure for managers to adopt. Separating the workforce into clusters is a direct approach to accomplish such a goal. Each cluster of employees works on interspersed days to prevent infection from spreading between groups. This proposition comes at the cost of limiting assignment possibilities and requires a few preprocessing strategies for the MILP model. Firstly, it is necessary to avoid assignment infeasibilities when isolating workers into specific clusters. This analysis can be conducted by inspecting the *WN* tuple to guarantee that at least one worker in each given cluster is capable of performing each task within the cycle time limit. Once all the combinations are assessed, managers can choose a pair of clusters that respects these conditions and inform their choice with the *WD* tuple, which will, in turn, exclude undesired combinations from a contagion-risk perspective.

Another interesting practical extension is to consider the presence of robots in the assembly process. They may coexist with humans in the same environment or form a whole dedicated robotic system. In any case, robots cannot be easily moved from one station to another, since they generally require outside wires or cabling and go through a complex installation process. Given that they are fixed to a station, the tuple *SW* can be created in a manner to only generate one possible allocation for each robot. Similarly, human workers can be disallowed to perform activities in those stations by merely having a sparse set. Analogously, the tuple *WN*'s utility can be expanded by also pondering infeasible assignments due to worker-task incompatibility for either human or robotic workers. For instance, humans might not be able to perform tasks that require the use of high mechanical power or geometric precision. In contrast, robots might be faster to execute operations, but unable to access some specific work-piece areas.

Additionally, when an assembly line has few robots among human workers and the objective is to minimise

wages, it is reasonable to keep the robots working every day at full capacity whilst just applying workload reductions to humans. As long as it is explicitly stated in the objective function, robotic workers do not contribute to any cost increase related to salaries. Limiting the tuples *WK* and *WD* to take this characteristic into account solves the issue. Moreover, workers may get paid different salaries due to higher qualification levels (Sungur and Yavuz 2015). In this case, a wage parameter to distinguish each individual employee can be added in the objective function. On the other hand, if a purely robotic assembly line is under analysis, it is coherent to rotate the robots in use to assure preventative maintenance schedules, to the extent that breakdowns caused by wear and tear irregularities can be prevented. In the latter case, the same model can be used with slimly distinct interpretations for the variables.

Finally, notice that these modifications are valid not only for the original model, but also for all versions of workload reduction regularity that were proposed. Likewise, we can implement the variable elimination technique presented in Section 4.3 with sparse sets *SN*, *WK*, and *DH*. The applicability of these practical extensions is demonstrated in Section 6. An industrial case study conducted in a truck cabin assembly line that possesses many of these characteristics is presented and solved.

5. Computational results

A comprehensive dataset for the studied problem is herein proposed. This section carries out a computational study based on instances contained in a well-known literature benchmark, which is available for download at < www.assembly-line-balancing.de >. Furthermore, an industrial case study presented by Sikora, Lopes, and Magatão (2017) is also evaluated in detail, separately in Section 6. The list of all instances is summarised in Table 5.

In Table 5, instances are addressed by their precedence graph (Case), number of tasks (n), cycle time to meet the initial demand (C^*), the number of heterogeneous workers (f, which is the same as the initial number of stations m^*, except for Sikora), the number of available stations after the implementation of social distancing protocols (m), the demand reduction scenario (DR), the number of shift options (#S), the number of reduction options in workload (#R), and the required cycle time to meet the new demand ($\overline{C}$).

A balanced assembly line operating under stable conditions is considered as the initial scenario. Except for the industrial case study, such balancing solutions are generated by SALOME (Scholl and Klein 1997) for the required cycle time to meet the demand (C^*), resulting in the optimal number of stations (m^*) and task assignment decisions for the initial scenario. Accordingly, the number of heterogeneous workers (f) must be the same as m^*, given that a SALBP instance is used as the baseline. The initial task assignment decisions define which subset of tasks a given worker is specialised in order to guarantee heterogeneity among workers. Therefore, the initial task processing time is sustained for them. The remaining tasks go through a duration increase penalty, whose values depend on how distant the worker was from the station in which the task used to be performed in the initial balancing solution. For the computational tests herein performed, a uniform distribution was used with $U[t, t \cdot \frac{X}{m^*}]$ to determine task times, being X the worker distance (in number of stations) from the original task placement. Other factors can easily be implemented instead to represent the specific scenario under study more realistically, such as task incompatibilities or qualification levels with different wages. Finally, due to social distancing protocols, the number of available stations (m) is limited to $\lceil \frac{m^*}{2} \rceil$, which ensures a gap between any pair of stations. Notice that Sikora, Lopes, and Magatão (2017) allowed the existence of more stations than workers, since the former was disassociated from the latter in their study considering travelling workers. However, such feature does not change the data generation process for this instance. One can use the balancing solution reported by the authors as the initial configuration for the assembly line and apply the same demand disruption, social distancing, and heterogeneity effects to it.

We are also interested in analysing the effect of severe demand disruption scenarios. For each case, three situations are evaluated: 50%, 75%, and 90% reductions in production requirements. These disruptions impact the cycle time limit ($\overline{C}$) as presented in Table 5's last column.

Table 5. Summary of dataset instances.

Case	n	C^*	$f(m^*)$	m	DR	#S	#R	$\overline{C}$
Mitchell	21	21	5	3	50%	2	4	42
					75%	2	5	84
					90%	3	6	210
Heskiaoff	28	216	5	3	50%	2	4	432
					75%	2	5	864
					90%	3	6	2160
Gunther	35	81	7	4	50%	2	4	162
					75%	2	5	324
					90%	3	6	810
Kilbridge	45	79	7	4	50%	2	4	158
					75%	2	5	316
					90%	3	6	790
Tonge	70	410	9	5	50%	2	4	820
					75%	2	5	1640
					90%	3	6	4100
Arcus	83	8998	9	5	50%	2	4	17,996
					75%	2	5	35,992
					90%	3	6	89,980
Sikora	81	215	10(15)[†]	8	50%	2	4	430

[†] $f = 10$ and $m^* = 15$, as reported in Sikora, Lopes, and Magatão (2017).

In combination with social distancing measures, mild demand disturbances would require a line length extension and possibly hiring temporary workers in order to forestall infeasible layouts. Shift options of 8 and 6 h are assumed, with an extra option of 4-hour shifts introduced to the worst demand reduction scenario (90%). The weekly regular labour hours is considered to be 40 h divided into 5 business days. Workload reduction options of 25%, 50%, 70%, 85%, and 90% are regarded. For the less critical scenario (50% of demand reduction), only the three first ones are included (i.e. 25%, 50%, and 70%). In contrast, the intermediate and most critical ones are granted one and two extra options (85% and 90%), respectively. Naturally, the possibility of not reducing any hours at all is present in all scenarios. Lastly, in order to equalise workload reductions among workers, the workload regularity strategies presented in Section 4.2 are tested for both benchmark and case study instances.

