Int. J. Production Economics 242 (2021) 108292

Contents lists available at ScienceDirect

International Journal of Production Economics

journal homepage: www.elsevier.com/locate/ijpe

An analysis of task assignment and cycle times when robots are added to human-operated assembly lines, using mathematical programming models

Tamás Koltai [a,*], Imre Dimény [a], Viola Gallina [b], Alexander Gaal [b], Chiara Sepe [c]

[a] Department of Management and Business Economics, Budapest University of Technology and Economics, H-1521, Budapest, P.O.B. 91, Hungary
[b] Fraunhofer Austria Research GmbH, Theresianumgasse 27, A-1040, Austria
[c] University of Naples Federico II, P.le Tecchio, Napoli, 80125, Italy

ARTICLE INFO

Keywords:
Assembly line balancing
Operations management
Human-robot collaboration
Automation
Mathematical programming

ABSTRACT

Adding robots to a human-operated assembly line influences both the short- and long-term operation of the line. However, the effects of robots on assembly line capacity and on cycle time can only be studied if appropriate task assignment models are available. This paper shows how traditional assembly line balancing models can be changed in order to determine the optimal number of workstations and cycle time when robots with different technological capabilities are able to perform a predetermined set of tasks. The mathematical programming models for the following three cases are presented and analysed: i) only workers are assigned to the workstations; ii) either a worker or a robot is assigned to a workstation; iii) a robot and a worker are also assigned to specific workstations. The data of an assembly line producing power inverters is used to illustrate the proposed calculations. Both the assignment of tasks and the changes of cycle time are analysed within the AIMMS modelling environment. The computational characteristics of the proposed mathematical programming models are also examined and tested using benchmark problems. The models presented in this paper can assist operations management in making decisions relating to assembly line configuration.

1. Introduction

In recent years, automation has had an increasing role in manufacturing, while rapid advances in robotics and an increasing trend of the application of robots in production systems can be perceived. Industrial robots are transforming the economics of many manufacturing sectors. Robots are able to perform ergonomically stressful and repetitive tasks with high precision. The application of robots increases the safety level and reduces the risk of health problems as well (Helms et al., 2002; Weckenborg et al., 2019). The automation of assembly lines allows organisations to produce higher volumes at lower costs and standardised quality. Fully automated lines are mainly implemented wherever the work environment is unsafe to human beings (Boysen et al., 2008).

However, in many cases a complete automation of the assembly line is impossible or economically not viable for several reasons. For example, certain tasks have not been efficiently automated especially when smaller lot sizes are used (Krüger et al., 2009), and frequently robots are slower than human workers (Weckenborg et al., 2019). Where this is true, hybrid systems and human-robot collaboration can be the solution.

The International Organization for Standardization (ISO, 2011) defines human–robot collaboration (HRC) as an operation between a person and a robot while both share a common workspace. Besides the ISO definition, some authors emphasise other characteristics of the human-robot collaboration: human and robot working without physical separation (Ogorodnikova, 2007), sharing the same working place and time without physical barriers (Chen et al., 2011). The closest type of cooperation between worker and robot occurs, if tasks at the same workpiece are processed jointly (Helms et al., 2002). Krüger et al. (2009) suggests that the term "collaboration" between worker and robot can be referred to the execution of the same task jointly, but it can also imply the execution of different tasks on the same piece in parallel.

The benefits of a combined or hybrid production system, featuring both manual and automated parts, are discussed by several papers. Michalos et al. (2014) and Tsarouchi et al. (2016) report that significant

* Corresponding author. Department of Management and Business Economics, Budapest University of Technology and Economics, H-1521, Budapest, P.O.B. 91, Hungary.
E-mail addresses: koltai.tamas@gtk.bme.hu (T. Koltai), dimeny.imre@gtk.bme.hu (I. Dimény), viola.gallina@fraunhofer.at (V. Gallina), alexander.gaal@fraunhofer.at (A. Gaal), chi.sepe@studenti.unina.it (C. Sepe).

https://doi.org/10.1016/j.ijpe.2021.108292
Received 13 June 2020; Received in revised form 22 July 2021; Accepted 2 September 2021
Available online 6 September 2021
0925-5273/

increases in productivity and improvement in operator working conditions were achieved with hybrid systems. Earlier, Lotter and Wiendahl (2006) found that a reduction in cost per piece was obtained in comparison with a fully automated assembly line. Meanwhile, Krüger et al. (2009) emphasise that hybrid systems are a good compromise, when full automation is not possible either technologically or due to financial constraints.

The main novelty of this paper is that models are proposed for analysing changes of task assignment and cycle time when some degree of automation is implemented in an assembly line. Two possible approaches to the implementation of robots are presented. First, only one kind of resource, either a worker or a robot is assigned to each station. Next, the possibility of using both a worker and a robot on the same station is considered: they either work in parallel on different pieces, or they work sequentially on the same piece. The major technological constraints considered in this case are the set of tasks the robot can perform, the difference of robot's task times compared to worker's task times, and the restriction on working in parallel by the worker and the robot in specific cases. The outcomes of the two cases are then compared with the completely human operated assembly line. The proposed models are illustrated with the data of a production line where power inverters are assembled. The implications for management of the different configurations are discussed, and in addition, the computational performance of the models is compared with existing benchmark problems. Linear programming and constraint programming solutions for the proposed models are presented and the advantages of constraint programming over the LP solution are also discussed.

The remainder of this paper is organised as follows. In Section 2, a brief overview of the literature related to assembly line balancing is given. Section 3 outlines the mathematical formulation of the proposed models. In Section 4, the results of three different scenarios are compared using a real-life case for illustration. In Section 5, managerial implications of the different line configurations are discussed, and in Section 6, the computational performance of the proposed models is analysed. Finally, the conclusions of the presented research are summarised in Section 7.

2. Literature review

Assembly lines are flow-oriented production systems made up by a sequence of workstations arranged along a conveyor belt. In the "progressive assembly" manufacturing process, the parts flow from station to station and components are gradually assembled together to obtain the final product.

Assembly line balancing problems (ALBPs) arise when assembly lines have to be designed or redesigned and a proper assignment of tasks to workstations must be found, taking into account one or more optimisation criteria, in order to meet the required production rate within a specific timeframe. The related optimisation problems and possible strategies have been extensively discussed in the literature (see for example, Scholl, 1999; Thomopoulos, 2014). The different types of ALBPs – categorised based on their assumptions and objectives – are best described by Boysen et al. (2007).

An efficient allocation of tasks may increase profit, the production rate and workload smoothness, besides contributing to a decrease in operation costs, cycle time, idle time and the number of workers. Tasks cannot be allocated to stations arbitrarily. Cycle time constraints, precedence relations, and zoning conditions as well as technological and logical requirements may influence the optimal assignment. Despite these restrictions, many feasible solutions may exist for the allocation of tasks to stations and optimisation models can be used to find the best task assignment.

It is possible to formulate the ALBP as a mathematical programming model. Being the ALB problem NP-hard, several exact, heuristic and meta-heuristic approaches have been proposed during the past decades to solve the problem either optimally or else pragmatically, within a reasonable computational time. Early research in the field focused on formulating and solving the Simple Assembly Line balancing problem (SALBP), assuming there to be indivisible tasks, fixed maximum cycle time, deterministic operation times, production of just one homogeneous product, no assignment restrictions besides the precedence constraints and a serial layout with equally-equipped stations (Scholl and Becker, 2006b). In the literature, two versions of the SALBP are considered to represent the basic problems:

- SALBP-1 focuses on the minimisation of the number of stations (workers) for a fixed cycle time, thus leading to a decrease in costs,
- SALBP-2 focuses on the minimisation of the cycle time for a defined number of stations, thus leading to the increase of production quantity.

The ALB problem was introduced by Bryton (1954) and formulated as a linear programming (LP) model by Salveson (1955), while Bowman (1960) suggested to use two integer programming (IP) models that were later modified by White (1961) with the introduction of binary decision variables.

