Journal of Intelligent Manufacturing (2022) 33:1–34
https://doi.org/10.1007/s10845-020-01641-7

A comprehensive review of robotic assembly line balancing problem

Parames Chutima[1,2]

Received: 18 November 2019 / Accepted: 31 July 2020 / Published online: 9 August 2020

Abstract

The research on the robotic assembly line balancing problem (RALBP) was originated for the first time nearly three decades ago. This problem is under the umbrella of the assembly line balancing problem in which robots and automated equipment are employed to take on human workers' roles to form a flexible assembly line. In this review paper, the development and generalisation throughout the time of the RALBP are addressed. To make the review easy to comprehend and effective, the RALBP is first classified based on the types of layouts and then further dividing up according to the 4 M (Man, Machine, Material and Method) concept. The main contributions of different articles are chronologically summarised in the form of a table. Besides, the research contribution precedence diagram is used to illustrate the sequential order and linkage relationship among researches. Finally, from the findings of the review, future research directions are pinpointed and discussed.

Keywords Review · Robotics · Assembly line · Line balancing problem

Introduction

An assembly line is a manufacturing process in which diversified parts are successively integrated with a prescribed sequence on the basic structure of the product as a semi-finished item while sequentially moving through some workstations along the line, normally by a conveyor system until the final operation is completed. Manufacturers can exploit the advantages of assembly lines to improve their productivities by effective use of standardised processes, specialised operations, and division of labours. Assembly lines are a common practice in the mass production of complex objects widely found in electronics, home appliances, and automobile industries.

Assembly operations performed in the assembly line could be realised manually, automatically, or a mixture of both. Owing to ever-changing consumer tastes, and also to survive and maintain competitiveness in the market, manufacturers have to pursue production strategies that lead towards increased flexibility and agility, reduced cost (wastes), and improved productivity and efficiency with uncompromised quality by deploying automation in their production processes. As a result, robots and various types of automated equipment are often encompassed to foster the smart factory concept. The assembly line that uses robots in its value-added operations (e.g. welding, assembling parts, or painting) is named a robotic assembly line (RAL). The RAL strengthens the adaptability of the system since its production resources could be effortlessly reconfigured to effectively produce a wide range of products.

Robots have been used in industry since the early 60 s. Metal and electrical/electronic industries are among the main industries that utilise industrial robots at substantial levels in their assembly lines to reduce labour costs and variability in operation times from manual works (International Federation of Robotics 2018). Robots have commonly replaced human workers in modern manufacturing processes to enhance automation and flexibility since they can be programmed and relentlessly worked 24/7, 365 days a year without tiredness or getting ill (Li et al. 2018a). Repetitive, difficult or dangerous, and fast and accurate operations are perfect tasks for robots. Besides, the pick and place operations are the main tasks performed by robots in the RAL (Ramachandram and Rajeswari 2004). Due to technological advancements, nowadays robots can be used to assemble virtually any object, regardless of its size or uniqueness. Moreover, highly automated assembly lines can bring about reliability, availability, productivity, product quality,

✉ Parames Chutima
cparames@chula.ac.th

[1] Industrial Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok, Thailand

[2] The Royal Society of Thailand, Bangkok, Thailand

performance, and cost reduction (Nilakantan and Ponnambalam 2016).

The RAL is worth considering as a prominent production process in the following situations. Diverse products with short life cycles and a moderate-to-high volume have to be assembled by the same equipment. Assembly tasks have to be executed swiftly, in a very short time. Manual handling of parts may not be possible or could easily cause damage. The complexity of parts makes manual assembly inefficient. Assembly work needs a high degree of repeatability and accuracy. Ergonomics, safety, and health hazards are significant human factors. Skilled labours are rare and demand substantially higher wages. A work environment in which immaculate cleanliness is necessary.