In summary, the tested dataset contains 18 instances from the benchmark plus the industrial case study and 4 regularity approaches (including the option of no regularity objectives), totalling 76 instances of different sizes and characteristics. For a more detailed breakdown of the instances data and results, both the dataset and individual solutions are made available as supplementary material to this paper.

The computational study is split into three parts. Firstly, the main model and its variants considering workload regularity metrics are applied to benchmark instances found in Table 5 (Sections 5.2 and 5.3) using the proposed formulations presented in Sections 4.1 and 4.2. Results are then compared to those that could have been obtained by a SALBP approach in terms of cost addition. The second part analyses possible trade-offs when resorting to some cost addition to accomplish more uniform workload reductions among workers (Section 5.4). The purpose of this comparison is to verify whether it is genuinely possible to acquire significant improvements in workload reduction regularities at minor expenses in the overall workforce salary cost.

Furthermore, Section 5.5 specifically targets large-size instances by attempting to solve them optimally. It exploits the acquired problem's knowledge from Section 5.2 along with lower bounding and variable reduction techniques explored in Section 4.3 in order to do so. Finally, we include the aforementioned practical extensions (Section 4.4) into the model to solve the ALWARP under demand disruption conditions for a truck cabin manufacturer (Section 6). We compare the original results to the worker regularity variations and discuss the final reassignment.

All instances were solved on Gurobi 8.1 (Gurobi Optimization 2019), which was employed as the universal solver. Preliminary tests were performed in order to tune the solver's running parameters to the proposed problem. Such parameter tuning procedure is described in Section 5.1. Four threads of a 64-bit Intel Core^{TM} i5-7200U CPU (2.5 GHz) with 8 GB 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 one hour for all cases.

5.1. Parameter tuning

This section describes how preliminary tests were conducted in order to choose and set the solver's parameters in the optimisation approach. The first half of the dataset presented in Table 5 is selected for the parameter tuning procedure. Initially, they are solved twice with default settings to mitigate variations due to random choices in the solver's heuristic procedures. After that, each instance is solved several times with different parameter configurations until the time limit of three hours is reached. Once again, each configuration is solved twice in order to increase the analysis'robustness. Finally, the best overall combination is chosen to solve the remaining cases.

According to these preliminary tests, setting the search focus and pre-solve level parameters to specific values seems to have the most consistent positive impacts for the ALWARP. The former parameter allows modifying the high-level solution strategy, while the latter controls the emphasis given in the solver's pre-solve phase. In our application, the best-suited configurations for these parameters are the ones that strike a search focus in the bound in combination with an aggressive pre-solve option. This improvement may be explained by the fact that aggressive pre-solve settings tend to find near-optimal primal bounds at the beginning of the search. On the other hand, focussing on bounds that used to move very slowly in the default configuration is an adequate overall solution strategy for the search procedure.

5.2. Main model

This section presents and discusses the results related to the main model when it is applied to the benchmark instances exhibited in Table 5 with default and tuned solver's configurations. In Table 6, #S stands for the number of shift options, #R for the number of workload reduction options, Cost is the objective function value for the cumulative wages to be paid (given by Expression (1) in equivalent full-time salaries), and s^* is the optimal SALBP cost for the instance's cycle time. We also present the relative cost addition in case the proposed workforce maintenance approach is used instead of

Table 6. Results for the main model with default and tuned optimisation parameters.

					Tuned				Default			
Case	DR	#S	#R	s^*	**Cost**	CA	Gap	CPU(s)	**Cost**	CA	Gap	CPU(s)
Mitchell	50%	2	4	3	**3.50**	16.7%	0.0%	1.6	**3.50**	16.7%	0.0%	6.7
	75%	2	5	1.5	**1.65**	10.0%	0.0%	15.7	**1.65**	10.0%	0.0%	16.9
	90%	3	6	0.75	**0.75**	0.0%	0.0%	6.6	**0.75**	0.0%	0.0%	8.7
Heskiaoff	50%	2	4	3	**3.30**	10.0%	0.0%	6.0	**3.30**	10.0%	0.0%	23.1
	75%	2	5	1.5	**1.50**	0.0%	0.0%	0.5	**1.50**	0.0%	0.0%	10.6
	90%	3	6	0.75	**0.75**	0.0%	0.0%	6.2	**0.75**	0.0%	0.0%	8.3
Gunther	50%	2	4	3	**4.40**	46.7%	0.0%	58.0	**4.40**	46.7%	0.0%	1077.7
	75%	2	5	2	**2.20**	10.0%	0.0%	1789.9	**2.20**	10.0%	18.2%	3600
	90%	3	6	0.75	**0.90**	20.0%	0.0%	72.3	**0.90**	20.0%	0.0%	268.9
Kilbridge	50%	2	4	4	**4.40**	10.0%	0.0%	974.5	**4.40**	10.0%	4.5%	3600
	75%	2	5	2	**2.25**	12.5%	0.0%	582.0	**2.25**	12.5%	17.8%	3600
	90%	3	6	0.75	**1.10**	46.7%	0.0%	96.4	**1.10**	46.7%	0.0%	129.3
Tonge	50%	2	4	5	**5.50**	10.0%	5.5%	3600	**5.50**	10.0%	5.5%	3600
	75%	2	5	3	**3.00**	0.0%	28.3%	3600	**3.60**	20.0%	40.3%	3600
	90%	3	6	1	**1.50**	50.0%	33.3%	3600	**1.50**	50.0%	40.0%	3600
Arcus	50%	2	4	5	**5.50**	10.0%	5.5%	3600	**5.60**	12.0%	10.7%	3600
	75%	2	5	3	**3.00**	0.0%	35.0%	3600	**3.00**	0.0%	40.0%	3600
	90%	3	6	1	**1.20**	20.0%	16.7%	3600	**1.50**	50.0%	33.3%	3600
Average	50%			3.83	**4.43**	17.2%	1.8%	1373.4	**4.45**	17.6%	3.5%	1984.6
	75%			2.17	**2.27**	5.4%	10.6%	1598.0	**2.37**	8.8%	19.4%	2404.6
	90%			0.83	**1.03**	22.8%	8.3%	1230.2	**1.08**	27.8%	12.2%	1269.2
	Overall			2.28	**2.58**	13.2%	6.9%	1400.5	**2.63**	15.6%	11.7%	1886.1

simply rebalancing the line as a SALBP (column CA). The last two columns indicate the gap obtained by Gurobi's MILP solver after one hour of computational time (Gap) and total computational processing time (CPU). Notice that the s^* columns' values can only be obtained with a reduced workforce in a SALBP, which implies that some employees would have to be dismissed. Moreover, fractional values of s^* mean that the desired production level could be achieved by just employing workers under a reduced workload shift.

For both the default and tuned configurations, feasible solutions could be obtained for all instances. However, optimality was reached in 9 and 12 out of 18 instances with the default and tuned parameters, respectively. The tuned configuration results outperformed the default ones in all instances in terms of solution quality, computational processing time, or integrality gap. Since this superiority was demonstrated in these tests, the following results (including the ones in Sections 5.3 and 6) exclusively refer to the tuned configuration outcomes.