Over the past decades, significant algorithmic development has supported the solution of simple assembly line balancing problems. A detailed review of exact and heuristic solution procedures for SALBPs has been prepared by Scholl and Becker (2006b). Bukchin and Raviv (2018) compare constraint programming (CP) and mixed-integer linear programming (MILP) to solve ALB problems. Battaïa and Dolgui (2013) classified ALB problems, published between the years 2007 and 2012, based on input data modelling, constraints, and objective functions.

The assumptions of the simple assembly line balancing problems are quite restrictive; consequently, the focus of the literature shifted from SALBPs to the formulation of General assembly line balancing problems (GALBPs) (Scholl and Becker, 2006a). General assembly line balancing problems, as extended forms of SALBPs include for example mixed-model assembly line balancing (MMALBPs), two-sided assembly line balancing (2 S-ALBPs), parallel assembly line balancing (PALBPs), U-shaped assembly line balancing (USALBPs), stochastic operation times, ergonomic assembly line balancing problems (ErgoALB), multi-manned assembly line balancing (MALBP). However, notwithstanding that GALBPs more closely resemble problems arising in practice, their solution procedures in most cases are based on SALBP algorithms, or balancing can be reduced to a simple assembly line balancing problem (Scholl and Becker, 2006b; Gallo, 2013). Scholl et al. (2010) provided an overview of how basic assembly line balancing problems might be extended with different types of assignment restrictions, while Boysen et al. (2008) emphasise how these models can be applied to real-world scenarios.

Two-sided assembly line balancing was first analysed by Bartholdi (1993) and, several heuristics and meta-heuristics were proposed in order to discover the optimal task assignment. Some examples of solutions to the two-sided assembly line balancing problem were presented by Özcan and Toklu (2009) and Purnomo et al. (2013).

The novelty of placing two or more assembly lines in parallel was presented by Gökçen (2006). He proposed a binary integer-programming model as a way to balance parallel assembly lines and showed that positioning two straight lines in parallel can help to reduce the number of workstations. A review of papers containing various techniques for solving PALBPs is provided by Lusa (2008).

The Ergonomic assembly line balancing problem (ErgoALB) deals with the inclusion of the ergonomic aspects in assembly line balancing models and focuses on workers' health and safety, balancing the workload among the stations in order to safeguard operators' health and posture (Hignett and McAtamney, 2000), In the ergonomic assembly line balancing problem, the conditions of the workers lifting high weights (Waters et al., 1993), performing stressful and repetitive tasks (Occhipinti, 1998), vibration exposure (Finco et al., 2020a), rest allowance (Finco et al., 2020b) are taken into account. Several indexes and KPIs have been proposed to monitor various ergonomic aspects. Otto and Scholl (2011) provided a general overview of widespread ergonomics methods and described how to model them.

When line configurations are prepared, attention should also be paid to the kind of resource used as well. Some tasks require more competent workers; consequently, constraints formulating the capability of workers must also be added to the basic models. In a case analysed by Koltai and Tatai (2011), it was found that some workers might be incompatible with certain tasks and, therefore, a dual assignment of operators and tasks to stations was suggested. By dividing workers into low-skilled, high-skilled and special workers, the authors proposed a modelling framework that could be used to describe several practical problems.

The Multi-manned Assembly Line Balancing Problem (MALBP) allows the assignment of more than one operator with identical skills and equal processing times to each workstation. Reducing the length of the assembly line and the amount of throughput time is among the advantages of such a configuration (Dimitriadis, 2006). Assembly line worker assignment and balancing problem (ALWABP) introduced by Miralles et al. (2007) assumes a limited number of skilled workers who can execute tasks with different processing times and task-worker incompatibilities. A formulation of the MALBP with skilled workers (MALBP-SW) is presented by Giglio et al. (2017). Yilmaz and Yilmaz (2019) extends the MALBP model with various assignment constraints, including the distance constraints but omits the skill constraints.

With the introduction of automation in modern manufacturing systems, the dual problem of assigning tasks to process resources and resources to stations (with resource-specific task times) manifested itself as the "equipment selection problem". In this stream in the literature, allocation-dependent task times are considered, since different equipment types lead to unequal efficiency levels during the performance of tasks. According to Scholl and Becker (2006a), the equipment selection problem is equivalent to the problem of selecting workers whose task performance speed is different. Rekiek and Dolgui (2002) provided a state-of-the-art optimisation method for an assembly line which provided a solution for equipment selection as well.

Bukchin and Rubinovitz (2003) showed that the problem of assembly line design with parallel stations can be treated as a special case of the problem of equipment selection. Levitin et al. (2006) solved the equipment selection problem of assembly lines with robots, using a genetic algorithm.

In recent years, there has been an increasing trend towards using robots in production systems. Although robotic assembly line balancing (RALB) is not as popular as the classic ALB, several authors have focused on this topic in the past few years. As with the SALB problems, RALBPs are classified into two main groups: RALBP-I minimises the number of workstations, while RALBP-II minimises the cycle time.

Rubinovitz et al. (1993) presented a robotic assembly line balancing problem and proposed a heuristic algorithm for solving a RALBP-I of the equipment selection problem in the case of an automated line. Kim and Park (1995) solved a RALBP-I problem using a cutting plane algorithm. Several authors (see for example Mukund Nilakatan et al., 2015a; Cil et al., 2016) presented various techniques to minimise the cycle time in RALB-II problems. More detailed reviews of RALB literature can also be found in Borba et al. (2018).

As far as the authors know, very little attention has been paid in the literature to lines with stations where robots and workers perform tasks together. Using robots and workers on the same line and at the same station allows the execution of tasks of the same piece in parallel using multiple types of resources and different execution times. Weckenborg et al. (2019) developed a genetic algorithm for solving a human-collaborative assembly line balancing and scheduling problem (HRCALBSP). In their solution, collaborative tasks can be performed by workers and robots jointly and in parallel. The model, formulated as a mixed-integer program, provides both the assignment of collaborative robots to stations and the distribution of workload to workers and robots. When joint execution of tasks is not allowed, the robots could be considered workers with specific skills and could be described with a special formulation of the MALBP-SW.

Currently, human-robot collaboration on assembly lines is still in evolution. Cherubini et al. (2016) offer an overview of the ideas recently proposed for human-robot collaboration in assembly. Tsarouchi et al. (2017) propose a method for the allocation of tasks in a HRC assembly cell in the automotive industry. Two methodologies to assess the automation potential of manual tasks with human-robot collaboration technology in assembly lines were presented by Teiwes et al. (2016) and Malik and Bilberg (2019). Human-robot workplace design and the allocation of tasks to humans and robots within a human-robot environment is discussed by Dianatfar et al. (2019), while Nikolakis et al. (2018) present an approach towards assigning tasks to available resources. Hashemi-Petroodi et al. (2020) present a detailed survey of operation management issues related to hybrid human-robot collaborative manufacturing systems.

In the following parts of the paper a modelling framework is proposed for analysing the effect of robots when they are introduced into human-operated assembly lines.

3. Mathematical descriptions of the problems

In the problem presented by this paper it is assumed that full automation of the production line is not possible. Some degree of automation, however, can be implemented and the effect of this automation on the operational characteristics of the assembly line is therefore investigated. Three different cases of an assembly line balancing problem are analysed and compared. In case I, only workers are used. In case II and case III, different levels of automation are implemented. In case II, only one kind of resource, either a worker or a robot is applied at each station. Case III allows the operation of a worker and a robot within the same station. Two versions of the models are presented for each case similarly to the logic of the SALB models: one version to minimise the number of stations and another version to minimise the cycle time when the number of stations is predefined. The consecutive solution of the two versions results in a bi-objective optimisation model.

The common attributes of the three models are the following:

- Task times are deterministic for each type of resource separately.
- The precedence relations of tasks are defined.
- Due to operational characteristics, a distance restriction is required: tasks pertaining to the same predefined group of tasks must be allocated to the same or neighbouring stations.
- The availability of human and robotic processing of tasks is defined.
- Parallel execution of tasks with a common root predecessor by a robot and a worker is not possible, since such tasks use the same intermediate parts and when the tasks are executed in parallel, workers may be obstructed by the presence of a robot.