In the Industrial 4.0 era, a fully automated RAL supported by data exchange facilities is an example that illustrates the successful implementation of the new digital industrial technology. It enables the collection and analysis of data across robotic workstations through the industrial internet of things, cloud computing and artificial intelligence. The smart factory uses this data in autonomous decision-making processes to optimise its production system in real-time. Human workers are no longer needed since their roles are fully superseded by intelligence robots. Besides technical benefits, the system as such could also be a key answer to the future business's survival. A recent taxonomy of Industry 4.0 was proposed and reviewed by Oztemel and Gursev (2020).

Among the various problems of assembly line design and operation, an assembly line balancing problem (ALBP) has received excessive attention in the literature. The classical definition of the ALBP, normally quoted in the manual straight-shaped assembly line, can be referred to as follows. Given a cycle time, a collection of tasks associated with their corresponding task times and precedence relationships, assign the tasks to workstations located serially to optimise some predefined objective function, without violating resource (e.g. tasks and workstations) and technological (e.g. precedence) constraints. The concept of the ALBP is not new since the first mathematical model was formalised more than sixty years ago by Salveson (1955). Nevertheless, the robotic ALBP (RALBP), an extension of the simple ALBP in which more realistic constraints are enhanced into the automated production process, was first presented by Rubinovitz and Bukchin (1991). After that, nearly three dozen papers have been published in this research area so far.

Since the simple ALBP simplifies many aspects of real-life implementation, model formulations that incorporate additional practical industrial settings have increasingly gained more attention from the research community. Under the context of RALs, the definition of the RALBP is further extended, since the same or different types of robots assigned to each workstation impose a new task assignment constraint to the problem. The actual values of task times are not fixed but depend on the specific robot chosen to perform them. In other words, the task time of any specific task executed by different robots may vary due to the performance capability of the robot being used. In addition to an attempt to optimally balance the assembly line similar to the conventional ALBP, the RALBP also has to allocate the most efficient robot to execute tasks at each workstation. Hence, two subproblems have to be answered when optimising the RALBP, i.e. (1) task-to-workstation assignment, and (2) robot-to-workstation assignment.

The purpose of this paper is to present an inclusive review of the RALBP research that is scoped at the demonstration of the chronological development in this research area from past to present. The intention is to show a holistic view and recent trends in the research rather than to delve in-depth into some specific technical domains, which would merit challenging review papers on their own in the future. The contributions of this paper are as follows. A novel classification scheme is provided as a structural framework for systematically reviewing the literature. The research contributions from different RALBP articles are highlighted and articulated in the tables and the research contribution diagram to help provide an easier understanding of the originality, similarity and discrepancy among them. Together with this framework, the research gaps from the published literature are identified as a guideline for pursuing appropriate research directions in the future. To the best of my knowledge, no review paper on the RALBP has been published before.

The following is the structure of the paper. "Classification of RALBP" section presents the basic problem definition of the RALBP, followed by the proposition of its classification scheme. The review of the RALBP literature is chronologically presented based on the layout structure in "Literature on RALBP" section. Important statistics collected from research findings are analysed, pictorially presented, and discussed in "Research findings" section. Research gaps and future research challenges are pointed out in "Future research directions" section. Finally, "Conclusion" section is dedicated to the concluding remarks.

Classification of RALBP

General problem definition of RALBP

In the last decade, the RALBP has attracted a great deal of attention from both academics and practitioners. Generally, the RALBP aims at optimising some objective, such as minimising the cycle time by striving to allocate tasks to the workstations placed along the line to achieve a uniform workload distribution among them and then selecting the most effective robot to execute the allocated tasks in each workstation, without violating the constraints imposed by the assembly system. The argument that the RALBP is NP-hard could be inferred by using the proof of Karp (1972) in that the simple ALBP, the most basic version of the problem, is NP-hard.

As mentioned, the RALBP is a generalised version of the ALBP. Therefore, some assumptions and constraints of the manual ALBP can be succeeded to the RALBP (Rubinovitz et al. 1993), but not all and these exceptions are detailed next. In the RALBP, task times may depend on the robot types, tooling, and assembly equipment chosen to execute the tasks at each workstation. Moreover, thanks to automation, the amount of the task time variation in the RAL (almost no deviation from their expected values) is substantially lower than in a manual assembly. Hence, it is sensible to assume that task times are deterministic.