Regarding salary costs, 5 instances presented the same cost as its SALBP counterpart, whilst the remaining ones yielded solutions with higher costs (positive CA relative values). As expected, there is a positive cost addition in most instances: an overall average increase of 13.2%. Most small and medium-size cases could be solved in a reasonable CPU time or left a small integrality gap at the end of the allowed computational time of one hour.

For most cases, instances in which DR is 75% seem to be exceptionally harder to solve, whereas CPU time for some cases with 50% DR is higher than its 90% counterpart. The combinatorial nature of the problem can explain this observation. A 90% DR likely leads to more obvious workload reductions, with solutions that reduce the working hours as much as allowed. Large-size instances showed to be more challenging, evidencing that the instance's core parameters (number of tasks, stations, and workers) play a bigger role in defining CPU time than the ALWARP specific parameters (demand rates, shift options, and workload reductions).

We note that even minor absolute deviations (LB) from the incumbent solution (UB) will appear to be relatively large when dealing with small numerical values. In order to address these cases with increased gaps, we use the problem's knowledge obtained thus far to further incorporate lower bounding and variable reduction techniques. Section 5.5 focuses on tackling these large-size instances.

5.3. Workload reduction regularity models

This section presents and discusses the results related to workload regularity models when applied to the benchmark instances of Table 5. Results are reported in the same fashion as in Section 5.2 for each workload regularity model, with the Reg column addition to stand for the workload reduction regularity strategy. Computational results for the three workload reduction regularity models are presented in Table 7.

Once again, feasible solutions could be obtained for all instances. However, optimality was reached in only 27 out of 54 instances. When it comes to salary costs, 13 instances presented the same cost as its SALBP counterpart, while the remaining ones granted solutions with higher costs. These positive cost additions were the most common result, with an overall average increase of 13.9%,

Table 7. Results for the workload reduction regularity models.

Case	DR	#S	#R	Reg	**Cost**	s^*	CA	Gap	CPU(s)
Mitchell	50%	2	4	1	**3.50**	3	16.7%	0.0%	2.1
				2	**3.50**		16.7%	0.0%	3.2
				3	**3.50**		16.7%	0.0%	2.0
	75%	2	5	1	**1.80**	1.5	20.0%	0.0%	260.0
				2	**1.65**		10.0%	0.0%	133.7
				3	**1.90**		26.7%	0.0%	167.8
	90%	3	6	1	**0.75**	0.75	0.0%	0.0%	6.6
				2	**0.75**		0.0%	0.0%	4.9
				3	**0.75**		0.0%	0.0%	2.0
Heskiaoff	50%	2	4	1	**3.30**	3	10.0%	0.0%	48.0
				2	**3.30**		10.0%	0.0%	18.3
				3	**3.75**		25.0%	0.0%	5.9
	75%	2	5	1	**1.50**	1.5	0.0%	0.0%	0.9
				2	**1.50**		0.0%	0.0%	0.9
				3	**1.50**		0.0%	0.0%	0.9
	90%	3	6	1	**0.75**	0.75	0.0%	0.0%	3.6
				2	**0.75**		0.0%	0.0%	5.5
				3	**0.75**		0.0%	0.0%	3.6
Gunther	50%	2	4	1	**4.40**	3	46.7%	8.7%	3600
				2	**4.50**		50.0%	0.0%	3425.2
				3	**4.50**		50.0%	13.3%	3600
	75%	2	5	1	**2.20**	2	10.0%	24.3%	3600
				2	**2.20**		10.0%	26.4%	3600
				3	**2.30**		15.0%	13.0%	3600
	90%	3	6	1	**0.90**	0.75	20.0%	0.0%	178.4
				2	**0.90**		20.0%	0.0%	142.5
				3	**0.90**		20.0%	0.0%	373.7
Kilbridge	50%	2	4	1	**4.40**	4	10.0%	0.0%	3531.2
				2	**4.40**		10.0%	7.0%	3600
				3	**4.50**		12.5%	0.0%	2879.9
	75%	2	5	1	**2.30**	2	15.0%	9.0%	3600
				2	**2.30**		15.0%	13.2%	3600
				3	**2.30**		15.0%	17.4%	3600
	90%	3	6	1	**1.10**	0.75	46.7%	0.0%	79.3
				2	**1.10**		46.7%	0.0%	116.9
				3	**1.10**		46.7%	0.0%	236.3
Tonge	50%	2	4	1	**5.50**	5	10.0%	4.9%	3600
				2	**5.50**		10.0%	6.9%	3600
				3	**5.75**		15.0%	7.0%	3600
	75%	2	5	1	**3.00**	3	0.0%	35.9%	3600
				2	**3.00**		0.0%	32.9%	3600
				3	**3.10**		3.3%	35.5%	3600
	90%	3	6	1	**1.50**	1	50.0%	42.2%	3600
				2	**1.50**		50.0%	35.3%	3600
				3	**1.50**		50.0%	43.3%	3600
Arcus	50%	2	4	1	**5.50**	5	10.0%	8.0%	3600
				2	**5.50**		10.0%	15.2%	3600
				3	**5.75**		15.0%	12.2%	3600
	75%	2	5	1	**3.00**	3	0.0%	42.4%	3600
				2	**3.00**		0.0%	43.8%	3600
				3	**3.10**		3.3%	41.9%	3600
	90%	3	6	1	**1.30**	1	30.0%	32.1%	3600
				2	**1.30**		30.0%	17.7%	3600
				3	**1.30**		30.0%	26.9%	3600
Average	50%			1	**4.43**	3.83	17.2%	3.6%	2396.9
				2	**4.45**		17.8%	4.8%	2374.5
				3	**4.63**		22.4%	5.4%	2281.3
	75%			1	**2.30**	2.17	7.5%	18.6%	2443.5
				2	**2.28**		5.8%	19.4%	2422.4
				3	**2.37**		10.6%	18.0%	2428.1
	90%			1	**1.05**	0.83	24.4%	12.4%	1244.6
				2	**1.05**		24.4%	8.8%	1244.9
				3	**1.05**		24.4%	11.7%	1302.6

13.8%, and 17.7% for regularity criteria 1, 2, and 3, respectively. Analogously, the overall average cost additions for each DR (50%, 75%, and 90%) were 17.5%, 6.8%, and 24.4%, respectively. These are slightly higher than the average cost increase found by the model without regularity constraints. Most small and medium-size cases could be solved to optimality with a considerable increase in the average CPU time compared to the main model. The trend on 75% DR instances being the most challenging ones stays unchanged for the dataset under analysis. Integrality gaps at the time limit were also increased for feasible solutions. Finally, the same explanations from the main model can be applied here for large-size instances.