The common basis of the three cases examined in this paper is a basic model which minimises either the number of stations or the cycle time. These basic models are extensions of the SALBP-1 and SALBP-2 models. The objective of the extension is to use multiple workforce types with different abilities and consequently, different task times.

Case I: Balancing the line when only workers are assigned to workstations.

When only workers are applied, the constraint related to parallel execution of tasks by a worker and a robot can be omitted and the allocation of tasks is restricted only to workers. Since all feasible solutions of this model are also feasible for case III the resulting minimum number of workstations yields an upper bound of the minimal number of workstations when human and robotic operation on the same station is allowed.

Case II: Either a worker or a robot is assigned to a workstation.

Koltai and Tatai (2011) constructed a general framework to model skill requirements and skill conditions for assembly line balancing models. In their paper, low skill constraints determine workstations for workers who are able to perform only some simple tasks, and high skill constraints consider tasks which require above-average skills from workers. In our case, robots are considered as low skilled workers since they can perform only a limited number of tasks. The minimum number of low skilled workers (robots) can be set below the minimum number of stations obtained in case I. The question is whether the number of stations using workers could be decreased with the use of robots.

Case III: human and robotic operation at the same station is allowed

The third case can be considered as a human-robot collaboration (HRC) problem. Workers and robots can work in parallel on different intermediate parts or sequentially on the same piece. In the mathematical model for case III, workers and robots share the same workspace, and tasks must be scheduled within these single stations in order to respect precedence relations and avoid different tasks interfering with each other. Such interference could be caused by using the same intermediate part. To avoid this scenario, the model forbids the parallel execution of tasks with a common root predecessor. The model can be derived from the MALBP-SW model, taking into consideration the interference constraint and the distance constraint.

Table 1 contains the summary of notations used in the following part of the paper. Table 2 presents all the objective functions and constraints of the mathematical programming models that have been applied for the three different cases.

Objective function (1) and constraints (3), (4), (5), (6), (7) and (8) in

Table 1
Summary of notations.

Indices	
i,k	index of tasks $(1,\ldots,I)$,
j	index of workstations$(1,\ldots,J)$,
w	index of workforce types (worker, robot),
l	index of final tasks,
g	index of task groups.
Parameters	
I	number of tasks,
J	maximum number of workstations,
N^*	minimum number of stations (the result of the station number minimisation model)
t_{iw}	time necessary to perform task i using workforce type w (task time),
T_c	cycle time of the assembly line,
K	number of robots.
Sets	
W	set of workforce types containing two elements: worker and robot, $\{\mathrm{H},\mathrm{R}\}$,
L	set of final tasks, $i \in L$, if task i does not precede any other task,
P_i	set of indices of those tasks which must be finished before task i is started,
NA_w	set of tasks for which resource type w has no ability,
G	set of task groups.
Decision variables	
x_{ijw}	0-1 decision variable; if $x_{ijw}=1$ then task i is assigned to station j using workforce type w, otherwise $x_{ijw}=0$,
y_{ik}	0-1 decision variable; if $y_{ik}=1$ then task i and k are assigned to the same station and i is executed before k,
u_{jw}	0-1 decision variable; if $u_{jw}=1$ then station$j$ uses only workforce type w,
l_{jw}	0-1 decision variable; if $l_{jw}=1$ then workforce type w is used on station j,
s_i	continuous decision variable: start time of task i on the respective station,
e_i	continuous decision variable: end time of task i on the respective station,
N	objective function variable for the number of workstations,
T	objective function variable for the cycle time related to a given number of stations.

Table 2
Models for minimising the number of stations and cycle time.

Basic models		
$Min(N)$		(1)
$Min(T)$		(2)
$\sum_{j,w} x_{ijw} = 1$	$\forall\, i$	(3)
$\sum_{j,w} j(x_{ijw} - x_{kjw}) \geq 0$	$\forall\,(i,k)\|\, k \in P_i$	(4)
$\sum_{j,w} j(x_{kjw} - x_{ijw}) \leq 1$	$\forall (i,k)\|\exists\ g \in \mathrm{G};\ i,k\ \in g$	(5)
$x_{ijw} = 0$	$\forall (i,j,w)\|i \in NA_w$	(6)
$\sum_i x_{ijw} t_{iw} \leq T_c$	$\forall (j,w)$	(7)
$N \geq \sum_i j^* x_{ijw}$	$\forall (j,w)$	(8)
$T \geq \sum_i x_{ijw} t_{iw}$	$\forall (j,w)$	(9)
$\sum_{i,w} x_{ijw} = 0$	$\forall\, j > N^*$	(10)
Case I: additional constraints when only workers are used at the stations		
$\sum_j x_{ijR} = 0$	$\forall\, i$	(11)
Case II: additional constraints when workers and robots are used but in different station		
$\sum_{i \notin NA_w} x_{ijw} \leq I l_{jw}$	$\forall (j,w)$	(12)
$\sum_w l_{jw} \leq 1$	$\forall\, j$	(13)
$\sum_i x_{ijw} \geq l_{jw}$	$\forall (j,w)$	(14)
$\sum_j l_{jR} \geq K$		(15)
Case III: additional constraints when workers and robots are used at the same station		
$e_i \leq T_c$	$\forall\, i$	(16)
$\left(\sum_w x_{ijw}\right)\left(\sum_w x_{kjw}\right)\left(s_k - e_i\right) \geq 0$	$\forall\,(i,k,j)\|i \in P_k$	(17)
$\left(\sum_w x_{ijw}\right)\left(\sum_w x_{kjw}\right)\left(s_k - e_i\right)(s_i - e_k) \leq 0$	$\forall\,(i,k,j)\ \|P_i \cap P_k \neq \varnothing$	(18)
$e_i = s_i + \sum_{j,w} x_{ijw} t_{iw}$	$\forall\, i$	(19)
$e_i \geq 0; s_i \geq 0$	$\forall\, i$	(20)
$e_i \leq T$	$\forall\, i$	(21)

Table 2 form the model for minimising the number of stations. Objective function (2) with constraints (3), (4), (5), (6), (9) and (10) in Table 2 form the model for minimising the cycle time for a given number of stations.

The distance restriction is described by constraint (5). Constraint (6) excludes the allocation of tasks to workforce type which lacks the ability to execute it.

As a result of constraint (7), the cycle time resulting from the allocation of tasks to stations remains within the predefined cycle time (T_c). Constraint (8) is related to the objective function when the number of stations is minimised. When minimising the cycle time, constraint (9) is the constraint related to the objective function and constraint (10) restricts the allocation of tasks to stations numbered below the minimum number of stations.

All three models compared in this paper restrict the search space of the basic models, but to different extents. Using the same core model helps to explain and illustrate the differences between the three cases.

- Additional constraint when only workers are assigned to workstations (Case I).

The only constraint added to the basic model to minimise the number of stations and the cycle time is constraint (11) which restricts the allocation of tasks only to workers.

- Additional constraints when either a worker or a robot is assigned to a workstation (Case II).

In this case the basic model is extended with constraints (12), (13), (14) and (15). The binary decision variable l_{jw} determines whether workforce type w is used on station j or not. According to constraint (12), tasks can be allocated to certain (j,w) combination of station and workforce type only if $l_{jw}=1$. In this case, workforce type w (worker or robot) is assigned to station j. Due to constraint (13), stations can only use one workforce type. Constraint (14) ensures that a workforce type is allocated to a station only if a task with the same workforce type is also allocated to the same station. Constraint (15) sets the minimum number of stations with robots to K . Since only one type of workforce is used on a station, the parallel execution of tasks by a worker and a robot is not possible.

- Additional constraints when human and robotic operation at the same station is allowed (Case III).

Sequencing and scheduling tasks within a station requires the introduction of new decision variables. The basic model is extended with decision variables related to the start time (s_i) and end time (e_i) of the tasks measured from the start of the activities on the respective station. The end time of tasks is forced to be less than the cycle time by constraint (16). According to constraint (17), two tasks with a precedence relation on the same station must satisfy precedence constraints. A pair of tasks that have common root predecessors and are allocated to the same station must be sequenced, in order to avoid simultaneous execution. This requirement is expressed by constraint (18). Constraint (18) can be easily generalised by defining the arbitrary pairs of tasks that can be executed in parallel. As for constraint (19), the end time of tasks is equal to the start time plus the execution time. Non-negative start and end times are imposed by constraint (20). When minimising the cycle time constraint (21) must be added to keep the end time below the cycle time decision variable T. Constraints (17) and (18) are non-linear, a fact which limits the application of the model in practical settings.