The flexible production system considered in the RALBP is comprised of an unlimited number of general-purpose robots with diverse types. Some of these robots, mostly with different types, are selected and installed in an assembly line to effectively produce a single finished good. The capability of each robot type may not be the same. Some robot types can be employed to produce any assembly task, but others cannot. Besides, the task time required to complete any task may be varied depending on the robot capability. Task times are often assumed to be deterministic since the execution times of robots are much more precise and accurate than those manual workers. The assumptions normally used in modelling the RALBP are listed below.

(1) The RAL is balanced for an exclusive model of a single finished good.
(2) An assembly task is the smallest work element that cannot be further subdivided among two or more workstations.
(3) The precedence relations between tasks are known and fixed.
(4) Task times are deterministic, and their actual values are dependent on the robots chosen to execute the tasks.
(5) A workstation can be used to execute any task if the robot assigned to the workstation is capable of completing it.
(6) Each workstation can accommodate only one robot resulting in the same number of workstations and robots is in the RAL.
(7) All robot types are always ready for use without any capacity limitation and breakdown.
(8) Robot movement, tool changing and setup times are sequence-independent and negligible or are already included as part of the task time.
(9) Loading, unloading, and transportation times of workpieces are negligible.
(10) The costs of robots are not considered.

Adopting the RAL to replace manual workers is a long-term strategic decision that involves a substantial investment. As a result, the RAL must be fully utilised to bring a rapid return on investment. To achieve the goal as such, line balancing becomes an effective tool in visualising and improving bottlenecks in the assembly line, minimising wastage, and promoting a one-piece flow. The following questions need to be answered when making robotic line balancing (RLB) decisions, i.e. (1) how to assign tasks to workstations?, (2) what robot types and how many of them should be purchased?, and (3) how to assign robots to workstations?

When establishing the RAL for the first time, given the cycle time of the line, the RALBP Type I needs to be solved to configure the assembly line by assigning tasks that a certain workstation and selecting the robot best suited to perform tasks in the workstation to realise the lowest number of workstations. As time goes by, changes in consumer tastes are normal and inevitable resulting in the need to introduce a new product model to the production process. The RAL has to be reconfigured using the existing resources (i.e. number of workstations and available robots) to respond to the change in an effective manner. This leads to the RALBP Type II which targets minimising the cycle time by assigning tasks to a given number of workstations and assigning suitable robots from the number available to each workstation.

Other types of the RALBP, which are no less important than the previous two types, are Type E, Type F, and Type Cost. Unlike Type I and Type II, the RALBP Type F examines if a feasible solution exists for predefined mixture workstations and the cycle time. Thus, maximising the line efficiency is the target of Type E, which attempts to concurrently minimise the cycle time and the number of workstations. If the monetary and economic aspects are the main concern in the configuration design of the RAL, the RALBP is classified as a Type Cost. For the problems that are not classified in the categories mentioned previously, they will be classified in Type O (others).

Various model formulations and solution methods for the manual ALBP have been proposed. An outstanding comprehensive problem classification and survey of the manual ALBP are given by Becker and Scholl (2006) and Boysen et al. (2007, 2008). Nevertheless, under the context of the RALBP, the basic formulation of the manual ALBP is a good starting point and inspiring further integration of relevant constraints to suit the operations environment of the RAL. To formulate the fundamental mathematical model of the RALBP Types I and II, the followings notations are defined (Nilakantan et al. 2017b).