This set of experiments demonstrates that directly taking workload reduction regularity into account generates a more complex problem. Trade-offs between these competitive objective functions (salary and regularity) are further analysed and discussed in Section 5.4.

5.4. Model comparison and discussion

In this section, results obtained from both main and workload regularity models are compared and their main differences exposed in Tables 8 and 9. New terms in the column headers mean, for each given instance, the highest/lowest workload reduction among all workers (δ^+/δ^-), the average workload reduction (α), and the decrease in the highest discrepancy between the

Table 8. Results and averages comparison between single-objective and workload reduction regularity models (part 1/2).

Case	DR	Reg	Cost	δ^+	δ^-	α	CA	ΔR
Mitchell	50%	–	3.50	50%	0%	30%	–	–
		1	3.50	50%	0%	30%	0.0%	0.0%
		2	3.50	50%	0%	30%	0.0%	0.0%
		3	3.50	50%	0%	30%	0.0%	0.0%
	75%	–	1.65	85%	25%	67%	–	–
		1	1.80	85%	50%	64%	9.1%	41.7%
		2	1.65	85%	25%	67%	0.0%	0.0%
		3	1.90	70%	50%	62%	15.2%	66.7%
	90%	–	0.75	85%	85%	85%	–	–
		1	0.75	85%	85%	85%	0.0%	0.0%
		2	0.75	85%	85%	85%	0.0%	0.0%
		3	0.75	85%	85%	85%	0.0%	0.0%
Heskiaoff	50%	–	3.30	70%	0%	34%	–	–
		1	3.30	70%	0%	34%	0.0%	0.0%
		2	3.30	70%	0%	34%	0.0%	0.0%
		3	3.75	25%	25%	25%	13.6%	100.0%
	75%	–	1.50	85%	25%	70%	–	–
		1	1.50	70%	70%	70%	0.0%	100.0%
		2	1.50	70%	70%	70%	0.0%	100.0%
		3	1.50	70%	70%	70%	0.0%	100.0%
	90%	–	0.75	85%	85%	85%	–	–
		1	0.75	85%	85%	85%	0.0%	0.0%
		2	0.75	85%	85%	85%	0.0%	0.0%
		3	0.75	85%	85%	85%	0.0%	0.0%
Gunther	50%	–	4.40	70%	0%	37%	–	–
		1	4.40	70%	0%	37%	0.0%	0.0%
		2	4.50	50%	0%	36%	2.3%	28.6%
		3	4.50	50%	0%	36%	2.3%	28.6%
	75%	–	2.20	85%	50%	69%	–	–
		1	2.20	85%	50%	69%	0.0%	0.0%
		2	2.20	85%	50%	69%	0.0%	0.0%
		3	2.30	70%	50%	67%	4.5%	42.9%
	90%	–	0.90	90%	70%	87%	–	–
		1	0.90	90%	70%	87%	0.0%	0.0%
		2	0.90	90%	70%	87%	0.0%	0.0%
		3	0.90	90%	70%	87%	0.0%	0.0%

Table 9. Results and averages comparison between single-objective and workload reduction regularity models (part 2/2).

Case	DR	Reg	Cost	δ^+	δ^-	α	CA	ΔR
Kilbridge	50%	–	4.40	70%	0%	37%	–	–
		1	4.40	70%	0%	37%	0.0%	0.0%
		2	4.40	70%	0%	37%	0.0%	0.0%
		3	4.50	50%	0%	36%	2.3%	28.6%
	75%	–	2.25	85%	25%	68%	–	–
		1	2.30	70%	50%	67%	2.2%	66.7%
		2	2.30	70%	50%	67%	2.2%	66.7%
		3	2.30	70%	50%	67%	2.2%	66.7%
	90%	–	1.10	90%	50%	84%	–	–
		1	1.10	90%	70%	84%	0.0%	50.0%
		2	1.10	90%	70%	84%	0.0%	50.0%
		3	1.10	90%	70%	84%	0.0%	50.0%
Tonge	50%	–	5.50	70%	0%	39%	–	–
		1	5.50	70%	0%	39%	0.0%	0.0%
		2	5.50	70%	0%	39%	0.0%	0.0%
		3	5.75	50%	0%	36%	4.5%	28.6%
	75%	–	3.00	85%	25%	67%	–	–
		1	3.00	85%	50%	67%	0.0%	41.7%
		2	3.00	85%	25%	67%	0.0%	0.0%
		3	3.10	70%	50%	66%	3.3%	66.7%
	90%	–	1.50	90%	70%	83%	–	–
		1	1.50	90%	70%	83%	0.0%	0.0%
		2	1.50	85%	70%	83%	0.0%	25.0%
		3	1.50	90%	70%	83%	0.0%	0.0%
Arcus	50%	–	5.50	70%	0%	39%	–	–
		1	5.50	70%	0%	39%	0.0%	0.0%
		2	5.50	70%	0%	39%	0.0%	0.0%
		3	5.75	50%	0%	36%	4.5%	28.6%
	75%	–	3.00	85%	50%	67%	–	–
		1	3.00	85%	50%	67%	0.0%	0.0%
		2	3.00	85%	25%	67%	0.0%	-71.4%
		3	3.10	70%	50%	66%	3.3%	42.9%
	90%	–	1.20	90%	70%	86%	–	–
		1	1.30	90%	70%	86%	8.3%	0.0%
		2	1.30	90%	70%	86%	8.3%	0.0%
		3	1.30	90%	70%	86%	8.3%	0.0%
Average	50%	–	4.43	67%	0%	36%	–	–
		1	4.43	67%	0%	36%	0.0%	0.0%
		2	4.45	63%	0%	36%	0.4%	4.8%
		3	4.63	46%	4%	33%	4.5%	35.7%
	75%	–	2.27	85%	33%	68%	–	–
		1	2.30	80%	53%	67%	1.9%	41.7%
		2	2.28	80%	41%	68%	0.4%	15.9%
		3	2.37	70%	53%	66%	4.8%	64.3%
	90%	–	1.03	88%	72%	85%	–	–
		1	1.05	88%	75%	85%	1.4%	8.3%
		2	1.05	88%	75%	85%	1.4%	12.5%
		3	1.05	88%	75%	85%	1.4%	8.3%

workload reduction applied to workers (ΔR) compared to the baseline solution obtained by the main model. Also, CA now indicates the relative cost increase when comparing original results obtained by the main model with the ones obtained by regularity models.

As anticipated, the original model (single objective, without any regularity objectives) obtained, on average, the lowest cost. Besides, as demand needs decrease, the average workload reduction increases. Furthermore, interesting trade-offs were verified for each workload reduction regularity approach: respectively for criteria 1, 2, and 3, the overall average cost additions were only 1.1%, 0.7%, and 3.6% with reference to the original result, with 16.7%, 11.0%, and 36.1% overall average reductions in the most significant difference among reductions applied to workers (i.e. $\delta^+ - \delta^-$). Similarly, the overall cost additions for each DR were 1.6%, 2.3%, and 1.4% in comparison to the main model result, with 13.5%, 40.6%, and 9.7% overall average reductions in the greatest $\delta^+ - \delta^-$ applied to workers.