To overcome the difficulties of solving non-linear models, two possibilities are hereby investigated. One of the possibilities is to transform the model into a mixed-integer linear programming (MILP) model with the linearisation of non-linear constraints. Table 3 presents the linear formulation of constraint (17) and (18) using additional binary variables (y_{ik}) related to the execution order of tasks allocated to the same station. The execution order of tasks within the station is determined by the start time of the tasks. When a task of a worker and a task of a robot should be started at the same time, the task of the worker is ordered first for technical reasons.

As for constraint (22), if execution of task i precedes task k on a certain station then execution of task k cannot precede the execution of task i and according to constraint (23), no task can be executed preceding itself. According to constraint (24), if two tasks with precedence relations are allocated to the same station, then their execution order must satisfy the corresponding precedence relation. According to constraint (25), if two tasks are allocated to the same station, an execution precedence relation must exist between the two tasks. According to constraint (26), if two tasks with precedence relations are allocated to the same station, then the start and end times of these tasks have to satisfy the corresponding precedence relation.

According to constraint (27) and (28), if the execution order for two tasks is set, then the end of the task with preceding execution has to be before the start of the other task in cases when they have common root predecessor task or are executed using the same workforce type.

As a consequence of the new additional constraints and variables, the computation time to find an optimal solution in case III is considerably higher compared to the computation time of the models for case I and II. The minimal number of workstations resulting from the case when only workers are used yields an upper bound for the case when both workers and robots are used. To reduce the computation time, the number of stations can be limited to the minimal number of stations when only workers are used.

Table 3
Constraints for the linearisation of the model for case III.

$y_{ik} + y_{ki} \leq 1$	$\forall(i,k)$	(22)
$y_{ii} = 0$	$\forall\, i$	(23)
$y_{ki} \geq \sum_w (x_{ijw} + x_{kjw}) - 1$	$\forall(i,k,j) \mid k \in P_i$	(24)
$y_{ik} + y_{ki} \geq \sum_w (x_{ijw} + x_{kjw}) - 1$	$\forall(i,k,j) \mid i \neq k$	(25)
$s_i \geq e_k - T_c\left(2 - \sum_w (x_{ijw} - x_{kjw})\right)$	$\forall(i,k,j) \mid k \in P_i$	(26)
$s_i \geq e_k - T_c(1 - y_{ki})$	$\forall(i,k) \mid P_i \cap P_k \neq \varnothing$	(27)
$s_i \geq e_k - T_c(3 - y_{ki} - x_{ijw} - x_{kjw})$	$\forall(i,k,j,w)$	(28)

The linearised model for case III provides a proper description of human-robot collaboration, but computational time still makes it difficult to use it for practical settings. As presented in the literature review, many alternative optimisation techniques and heuristics are available to solve general assembly line balancing problems. With the use of both worker and robot at the same station the problem has become more of a scheduling problem; constraint programming (CP) is considered a good alternative method to apply.

The basic idea of CP is to model the problem as a set of variables with domains and a set of constraints restricting the possible combinations of the variables' values. Constraint programming has a rich modelling language to represent complex real-world problems with discrete variables. In the above model, the start and end times of the tasks are the only non-discrete variables. Defining a base time unit, these variables could be considered discrete and the problem could be viewed as a resource-constrained scheduling problem with some additional constraints.

The activities to be scheduled are the tasks. The *start*, *end* and *presence* property of an activity used in CP formulations covers the variables s_i e_i, and x_{ijw}. Tasks using the same root piece cannot be processed in parallel by a worker and a robot due to possible interference. To implement constraint (18) into the CP model, a material for each root task and a material resource requirement for each activity based on its root tasks are defined. The nonlinear constraint (17) is kept as it is. With the use of a CP solver the solution time can be reduced to a level which permits the use of the model for practical problems.

The solution of the models defined for the three different cases requires a flexible mathematical modelling tool. Such a tool is provided by the AIMMS Prescriptive Analytics Platform, which is often applied for solving commercial optimisation problems (Roelofs and Bisschop, 2018). AIMMS offers a straightforward mathematical modelling environment and a wide range of available solvers including CPLEX to solve LP and MILP problems and CPOptimizer to solve CP problems. In this research AIMMS version 4.42 was used to create the required mathematical models, implement the algorithms and to create simple user interface. CPLEX version 12.7.1 was used to solve the generated LP and MILP problems and CPOptimizer version 12.6 was applied for the solution of the CP models.

4. Practical illustration of the presented models

The data of an assembly line producing power inverters are used to illustrate the proposed calculations. Power inverters usually consist of a large number of individual parts. Due to the nature of the individual parts, the assembly of these inverters is often characterised by pure manual assembly. Technology exists, however, to perform several assembly tasks by robots. Table 4 contains the task times for the workers and for the robots as well. Tasks with no defined task time for robots can only be performed by workers.

In the initial state of the factory the assembly line consisted of 8 stations and the targeted cycle time (T_c) was 60 s. The objective is to reduce the number of workstations without the deterioration of the cycle time.

It can be noted that tasks require longer execution time when processed by robots. The explanation of this contradiction is that for a robot it takes more time to analyse the piece and find the proper way to handle it and put it in the right position.

Fig. 1 shows the precedence relations of tasks. Tasks are denoted by letters starting from An up to AW. Two groups are defined for tasks that must be processed in the same or neighbouring station. Group 1 consists

Table 4
Task times of workers (t_{iH}) and robots (t_{iR}).

Task	A	B	C	D	E	F	G	H	I	J	K	L	M
t_{iH}	2.6	0.6	2.8	4.5	3.5	2.9	4.6	6.9	2.2	2.2	25.5	4.1	4.8
t_{iR}	3.9		4.2	6.75		4.35	6.9				38.25		
Task	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
t_{iH}	2.7	1.6	1.6	9.2	3.9	24.4	11.3	3.7	7.6	5.2	1.8	12.2	7.6
t_{iR}						36.6	16.95		11.4				11.4
Task	**AA**	**AB**	**AC**	**AD**	**AE**	**AF**	**AG**	**AH**	**AI**	**AJ**	**AK**	**AL**	**AM**
t_{iH}	2.5	1.8	9.2	2.2	30.3	1.6	1.6	2.4	12.9	7.6	1.3	1.3	2.8
t_{iR}	3.75		13.8			2.4	2.4		19.35				
Task	**AN**	**AO**	**AP**	**AQ**	**AR**	**AS**	**AT**	**AU**	**AV**	**AW**			
t_{iH}	1.6	17	1.1	1.7	6.1	1.1	5.1	4.5	2.2	2.8			
t_{iR}	2.4												

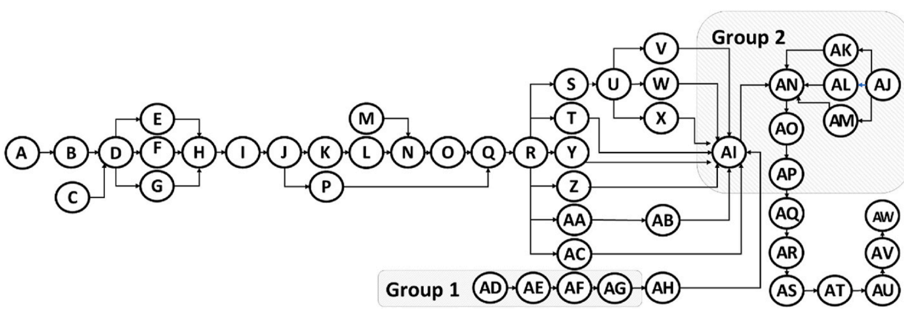

Fig. 1. Precedence relations between the tasks.

of tasks AD, AE, AF, AG and group 2 consists of tasks AI, AJ, AK, AL, AM, AN, AO, AP.