As expected, Reg3 shows the highest deviation in both additional costs and irregularity minimisation, since the latter is explicitly evaluated in this version of the model. Nevertheless, the results in Tables 8 and 9 demonstrate that the other two criteria were also effective in smoothing the workload reduction among employees. These solutions were almost as efficient in terms of cost as the simplistic approach of laying-off some workers and maintaining a SALBP configuration with the reduced personnel. This outcome is due to the explicit consideration of the workforce's heterogeneity.

5.5. Improving the gap

The most difficult instances in the proposed dataset are herein explored (i.e. the Tonge and Arcus cases from Table 5). These instances were demonstrated to be challenging in Section 5.2, yielding large integrality gaps for all their parameter settings (Table 6). One of the dissatisfactions with the main model's inadequate performance when applied to large-size instances was the stagnation of lower bounds on low quality values. As most nodes in the searching process were offering unrealistic good solutions, it was hard for the solver to prune those branches and only focus the investigation on promising directions. Another difficulty the proposed model faces is the exponentially large number of variables it creates for the Tonge and Arcus cases, which can be partially solved by applying some variable elimination techniques. In order to confront these issues, we use the problem's knowledge acquired from testing the main model to eliminate assignment possibilities, workload reductions, and shift options. These modifications improve our formulation in a post-optimisation phase (Section 4.3).

Lastly, assembly line rebalancing is a long-term strategy. Hence, it is not a problem extending the CPU time to 2 h for the computational tests concerning difficult cases. Table 10 displays the results obtained for large-size instances (Tonge and Arcus) when lower bounding and variable reduction techniques are applied to the main model (LB+VR). For comparison purposes, the tuned configuration version of the main model is also tested with a CPU time of 2 h. The new numbers of shift options (#S$^-$) and workload reductions (#R$^-$) come from the problem's knowledge, while LB* refers to the lower bound value obtained by the procedure described in Section 4.3.

Table 10. Results for large-size instances when integrating the problem's knowledge with lower bounding and variable reduction techniques.

					LB+VR			Tuned		
Case	DR	#S⁻	#R⁻	LB*	**Cost**	Gap	CPU	**Cost**	Gap	CPU
Tonge	50%	2	3	5	**5.50**	0.0%	3749.2	**5.50**	4.6%	7200
	75%	2	3	2.5	**3.00**	0.0%	3616.2	**3.00**	23.3%	7200
	90%	2	3	1	**1.50**	0.0%	3775.2	**1.50**	30.0%	7200
Arcus	50%	2	3	5	**5.50**	0.0%	3960.4	**5.50**	4.6%	7200
	75%	2	3	2.5	**3.00**	0.0%	4790.1	**3.00**	35.0%	7200
	90%	2	3	1	**1.20**	8.3%	7200	**1.20**	16.7%	7200

By using the proposed lower bounds, applying the variable reduction techniques, and increasing the computational processing time limit, we were able to optimally solve 5 out of 6 large-size instances and halve the gap in the remaining one. On the other hand, we also certified that simply increasing the CPU time does not yield the same level of improvement.These results evidence the post-optimisation phase effectiveness and the importance of exploiting relevant knowledge of the problem when solving complex or real-world cases, as further demonstrated in the following section. This advancement was made possible by the use of parameters from the initial feasible solutions in order to obtain a better lower bound and further reduce the problem's size with the proposed variable reduction techniques.

6. Industrial case study

An industrial case study is presented in detail and solved in this section. A real-world line found in an automotive manufacturer that produces truck cabins is altered into the instance herein presented (Sikora, Lopes, and Magatão 2017). The company is located on the outskirts of Curitiba, Brazil, and the variety of practical extensions observed within this case study motivated the adaptations conducted on the main model to efficiently solve the proposed problem inside this context (Section 4.4).

This truck cabin assembly line comprises 15 stations and 10 workers, of which 9 are humans and 1 is a robot placed at the end of the line. Two model variations account for a wide miscellanea of operations that can be gathered in 81 tasks. These models are very similar, with over 80% of tasks having identical task processing times. The simplest model has a small production rate (less than 15% of the product mix). The most intricate model, whose tasks are always more complex than or at least as complex as those in the former model, is responsible for the majority of the demand (approximately 85%). The disruptive scenario cuts down the demand for both the simple and complex models' production. Therefore, as in Sikora, Lopes, and Magatão (2017), the line is treated as a single-model assembly line for the more complex product variant. The system's optimised cycle time was initially reported to be 214.65 time units (requiring helper scheduling) or 215.3 time units (without scheduling). For this case study, the desired cycle time is rounded up to 430 time units, already considering a demand reduction of roughly 50%. Table 11 presents the adapted instance data used to solve the industrial case study. It shows which tasks were initially assigned to which workers and exhibits manual and robotic task processing times, precedence relations, and station-task assignment limitations.

The workers operating this line are responsible for welding together numerous cabin parts. In order to assemble such parts in the correct position, most of these welding procedures require specialised clipping stations to lock and hold the work-pieces. For this adapted instance, 24 out of the 81 tasks are considered to need specific equipment. These tasks are restricted to stations that contain such tools. As mentioned in Section 4.4, this feature prevents the usage of Constraints (18) in the optimisation model and modifies tuple *SN* to contemplate only feasible station-task assignments.

Furthermore, a worker handling heavy or large work-pieces might require assistance: positioning truck bench structures is an example. Such operations were classified as common tasks (Yazgan et al. 2011), since they needed more than one worker to perform them simultaneously, but were not characterised in the multi-manned category, where parallel work occurs within stations (Michels et al. 2019). Finally, since the number of stations was greater than the number of workers, they were allowed to travel along the line to execute tasks in different stations, as long as moving times were taken into account in the cycle time.

Social distancing requirements impose that 7 out of the 15 stations are closed to provide enough space between them. Any specialised equipment in removed stations is moved to their respective previous station. Common tasks are now mechanically assisted to avoid unnecessary contact and interaction. For the same reason, travelling workers were forbidden for this period. Notice that, although in this case the cost of moving tasks and equipment was neglected because they were inevitable and virtually insignificant when compared to wages, it does not mean that there was no cost at all in doing so. Moreover, these practical features can be adjusted to each particular case's objective function when deciding for the task and worker reassignment. Since the production layout is already in place and physical space is not a limiting constraint, there is no cost associated with station opening or closure (Makssoud et al. 2015).

As observed in Table 11, the majority of tasks are performed by manual labour and a few of them by a robot

Table 11. Adapted instance data for the industrial case study from Sikora, Lopes, and Magatão (2017).