Figs. 2–5 show the Gantt charts of the analysed cases. Comparing the four Gantt charts, the change of cycle times, the allocation of task to stations, and the assignment of robots and workers to the different stations can be analysed.

When only workers are used, the minimal number of stations resulting from the model for case I is 5, and the minimal cycle time is 56.8 s. Fig. 2 shows the assignment of task to stations and the corresponding Gantt chart of this case.

In the case when robots are used, but not sharing stations with workers (case II), the minimum number of robots has to be predefined. When the minimum number of robots is set to 1 ($K = 1$) the minimum number of workstations is 6 with one of them being allocated to a robot, and the optimal cycle time is 50.3. Since not all tasks can be processed by robots, a completely robotic station cannot be created without increasing the number of stations. Fig. 3 presents a possible layout for this case. The figure shows that the robot is assigned to station 4 and performs tasks S and Z.

Increasing the minimum number of robots to two ($K = 2$) the minimum number of workstations remains 6, but in this case only 4 workers are required, while the optimal cycle time increases to 58.3. A possible allocation for this case is presented in Fig. 4. The figure shows that the robots are assigned to station 3 and 5.

In the case when workers and robots are allowed to work in parallel on the same station, a decrease in cycle time can be achieved, although the number of station cannot be decreased. The minimal value for the cycle time is 52 s. Fig. 5 shows a Gantt chart of a possible allocation of tasks related to the new optimal cycle time value, where each station is represented by two rows, one for the worker and one for the robot. In the solution presented in Fig. 5, station 1 and station 4 are using robots while stations 2, 3 and 5 have only workers.

This problem does not have a unique solution and the optimal cycle time can be achieved with different task allocations, processing orders

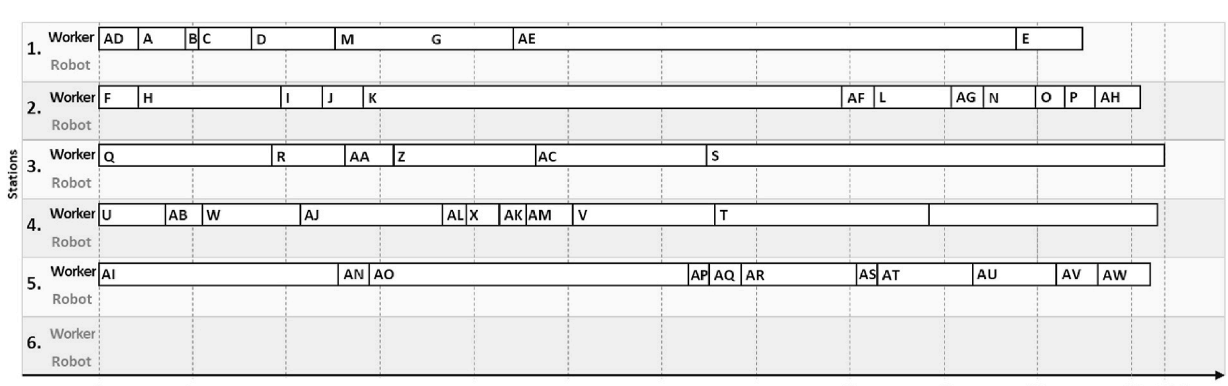

Fig. 2. Allocation of tasks to stations with optimal cycle time for case I.

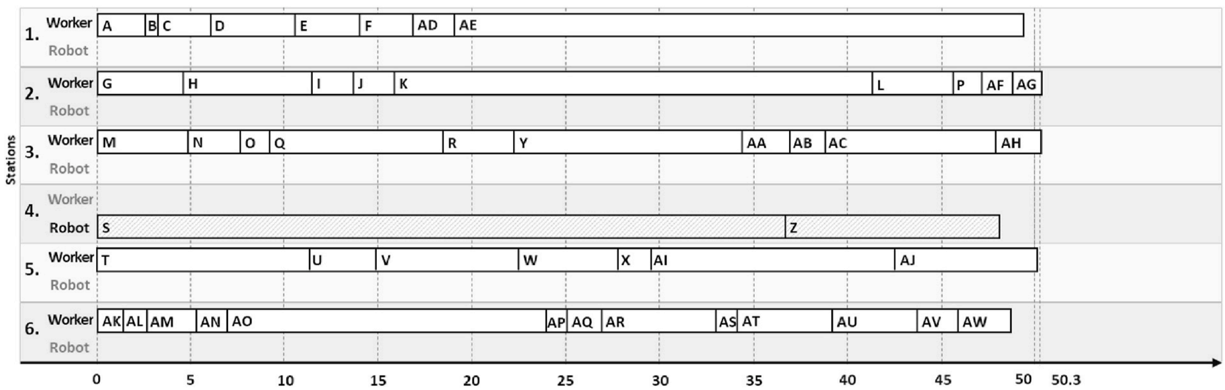

Fig. 3. Allocation of tasks to stations and workforce with optimal cycle time for case II (K = 1).

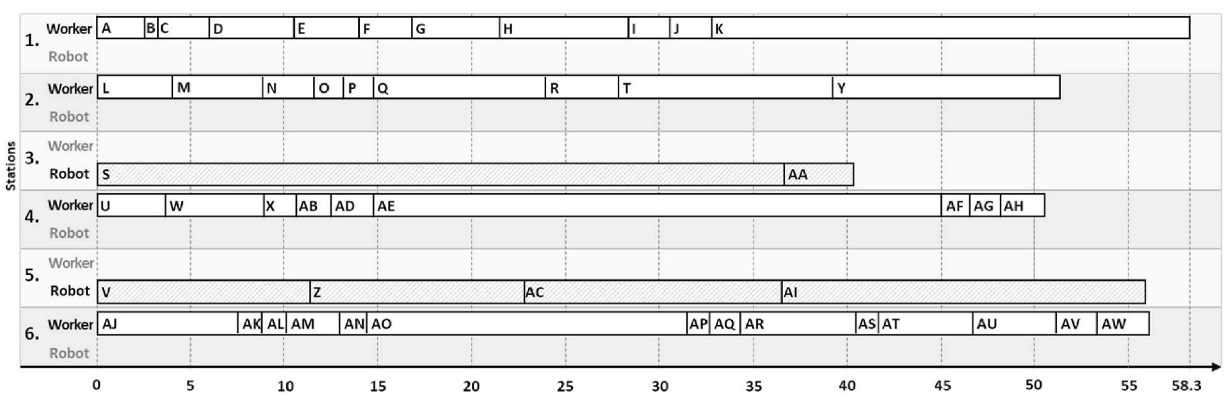

Fig. 4. Allocation of tasks to stations and workforce with optimal cycle time for case II (K = 2).

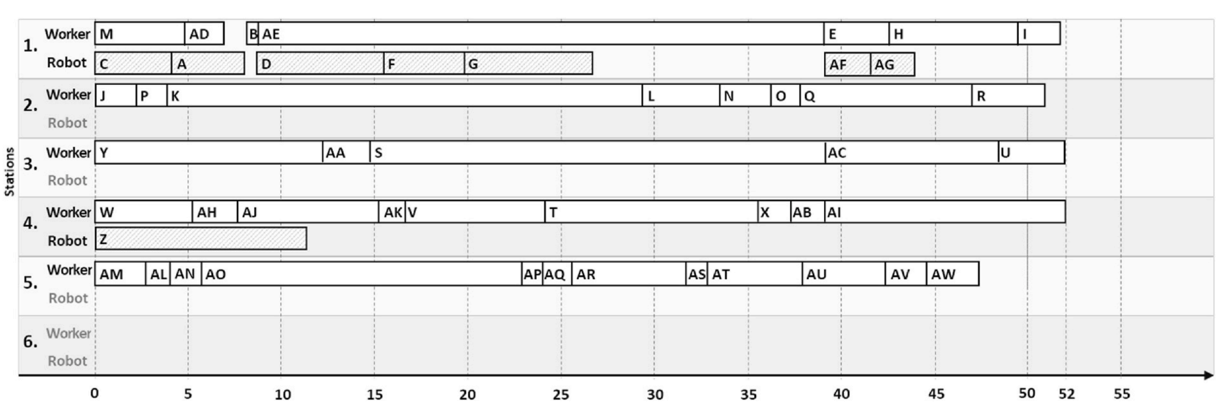

Fig. 5. Allocation of tasks to stations and workforce with optimal cycle time for case III.

and start times. The execution order of tasks M and AD allocated to the worker of station 1 has no impact on the cycle time. The start of task AD can be delayed to some extent whitout increasing the cycle time. In some cases the worker has idle time and has to wait for the robot to finish a task. In the case when only workers are used, idle times on a station can be connected and could be used as a buffer time. In the solution presented in Fig. 5, the worker of station 1 has some idle time between tasks AD and B.