Task index	Performed by worker†	Human processing time‡	Robot processing time	Precedes task(s)	Restricted to station(s)
1	1	27		2	
2	1	7		3, 4	
3	1	11	8		1, 6, 8
4	1	3		46	
5	1	69		13	
6	2	69		13	
7	1	4		8	
8	1	12		9, 10	
9	1	12	10		2, 4, 6, 8
10	1	21		11, 12	
11	1	25			
12	1	21			
13	3	38		14, 15	
14	3	24		16, 17	
15	4	26	22		3, 4, 6, 8
16	3	21		18, 19	
17	3	10	9		3, 4, 6, 8
18	3	16		20	
19	4	5	5		3, 4, 6, 8
20	3	25		24	
21	2	128		22, 23	
22	4	22	18		3, 4, 6, 7
23	2	8		24	
24	4	9		25	
25	4	8		26	
26	4	83		27, 28	
27	4	9	9		4, 6, 8
28	4	3		29, 30	
29	4	19		31	
30	4	22	20		4, 6, 8
31	4	5		32, 33	
32	8	3			4, 6
33	4	6		46	
34	5	157		35, 36	
35	5	31	27		4, 6, 7, 8
36	5	14		46	
37	6	157		38, 39	
38	6	31			5, 6, 7, 8
39	6	14		46	
40	7	63		41	
41	7	53		42, 43, 44	
42	7	34		45	
43	7	28	24		6, 8
44	10	63	37		6, 8
45	7	27		46	
46	8	9		47, 48, 49	
47	8	12	10		6, 7, 8
48	10	141	89		6, 7, 8
49	8	26		50	
50	8	3		51	
51	8	4		52	
52	8	5		53	
53	8	65		54, 55	
54	8	10		56, 57	
55	8	7	7		6, 7, 8
56	8	14		58, 59	
57	8	11	10		6, 7, 8
58	8	9		60	
59	8	30	27	61	6, 7, 8
60	8	10			
61	3	18		62, 63	
62	10	2	2		7, 8
63	3	8		64, 65	
64	3	18		66	
65	3	2	2		7, 8
66	3	11		67	
67	3	2		68, 69	
68	3	4		70	
69	3	4	4		7, 8
70	3	8		71	
71	3	5		72, 73, 74	
72	3	12		75, 76, 77, 78 79, 80, 81	7
73	10	55	39		7, 8
74	10	62	47		7, 8
75	9	109			
76	9	10			
77	9	16			
78	9	23			
79	9	41			
80	9	10			
81	9	5			

†Workers 1 to 9 are humans and 10 is a robot positioned at the last station.
‡Task processing times in normalised time units for the worker who usually performs the task.

located at the end of the line (station 8). An assembly line with mixed-labour may present complications. Robots cannot move between stations in different days (tuple *SWD*), their capacity of performing tasks might be distinct and should be mapped (tuple *WN*), and there is no reason for reducing its workload (tuple *WK*). In this line, welding procedures that unite work-pieces must be performed by human workers in specific stations with clipping tools to guarantee geometric precision. Finishing welding points used to reinforce the structure can be assigned to either manual or robotic workers (Michels et al. 2018). For this reason, the robotic worker is placed in the last station, since it can only process finishing tasks with a lower processing time.

Therefore, as mentioned at the beginning of this section, several practical extensions presented in Section 4.4 must be integrated into the model to solve the industrial case study. For instance, the objective function has to be modified to exclude robots from the paid workforce. There are two ways to do that: by creating subsets for human and robot workers, or robots can simply be referred to by the last indexes to the worker set W. As this study uses only one robot, which is obligatorily placed in the last station, it has been decided that the last element in W shall represent the robotic worker. Hence, the objective function of this industrial case study model reads:

$$\text{Minimise}: \sum_{(w,k)\in WK \mid w<10} r_k \cdot z_{wk}. \tag{32}$$

Likewise, Expressions (19)–(29) must be equally altered to contemplate this specific change. Additionally, as explained in Section 4.4, infeasible assignments and shift

Table 12. Results for the case study.

Case	m^*	f	m	DR	Reg	**Cost**	δ^+	δ^-	α	CA	ΔR	Gap	CPU(s)
Sikora	15	10	8	50%	–	**5.1**[†]	70%	25%	43%	–	–	24.5%	3600
					1	**5.30**	70%	25%	41%	3.9%	0.0%	29.4%	3600
					2	**5.00**	50%	25%	44%	−2.0%	44.4%	25.8%	3600
					3	**5.25**	50%	25%	42%	2.9%	44.4%	28.2%	3600

[†]By using the same lower bounding and variable reduction techniques presented in Section 4.3, we found an improved primal bound of 4.9 with only 6.1% gap in 7200 s.

options are modelled by changing the appropriate data tuples. Finally, Constraints (18) were removed as they are no longer valid under the new assumptions. Table 12 presents the obtained results. The column headers have the same meanings as previously described and the time limit is once again set to one hour for all cases.

The initial line was composed of 9 full-time human workers and 1 robot. Therefore, it can be observed in Table 12 that, depending on the workload reduction regularity strategy, half of the initial production rate could be achieved with 55.56% ($\frac{5}{9}$, Reg2) to 58.89% ($\frac{5.3}{9}$, Reg1) of the wage costs. The great advantage here is that no worker has to be dismissed to get such cost reduction. It also implies in future, long-term savings when it comes to lay-off and retraining costs.

Remarkably, the second strategy of the regularity model found a lower cost than the original model within the imposed time limit. The company would likely use this solution, since it also guarantees a fair workload reduction distribution among employees. In this case, workers 5 and 8 had their labour hours reduced by only 25%, whereas the others had their workload decreased by 50%. These worker reassignment solutions for the case study are also made available in the supplementary material. Lastly, at least for this case study, Reg2 demonstrated the best performance, which hints to the necessity of testing all proposed approaches when solving real-world instances. Ultimately, picking the best available solution afterwards to fit into managerial decisions is recommended.

7. Conclusion

The ALWARP with the objective of minimising the total worked hours under demand disruption scenarios has been addressed in this study. The existing literature on both separate problems – heterogeneous workforce assignment and assembly line rebalancing – has been reviewed. The main contribution of this article is the integration of the above-mentioned problems under circumstances that industries face a reduction in market demand. We propose a problem in which workload cutbacks must be applied to workers with different skills while avoiding worker discharges. The situation modelled and solved includes reducing the line size to impose temporary social distancing rules and cope with lower production rate requirements.

In order to formalise this practical situation, new MILP models are developed. These models ensure demand is met while minimising labour costs. Secondary objectives distribute the social cost of salaries reduction among the workforce. Realistic practical extensions are outlined and incorporated into the model to solve an industrial case in a truck cabin assembly line. The results obtained from the proposed dataset indicate that it is possible to achieve solutions that can keep wage costs as low as the straightforward approach of laying-off excessive employees whilst not dismissing a single worker. For the case study, the models were able to maintain half of the initial production rate (and the entire workforce employed) with a 55.56% fraction of the initial wage costs. Ultimately, the models were shown to be effective and flexible when dealing with demand disruption situations, providing overall good solutions for benchmark instances and applicable results for the practical case study, as well as valuable insights on workload reduction regularity strategies.