It can be concluded that the two investigated ways of using robots in the assembly line had different impact on operational characteristics. Using only robots on some stations is possible only by increasing the number of stations. The reduction of the number of worker operated stations is possible only if two robots are applied. Assigning workers and robots to the same station decreases the cycle time without increasing the number of stations.

5. Managerial implications of the presented case

Technology capable of performing certain tasks in an assembly line with robots is available today at a feasible cost. The different models of Universal Robots applied in the presented case can be set up for performing a wide variety of simple and frequently repeated tasks, either alone or with the cooperation of workers.

In the assembly line presented in this paper, robot type UR10 of Universal Robot can be applied, due to its size and technological features. However, a method is needed to find the best possible line configuration. This requires the analysis of the effect of different numbers of workers and robots on the cycle time. The presented mathematical programming models can provide this information.

The four cases presented in section 5 are all feasible options for management. The advantages and disadvantages of these cases are as follows:

- Only workers are used (case 1). This is the base problem for comparison. In this case five workstations are required, consequently, five workers are working, and the cycle time is 56.8 s. Only worker-related costs are incurred.
- One robot is working at one of the stations (case 2). In this case, the use of six workstations is optimal. The number of workers cannot be decreased, but the cycle time improves (50.3 s), albeit at the extra cost of the robot.
- Two robots are working alone on the respective stations (also case 2). In this case, the use of six workstations is optimal again. This time, the number of workers is decreased to four, but the cycle time increases to 58.3 s. Consequently, the worker's cost decreases but the cost of robots increases, and line capacity decreases as a consequence of the deterioration in cycle time.
- One robot is working with collaboration of a worker at the same workstation (case 3). In this case, the use of five workstations is optimal. The number of workers is five (as in the base case), and the cycle time decreases to 52 s (below the base case). This is a compromise solution; it entails higher cost as a consequence of the application of one robot, but capacity increases as a consequence of the improvement of cycle time.

These were the most apparent configurations; however, a wider range of other possibilities can be explored if a long term, company-wide economic analysis is required. Where managerial decisions need to be taken about the deployment of robots on assembly lines, the operational and economic aspects must all be considered.

In cases like the example presented in this paper, collaborative robots can be considered similar to workers if some robot-specific and human-robot collaboration-related constraints are properly applied during task allocation. The possibility of applying the robot flexibly for different tasks allows robots to be applied at different stations in a specific line, or even to be allocated to other lines of the company. In this way, the robot can be applied flexibly and redeployed within the manufacturing systems as a result of management decisions.

The robot-related costs, however, are different from a worker's cost. The application of robots requires a significant initial investment, but the cost of their operation is lower compared to a worker's cost. Other advantages of robots over workers are their consistent accuracy, the elimination of fatigue, and the different probability and impact of accident and injury events.

A proper financial evaluation of the application of robots requires information about the long-term operation of the plant concerned, the frequency and technological consequences of product changes, and the potential for applying the robots in other assembly lines of the company. This type of examination is outside of the scope of this paper. Nevertheless, financial analysis cannot be performed without information of the effect of robots on assembly line configuration data (number of robots, number of workers, task allocation), and without the cycle time information of the different configurations. This information can easily be provided by the calculations presented in this paper.

6. Evaluation of the computational performance

The mathematical programming models presented in this paper are not easy to solve, especially in scenarios that are on a practical scale. To evaluate the computational performance of the presented models, 11 datasets from Scholl (1993) were used; each dataset is referred to by its name. Tasks in these problems are denoted by integer numbers. Computations were performed with the following settings in order to facilitate the comparison of results:

- No distance restriction was defined.
- The following tasks can be performed by robots and workers as well in each dataset: 1, 3, 4, 6, 7, 11, 19, 20, 22, 26, 27, 29, 32, 33, 35, 40, and 46 to 75.
- In each dataset the task time of robotic operation is 150% of the task time of workers operation. Proper rounding is applied to get integer values for all task times.
- For datasets where tasks have only one root task, the constraint related to the forbidden parallel execution of tasks with common root predecessor was omitted (in datasets Arcus1, Arcus2, Gunther, Lutz3)
- The model for Case II is solved by setting the minimum required number of robots to one ($K = 1$)

The complexity of the datasets is described by the following indicators:

- Shortest task duration (*STD*): the minimum value of all the task times.
- Longest task duration (*LTD*): the maximum value of all the task times.
- Order strength (*OS*): the percentage of the number of arcs in the transitive closure of the precedence graph related to the maximal number of arcs in an acyclic graph with I number of nodes.
- Time variability ratio (*TV*): the ration of the longest task duration to the shortest task duration.

Computational time was evaluated when using a laptop computer with 1.8 GHz Intel i7 processor and 16 GB of RAM.

Table 5 details the results and computation times related to the three models defined in section 3. The bold face numbers in the table are the optimal values of the respective objective functions. When the required computational time exceeded 60 min, the gap percentage after 60 min of computation and the solution (in parenthesis) when it differs from the optimal solution can be found in the corresponding "CPU time" row.

Analysing the results and computational time, the following characteristics can be observed:

- In most of the cases computation time of case II is acceptable for practical application, but for dataset Arcus2, when the cycle time is minimised, a gap of 0.02% remained after 60 min. This dataset is characterised by high *TV* value.
- In case III, the MIP model has similar performance to the CP model when the dataset is characterised by a small number of tasks or a high *OS* value.
- In case III, the MIP model had problems in the following cases: the optimum number of stations were not found within the 60-min solver run for datasets Arcus1, Arcus2 and Bartholdi; the station number computation ended with a gap for Tonge and Warnecke; the optimal cycle time was not found for datasets Arcus1, Arcus2, Bartholdi, Lutz3, Tonge, Warnecke.

To sum up, the numerical analysis proves the efficiency and the practical relevance of the MIP model for case II. For case III, however the CP model is recommended, since for large and complex problems MIP did not provide the optimal solution within a reasonable time, or at all.

7. Conclusions

In this paper, the possibility of partial automation of an assembly line was investigated. Partial automation requires some level of human-robot collaboration. A mathematical modelling framework was provided to compare three different basic setups.

In the first case, only workers perform the tasks at the workstations. This case can be described with traditional assemble line balancing models and the solution can be easily obtained. In the second case,

Table 5
Summary of the computational results.

		Arcus 1	Arcus 2	Gunther	Lutz 3	Bartholdi	Hahn	Heskiaoff	Kilbridge	Sawyer	Tonge	Warnecke
	I	83	111	35	89	148	53	28	45	30	70	58
	STD	233	10	1	1	3	40	1	3	1	1	7
	LTD	3691	5689	40	74	383	1775	108	55	25	156	53
	TV	15.8	568.9	40	74	127.7	44.4	108	18.3	25	156	7.6
	OS	40.4	40.4	59.5	77.6	25.8	83.8	22.5	44.6	44.9	59.4	59.1
	T_c	10816	11570	41	150	805	2004	138	57	30	527	111
Case 1	*N*	**8**	**14**	**14**	**12**	**7**	**8**	**8**	**10**	**12**	**7**	**14**
	CPU time (s)	8.88	235.16	1.42	8.94	7.89	0.66	0.19	1.94	0.41	1.63	3.23
	T	**9554**	**10747**	**40**	**138**	**805**	**1907**	**129**	**56**	**28**	**502**	**111**
	CPU time (s)	20.81	261.86	0.28	1.25	0.14	0.13	0.28	0.09	0.25	0.48	0.03
Case 2	*N*	**8**	**14**	**14**	**12**	**8**	**8**	**8**	**11**	**12**	**7**	**15**
	CPU time (s)	2.09	88.08	3.58	1.16	3.55	2.38	0.3	15.98	2.08	3.2	30.3
	T	**9909**	11008	**40**	**142**	**735**	**1907**	**134**	**55**	**30**	**527**	**107**
	CPU time (s)	231.12	*0.02%*	0.45	1.39	69.36	0.22	0.09	0.11	27.17	3.72	642.84
Case 3	*N*	**5**	**9**	**11**	**10**	**6**	**8**	**7**	**8**	**9**	**5**	**11**
	CP CPU time (s)	1852.8	2292.3	7.8	163.59	13.05	8.36	1.86	4.05	7.31	49.98	274.47
	MIP CPU time (s)	*16.67% (6)*	*64.29% (14)*	6.73	307.78	*71.43% (7)*	8.38	12.08	813.13	37.13	*20%*	*9.10%*
	T	**10537**	**11498**	**41**	**139**	**732**	**1827**	**126**	**55**	**30**	**389**	**104**
	CP CPU time (s)	3512.4	3392.1	5.77	306.17	1413.2	12.52	9.63	2.84	5.16	81.46	1206.9
	MIP CPU time (s)	*18.14% (9009)*	*47.09% (10752)*	1.97	*6.47% (139)*	*52.42% (805)*	23.41	42.58	2.74	3.77	*22.7% (459)*	*14.4% (109)*

robots and workers are applied, but only either of the two is assigned to a specific workstation. Finally, in the third case, workers and robots can also be assigned to the same workstation and their collaboration must be organised. A nonlinear mathematical programming model is presented for this third situation. In the case when a worker and a robot operate jointly at the same station a scheduling problem is added to task assignment and the complexity of the problem increased significantly. An MIP and a CP model are presented for this complex situation and the CP model is used in the practical implementation.

The real case of an assembly line producing power inverters was used to illustrate the application of the models. The results showed that the use of robots may decrease the cycle time, but does not reduce the number of stations, however, other considerations could also influence management decision related to the investment in automation. Tasks executed by robots have less uncertainty in execution time, could reduce the exposure of workers to hazards and absence workers do not reduce production capacity. In the other hand, implementation and maintenance costs related to automation have to be considered as well.

Note, that safety and ergonomic issues are also important when workers and robots are working together. It is assumed, however, that when a task is eligible for a robot, then safety and ergonomic issues are already considered. That is, only those tasks can be assigned to robots, which can be performed safely, even if a connecting task is performed by a worker. When special safety or ergonomic issues are expected, then additional constraints can be added to exclude collaboration in such cases.

The mathematical programming models presented in this paper show how traditional assembly line balancing models can be extended to consider the application of workers and robots at the same line. Different forms of worker robot configurations, and the effect of partial automation on task assignment and on the cycle time can be studied using the proposed methods. However, a full cost analysis of the different line configurations and a financial evaluation capable of supporting assembly line-related management decisions requires a complex decision support system. Such a system must integrate operational information for different line configurations, marketing data relating to product mix stability, and information concerning company-wide manufacturing resource reallocation possibilities. The preparation of such a system is a challenging problem requiring further research.

Credit author statement

Tamás Koltai: Conceptualization, Investigation, Formal analysis, Supervision, Writing - review & editing. Imre Dimény: Methodology, Formal analysis, Software, Visualization, Writing - original draft. Viola Gallina: Conceptualization, Investigation, Data curation, Validation. Alexander Gaal: Data curation. Chiara Sepe: Investigation, Validation, Writing - original draft.

Acknowledgements

The research reported in this paper and carried out at the Budapest University of Technology and Economics has been supported by the National Research Development and Innovation Fund (TKP2020 National Challenges Subprogram, Grant No. BME-NC) based on the charter of bolster issued by the National Research Development and Innovation Office under the auspices of the Ministry for Innovation and Technology.

References

Bartholdi, J.J., 1993. Balancing two-sided assembly lines: a case study. Int. J. Prod. Res. 31 (10) https://doi.org/10.1080/00207549308956868, 2447-246.

Battaïa, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. Int. J. Prod. Econ. 142, 259–277. https://doi.org/10.1016/j.ijpe.2012.10.020.

Borba, L., Ritt, M., Miralles, C., 2018. Exact and heuristic methods for solving the robotic assembly line balancing problem. Eur. J. Oper. Res. 270, 146–156. https://doi.org/10.1016/j.ejor.2018.03.011.

Bowman, E.H., 1960. Assembly line balancing by linear programming. Oper. Res. 8, 385–389. https://doi.org/10.1287/opre.8.3.385.

Boysen, N., Fliedner, M., Scholl, A., 2007. A classification of assembly line balancing problems. Eur. J. Oper. Res. 183, 674–693. https://doi.org/10.1016/j.ejor.2006.10.010.

Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: which model to use when? Int. J. Prod. Econ. 111, 509–528. https://doi.org/10.1016/j.ijpe.2007.02.026.

Bryton, B., 1954. Balancing of a Continuous Production Line. Unpublished M.S. Thesis, Northwestern University, Evanston, USA.

Bukchin, Y., Raviv, T., 2018. Constraint programming for solving various assembly line balancing problems. Omega 78, 57–68. https://doi.org/10.1016/j.omega.2017.06.008.

Bukchin, J., Rubinovitz, J., 2003. A weighted approach for assembly line design with station paralleling and equipment selection. IIE Trans. 35 (1), 73–85. https://doi.org/10.1080/07408170304429.

Chen, F., Sekiyama, K., Sasaki, H., Huang, J., Sun, B., Fukuda, T., 2011. Assembly strategy modeling and selection for human and robot coordinated cell assembly. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, pp. 4670–4675. https://doi.org/10.1109/iros.2011.6048306.

Cherubini, A., Passama, R., Crosnier, A., Lasnier, A., Fraisse, P., 2016. Collaborative manufacturing with physical human–robot interaction. Robot. Comput. Integrated Manuf. 40, 1–13. https://doi.org/10.1016/j.rcim.2015.12.007.

Cil, Z.A., Mete, S., Ağpak, K., 2016. A Goal Programming approach for robotic assembly line balancing problem. IFAC-PapersOnLine 49 (12), 938–942. https://doi.org/10.1016/j.ifacol.2016.07.896.

Dianatfar, M., Latokartano, J., Lanz, M., 2019. Task balancing between human and robot in mid-heavy assembly tasks. Procedia CIRP 81, 157–161. https://doi.org/10.1016/j.procir.2019.03.028.

Dimitriadis, S.G., 2006. Assembly line balancing and group working: a heuristic procedure for workers's groups operating on the same product and workstation. Comput. Oper. Res. 33 (9), 2757–2774. https://doi.org/10.1016/j.cor.2005.02.027.

Finco, S., Abdous, M.-A., Calzavara, M., Battini, D., Delorme, X., 2020a. A bi-objective model to include workers' vibration exposure in assembly line design. Int. J. Prod. Res. 59 (13), 4017–4032. https://doi.org/10.1080/00207543.2020.1756512.

Finco, S., Calzavara, M., Sgarbossa, F., Zennaro, I., 2020b. Including rest allowance in mixed-model assembly lines. Int. J. Prod. Res. 1–23. https://doi.org/10.1080/00207543.2020.1843731.

Gallo, M., Murino, T., Santillo, L.C., 2013. A proposal for an exact algorithm for solving a SALBP-1 problem. In: Proceedings of the 19th ISSAT International Conference of Reliability and Quality in Design, Honolulu, Hawaii, USA, pp. 383–388.

Giglio, D., Paolucci, M., Roshani, A., Tonelli, F., 2017. Multi-manned assembly line balancing problem with skilled workers: a new mathematical formulation. IFAC PapersOnLine 50 (1), 1211–1216. https://doi.org/10.1016/j.ifacol.2017.08.344.

Gökçen, H., Ağpak, K., Benzer, R., 2006. Balancing of parallel assembly lines. Int. J. Prod. Econ. 103 (2), 600–609. https://doi.org/10.1016/j.ijpe.2005.12.001.