Developing a framework on how to conserve the workforce during temporary disruptive circumstances can be an important contribution to support manufacturing industries in demand compromised situations. Given that the rebalancing problem is not at the operational level but rather a decision to be made before the production resumes, time limits could be considerably increased to obtain optimal solutions. Nonetheless, as instance sizes increase, any exact method will eventually reach its limitation and resorting to heuristic methods will become the only feasible option. Still, the size of instances that can be tackled with the proposed models is sufficiently large to enable the analysis of the proposed approach's effects. Since the study of these new ways to deal with demand disruption focussing on avoiding lay-offs is the goal of this paper, we list the possibility of implementing heuristic solutions as a future research avenue.

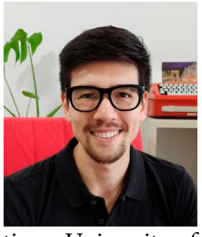
Furthermore, this research can be expanded to deal with different line layouts (U-line, parallel stations), task reassignment costs (moving machinery, retraining workers), and product variety (multi and mixed model lines).

Additionally, further research should focus on workforce flexibility to evaluate the impacts of allowing a limited amount of dismissals and sick leaves. Contrarily, demand increases on seasonal businesses or healthcare industries can be adapted into the herein proposed model by considering overtime on extended shifts, line length extension, and the use of unskilled temporary workers. Lastly, the counterpart of the proposed problem can also be examined as an alternative to find the minimum cycle time for a given workforce availability and the new configuration. Workload regularity and other practical extensions can be maintained in this twin version.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors



Adalberto Sato Michels holds a degree in Industrial Electrical Engineering from the Federal University of Technology – Paraná (UTFPR). He also holds a master's degree from the Graduate Program in Electrical and Computer Engineering (CPGEI) at UTFPR. He is currently a PhD candidate at the School of Mathematics and Statistics – University of Melbourne. His research involves applications of Operational Research techniques, focussing on Mixed-Integer Linear Programming, heuristics, and decomposition methods. His current projects include extensions of the Assembly Line Balancing Problem, namely multi-manned stations and workforce heterogeneity.



Alysson M. Costa is a Senior Lecturer in Operations Research at the School of Mathematics and Statistics – University of Melbourne. He is interested in theory and applications of Optimisation. Throughout his career, he has worked extensively with mixed integer programming (modelling and solution methods) applied to problems in different areas such as environmental water management, disaster relief operations, educational timetabling, crop rotation, assembly line balancing and city logistics, among others. He received his PhD in 2006 from HEC Montreal/University of Montreal – Canada. His thesis, titled 'Models and algorithms for two network design problems', received the Cecil Graham doctoral dissertation award from the Canadian Applied and Industrial Mathematics Society. Before that, he received MSc. and B.Eng. Degrees in Electrical Engineering from the State University of Campinas – Brazil.

Data Availability Statement

The data that support the findings of this study are openly available in Mendeley Data at http://doi.org/10.17632/tb5gdzdvhx.2.

ORCID

Adalberto Sato Michels http://orcid.org/0000-0003-2635-9700
Alysson M. Costa http://orcid.org/0000-0002-3135-793X

References

Araújo, Felipe F. B., Alysson M. Costa, and Cristóbal Miralles. 2012. "Two Extensions for the ALWABP: Parallel Stations and Collaborative Approach." *International Journal of Production Economics* 140 (1): 483–495.

Araújo, Felipe F. B., Alysson M. Costa, and Cristóbal Miralles. 2015. "Balancing Parallel Assembly Lines with Disabled Workers." *European Journal of Industrial Engineering* 9 (3): 344–365.

Australian Government. 2020a. "Coronavirus Economic Response Package Omnibus Act 2020." Accessed 25 March 2020. https://www.legislation.gov.au/Details/C2020C00139/696c6b47-9760-41b5-98fb-778719e735e4.

Australian Government. 2020b. "Coronavirus Economic Response Package (Payments and Benefits) Rules 2020." Accessed 25 March 2020. https://www.legislation.gov.au/Details/F2020C00401/b8811ae9-463d-4949-98ab-1f08054fc7f9.

Battaïa, Olga, and Alexandre Dolgui. 2013. "A Taxonomy of Line Balancing Problems and Their Solution Approaches." *International Journal of Production Economics* 142 (2): 259–277.

Bautista, Joaquín, and Jordi Pereira. 2009. "A Dynamic Programming Based Heuristic for the Assembly Line Balancing Problem." *European Journal of Operational Research* 194 (3): 787–794.

Baybars, İlker. 1986. "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem." *Management Science* 32 (8): 909–932.

Belassiria, Imad, Mohamed Mazouzi, Said ELfezazi, Anass Cherrafi, and Zakaria ELMaskaoui. 2018. "An Integrated Model for Assembly Line Re-balancing Problem." *International Journal of Production Research* 56 (16): 5324–5344.

Borba, Leonardo, and Marcus Ritt. 2014. "A Heuristic and a Branch-and-bound Algorithm for the Assembly Line Worker Assignment and Balancing Problem." *Computers and Operations Research* 45: 87–96.

Boysen, Nils, Malte Fliedner, and Armin Scholl. 2007. "A Classification of Assembly Line Balancing Problems." *European Journal of Operational Research* 183 (2): 674–693.

Brazilian Ministry of Economy. 2020. "Medida Provisória 936/2020." Accessed 1 April 2020. https://www.planalto.gov.br/ccivil-03/-ato2019-2022/2020/mpv/mpv936.htm.

Buxey, G. M. 1974. "Assembly Line Balancing with Multiple Stations." *Management Science* 20: 1010–1021.

Corominas, Albert, Rafael Pastor, and Joan Plans. 2008. "Balancing Assembly Line with Skilled and Unskilled Workers." *Omega* 36 (6): 1126–1132.

Cortez, Pâmela M. C., and Alysson M. Costa. 2015. "Sequencing Mixed-model Assembly Lines Operating with a Heterogeneous Workforce." *International Journal of Production Research* 53 (11): 3419–3432.

Costa, Alysson M., and Cristóbal Miralles. 2009. "Job Rotation in Assembly Lines Employing Disabled Workers." *International Journal of Production Economics* 120 (2): 625–632.

Falkenauer, Emanuel. 2005. "Line Balancing in the Real World." In *Proceedings of the International Conference on Product Lifecycle Management PLM 05*, Lyon, 360–370.

Gamberini, R., E. Gebennini, A. Grassi, and A. Regattieri. 2009. "A Multiple Single-pass Heuristic Algorithm Solving the Stochastic Assembly Line Rebalancing Problem." *International Journal of Production Research* 47 (8): 2141–2164.

Gamberini, Rita, Andrea Grassi, and Bianca Rimini. 2006. "A New Multi-objective Heuristic Algorithm for Solving the Stochastic Assembly Line Re-balancing Problem." *International Journal of Production Economics* 102 (2): 226–243.