Hashemi-Petrodi, S.E., Thevenin, S., Kovalev, S., Dolgui, A., 2020. Operations management issues in design and control of hybrid human-robot collaborative manufacturing systems: a survey. Annu. Rev. Contr. 49, 264–276. https://doi.org/10.1016/j.arcontrol.2020.04.009.

Helms, E., Schraft, R.D., Hagele, M., 2002. rob@work: robot assistant in industrial environments. In: IEEE ROMAN 2002, 11th IEEE International Workshop on Robot and Human Interactive Communication, Berlin, Germany, pp. 399–404.

Hignett, S., McAtamney, L., 2000. Rapid entire body assessment (REBA). Appl. Ergon. 31, 201–205. https://doi.org/10.1016/S0003-6870(99)00039-3.

Kim, H., Park, S., 1995. A strong cutting plane algorithm for the robotic assembly line balancing problem. Int. J. Prod. Res. 33 (8), 2311–2323. https://doi.org/10.1080/00207549508904817.

Koltai, T, Tatai, V, 2011. Formulation of simple workforce skill constraints in assembly line balancing models. Period. Polytech. Soc. Man. Sci. 19 (1), 43–50. https://doi.org/10.3311/pp.so.2011-1.06.

Krüger, J., Lien, T.K., Verl, A., 2009. Cooperation of human and machines in assembly lines. CIRP Ann. - Manuf. Technol. 58, 628–646. https://doi.org/10.1016/j.cirp.2009.09.009.

Levitin, G., Rubinovitz, J., Shnits, B., 2006. A genetic algorithm for robotic assembly line balancing. Eur. J. Oper. Res. 168 (3), 811–825. https://doi.org/10.1016/j.ejor.2004.07.030.

Lotter, B., Wiendahl, H.P., 2006. Montage in Deutschland—herausforderungen und Chancen. Zeitschrift für wirtschaftlichen Fabrikbetrieb 101 (9), 492–499. https://doi.org/10.3139/104.101054.

Lusa, A., 2008. A survey of the literature on the multiple or parallel assembly lines balancing problem. Eur. J. Ind. Eng. 2 (1), 50–72. https://doi.org/10.1504/EJIE.2008.016329.

Malik, A.A., Bilberg, A., 2019. Collaborative robots in assembly: a practical approach for tasks distribution. Procedia CIRP 81, 665–670. https://doi.org/10.1016/j.procir.2019.03.173.

Michalos, G., Makris, S., Spiliotopoulos, J., Misios, I., Tsarouchi, P., Chryssolouris, G., 2014. ROBO-PARTNER: seamless human-robot cooperation for intelligent, flexible and safe operations in the assembly factories of the future. Procedia CIRP 23, 71–76. https://doi.org/10.1016/j.procir.2014.10.079.

Miralles, C., García-Sabater, J.P., Andrés, C., Cardos, M., 2007. Advantages of assembly lines in sheltered work centres for disabled. A case study. Int. J. Prod. Econ. 110 (1–2), 187–197. https://doi.org/10.1016/j. ijpe.2007.02.023.

Mukund Nilakantan, J., Ponnambalam, S.G., 2015. Robotic U-shaped assembly line balancing using particle swarm optimization. Eng. Optim. 48 (2), 231–252. https://doi.org/10.1080/0305215X.2014.998664.

Nikolakis, N., Kousi, N., Michalos, G., Makris, S., 2018. Dynamic scheduling of shared human-robot manufacturing operations. Procedia CIRP 72, 9–14. https://doi.org/10.1016/j.procir.2018.04.007.

Occhipinti, E., 1998. OCRA: a concise index for the assessment of exposure to repetitive movements of the upper limbs. Ergonomics 41, 1290–1311. https://doi.org/10.1080/001401398186315.

Ogorodnikova, O., 2007. Robot introduction in human work environment. Developments, challenges and solutions. In: 5th IEEE International Conference on Computational Cybernetics, pp. 167–172.

Otto, A., Scholl, A., 2011. Incorporating ergonomic risks into assembly line balancing. Eur. J. Oper. Res. 212, 277–285. https://doi.org/10.1016/j.ejor.2011.01.056.

Özcan, U., Toklu, B., 2009. A tabu search algorithm for two-sided assembly line balancing. Int. J. Adv. Manuf. Technol. 43 (7), 822–829. https://doi.org/10.1007/s00170-008-1753-5.

Purnomo, H.D., Wee, H., Rau, H., 2013. Two-sided assembly lines balancing with assignment restrictions. Math. Comput. Model. 57, 189–199. https://doi.org/10.1016/j.mcm.2011.06.010.

Rekiek, B., Dolgui, A., Delchambre, A., Bratcu, A., 2002. State of art of optimization methods for assembly line design. Annu. Rev. Contr. 26, 163–174. https://doi.org/10.1016/S1367-5788(02)00027-5.

Roelofs, M., Bisschop, J., 2018. AIMMS the User's Guide. AIMMS B.V., Haarlem, The Netherlands.

Rubinovitz, J., Bukchin, J., Lenz, E., 1993. RALB–A heuristic algorithm for design and balancing of robotic assembly lines. CIRP Ann. - Manuf. Technol. 42 (1), 497–500. https://doi.org/10.1016/S0007-8506(07)62494-9.

Salveson, M.E., 1955. The assembly line balancing problem. J. Ind. Eng. 6, 18–25.

Scholl, A., 1993. Data of assembly line balancing problems. Schriften zur Quantitativen Betriebswirtschaftslehre, 16/93. TU Darmstadt.

Scholl, A., 1999. Balancing and Sequencing Assembly Lines, second ed. Physica, Heidelberg.

Scholl, A., Becker, C., 2006a. A survey on problems and methods in generalized assembly line balancing. Eur. J. Oper. Res. 168, 694–715. https://doi.org/10.1016/j.ejor.2004.07.023.

Scholl, A., Becker, C., 2006b. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. Eur. J. Oper. Res. 168, 666–693. https://doi.org/10.1016/j.ejor.2004.07.022.

Scholl, A., Fliedner, M., Boysen, N., 2010. Absalom: balancing assembly lines with assignment restrictions. Eur. J. Oper. Res. 200 (3), 688–701. https://doi.org/10.1016/j.ejor.2009.01.049.

Teiwes, J., Bänziger, T., Kunz, A., Wegener, K., 2016. Identifying the potential of human–robot collaboration in automotive assembly lines using a standardised work description. In: 22nd International Conference on Automation and Computing (ICAC), Colchester, pp. 78–83. https://doi.org/10.1109/iconac.2016.7604898.

Thomopoulos, N.T., 2014. Assembly Line Planning and Control. Springer International Publishing, Switzerland. https://doi.org/10.1007/978-3-319-01399-2.

Tsarouchi, P., Matthaiakis, A.S., Makris, S., Chryssolouris, G., 2016. On a human-robot collaboration in an assembly cell. Int. J. Comput. Integrated Manuf. 30 (6), 580–589. https://doi.org/10.1080/0951192X.2016.1187297.

Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K., Chryssolouris, G., 2017. On a human–robot workplace design and task allocation system. Int. J. Comput. Integrated Manuf. 30 (12), 1272–1279. https://doi.org/10.1080/0951192X.2017.1307524.

Waters, T.R., Putz-Anderson, V., Garg, A., Fine, L.J., 1993. Revised NIOSH equation for the design and evaluation of manual lifting tasks. Ergonomics 36, 749–776. https://doi.org/10.1080/00140139308967940.

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., Spengler, T.S., 2019. Balancing of assembly lines with collaborative robots. Business Research 13, 93–132. https://doi.org/10.1007/s40685-019-0101-y.

White, W.W., 1961. Comments on a paper by bowman. Oper. Res. 9 (2), 274–276.

Yilmaz, H., Yilmaz, M., 2019. A mathematical model and tabu search algorithm for multi-manned assembly line balancing problems with assignment restrictions. Eng. Optim. 52 (5), 856–874. https://doi.org/10.1080/0305215x.2019.1618288.