Gurobi Optimization. 2019. "Gurobi Optimizer reference manual."

Hui, David S., Esam I. Azhar, Tariq A. Madani, Francine Ntoumi, Richard Kock, Osman Dar, and Giuseppe Ippolito, et al. 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.

Kramer, Raphael, Mauro Dell'Amico, and Manuel Iori. 2017. "A Batching-move Iterated Local Search Algorithm for the Bin Packing Problem with Generalized Precedence Constraints." *International Journal of Production Research* 55 (21): 6288–6304.

Li, Yuchen. 2017. "The Type-II Assembly Line Rebalancing Problem Considering Stochastic Task Learning." *International Journal of Production Research* 55 (24): 7334–7355.

Makssoud, Fatme, Olga Battaïa, Alexandre Dolgui, Khumbalani Mpofu, and Olayinka Olabanji. 2015. "Re-balancing Problem for Assembly Lines: New Mathematical Model and Exact Solution Method." *Assembly Automation* 35 (1): 16–21.

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 (Feb.): 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.

Miralles, Cristóbal, José P. García-Sabater, Carlos Andrés, and Manuel Cardós. 2008. "Branch and Bound Procedures for Solving the Assembly Line Worker Assignment and Balancing Problem: Application to Sheltered Work Centres for Disabled." *Discrete Applied Mathematics* 156 (3): 352–367.

Moreira, Mayron César O., Cristóbal Miralles, and Alysson M. Costa. 2015. "Model and Heuristics for the Assembly Line Worker Integration and Balancing Problem." *Computers & Operations Research*54: 64–73.

Moreira, Mayron César O., Rafael Pastor, Alysson M. Costa, and Cristóbal Miralles. 2017. "The Multi-objective Assembly Line Worker Integration and Balancing Problem of Type-2." *Computers & Operations Research* 82: 114–125.

Moreira, Mayron César O., Marcus Ritt, Alysson M. Costa, and Antonio A. Chaves. 2012. "Simple Heuristics for the Assembly Line Worker Assignment and Balancing Problem." *Journal of Heuristics* 18 (3): 505–524.

Morrison, David R., Edward C. Sewell, and Sheldon H. Jacobson. 2014. "An Application of the Branch, Bound, and Remember Algorithm to a New Simple Assembly Line Balancing Dataset." *European Journal of Operational Research* 236 (2): 403–409.

Mutlu, Özcan, Olcay Polat, and Aliye Ayca Supciller. 2013. "An Iterative Genetic Algorithm for the Assembly Line Worker Assignment and Balancing Problem of Type-II." *Computers and Operations Research* 40 (1): 418–426.

Oksuz, Mehmet Kursat, Kadir Buyukozkan, and Sule Itir Satoglu. 2017. "U-shaped Assembly Line Worker Assignment and Balancing Problem: A Mathematical Model and Two Meta-heuristics." *Computers and Industrial Engineering* 112: 246–263.

Oliveira, F. S., K. Vittori, R. M. O. Russel, and X. L. Travassos. 2012. "Mixed Assembly Line Rebalancing: A Binary Integer Approach Applied to Real World Problems in the Automotive Industry." *International Journal of Automotive Technology* 13 (6): 933–940.

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.

Pape, T. 2015. "Heuristics and Lower Bounds for the Simple Assembly Line Balancing Problem Type 1: Overview, Computational Tests and Improvements." *European Journal of Operational Research* 240: 32–42.

Parliament of the United Kingdom. 2020. "Coronavirus Act 2020 (c. 7)." Accessed 25 March 2020. http://www.legislation.gov.uk/ukpga/2020/7/enacted/data.pdf.

Patterson, James H., and Joseph J. Albracht. 1975. "Assembly-Line Balancing: Zero-One Programming with Fibonacci Search." *Operations Research* 23 (1): 166–172.

Polat, Olcay, Can B. Kalayci, Özcan Mutlu, and Surendra M. Gupta. 2016. "A Two-phase Variable Neighbourhood Search Algorithm for Assembly Line Worker Assignment and Balancing Problem Type-II: An Industrial Case Study." *International Journal of Production Research* 54 (3): 722–741.

Ramezanian, Reza, and Abdullah Ezzatpanah. 2015. "Modeling and Solving Multi-objective Mixed-model Assembly Line Balancing and Worker Assignment Problem." *Computers and Industrial Engineering* 87: 74–80.

Sancı, Ece, and Meral Azizoğlu. 2017. "Rebalancing the Assembly Lines: Exact Solution Approaches." *International Journal of Production Research* 55 (20): 5991–6010.

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.

Scholl, Armin, Malte Fliedner, and Nils Boysen. 2010. "ABSALOM: Balancing Assembly Lines with Assignment Restrictions." *European Journal of Operational Research* 200 (3): 688–701.

Scholl, Armin, and Robert Klein. 1997. "SALOME: A Bidirectional Branch-and-Bound Procedure for Assembly Line Balancing." *INFORMS Journal on Computing* 9 (4): 319–334.

Sewell, E. C., and S. H. Jacobson. 2012. "A Branch, Bound, and Remember Algorithm for the Simple Assembly Line Balancing Problem." *INFORMS Journal on Computing* 24 (3): 433–442.

Sikora, Celso Gustavo Stall, Thiago Cantos Lopes, and Leandro Magatão. 2017. "Traveling Worker Assembly Line (re)balancing Problem: Model, Reduction Techniques, and

Real Case Studies." *European Journal of Operational Research* 259: 949–971.

Sternatz, J. 2014. "Enhanced Multi-Hoffmann Heuristic for Efficiently Solving Real-world Assembly Line Balancing Problems in Automotive Industry." *European Journal of Operational Research* 235: 740–754.

Sungur, Banu, and Yasemin Yavuz. 2015. "Assembly Line Balancing with Hierarchical Worker Assignment." *Journal of Manufacturing Systems* 37: 290–298.

United States Congress. 2020. "H.R. 748: The Coronavirus Aid, Relief, and Economic Security Act." Accessed 27 March 2020. https://www.govinfo.gov/content/pkg/BILLS-116hr748enr/pdf/BILLS-116hr748enr.pdf.

Vilà, Mariona, and Jordi Pereira. 2014. "A Branch-and-bound Algorithm for Assembly Line Worker Assignment and Balancing Problems." *Computers and Operations Research* 44: 105–114.

Yang, Caijun, Jie Gao, and Linyan Sun. 2013. "A Multi-objective Genetic Algorithm for Mixed-model Assembly Line Rebalancing." *Computers and Industrial Engineering* 65 (1): 109–116.

Yazgan, Harun Resit, Ismail Beypinar, Semra Boran, and Ceren Ocak. 2011. "A New Algorithm and Multi-Response Taguchi Method to Solve Line Balancing Problem in An Automotive Industry." *International Journal of Advanced Manufacturing Technology* 57: 379–392.