Notations:

- i, j variables representing the assembly tasks ($i, j = 1, \ldots, N_t$)
- r variable representing the robots ($r = 1, \ldots, N_r$)
- w variable representing the workstations ($w = 1, \ldots, W$)
- W maximum number of workstations possible ($W \leq N_t$)
- N_w given number of workstations (Type II); or minimum number of workstations (Type I)
- N_t given number of tasks
- N_r given number of robots
- S_w collection of tasks allocated to workstation w
- $t(S_w)$ total task time of workstation w (or workstation time)
- c cycle time
- c_T given cycle time
- $pre(j)$ task j's direct predecessor
- t_{ir} time required by robot r to execute task i
- δ a very large positive number
- x_{iw} 1, if task i is allocated to workstation w; 0, otherwise
- x_{irw} 1, if task i is executed by robot r allocated to workstation w; 0, otherwise
- y_{rw} 1, if robot r is allocated to workstation w; 0, otherwise

Formal formulation: (RALBP Type I: minimise the number of workstations under a given cycle time)

$$\min N_w = \sum_{w=1}^{W} \sum_{r=1}^{Nr} y_{rw} \tag{1}$$

subject to

$$\sum_{w=1}^{W} \sum_{r=1}^{Nr} w.x_{irw} - \sum_{w=1}^{W} \sum_{r=1}^{Nr} w.x_{jrw} \leq 0; \forall (i,j) \in pre(j) \tag{2}$$

$$\sum_{w=1}^{W} \sum_{r=1}^{N_r} x_{irw} = 1; \quad \forall i \tag{3}$$

$$\sum_{w=1}^{W} y_{rw} = 1; \quad \forall r \tag{4}$$

$$\sum_{w=1}^{N_r} y_{rw} = 1; \quad \forall w \tag{5}$$

$$t(S_w) = \sum_{i=1}^{N_t} \sum_{r=1}^{R} t_{ir}.x_{irw} \leq c_T; \quad \forall w \tag{6}$$

$$\sum_{i=1}^{N_t} x_{irw} \leq \delta.y_{rw}; \forall r, w \tag{7}$$

$$x_{irw} \in \{0, 1\}; \forall i, r, w \tag{8}$$

$$y_{rw} \in \{0, 1\}; \forall r, w \tag{9}$$

An integer programming (IP) formulation of the RALBP Type I can be explained as follows. In Eq. (1), with predefined cycle time (c_T), the objective function of the RALBP Type I targets at minimising the number of workstations computed from the total number of robots assigned to workstations. The inequality in Eq. (2) expresses the precedence relationships between tasks to ensure that task j must either be allocated to the same workstation in which its directed predecessor is assigned, or the workstation located behind its direct predecessor. Then, Eq. (3) defines that each task must be allocated to one and only one workstation and must be executed by only one robot. Similarly, the expression to show that each robot must be assigned to one and only one workstation is presented in Eq. (4), while Eq. (5) states that there exists one and only one robot resided in any workstation. At each workstation, the workstation time must not be greater than the given cycle time is expressed in Eq. (6). The inequality equation presented in Eq. (7) shows the robot type to be assigned to a given workstation. Finally, the integrity restrictions of the binary variables are expressed in Eqs. (8) and (9).

Formal formulation: (RALBP Type II: minimise the cycle time under a predefined number of workstations). The mathematical model was formulated by Gao et al. (2009).

$$\min c = \max_{1 \leq w \leq N_w} \left\{ \sum_{i=1}^{N_t} \sum_{w=1}^{N_w} (t_{ir} \cdot x_{iw} \cdot y_{rw}) \right\} \tag{10}$$

subject to

$$\sum_{w=1}^{N_w} w.x_{iw} - \sum_{w=1}^{N_w} w.x_{jw} \leq 0; \quad \forall (i,j) \in pre(j) \tag{11}$$

$$\sum_{w=1}^{N_w} x_{iw} = 1; \quad \forall i \tag{12}$$

$$\sum_{w=1}^{N_w} y_{rw} = 1; \quad \forall r \tag{13}$$

$$\sum_{r=1}^{N_r} y_{rw} = 1; \quad \forall w \tag{14}$$

$$\sum_{w=1}^{N_w}\sum_{r=1}^{N_r} y_{rw} \leq N_w \tag{15}$$

$$x_{iw} \in \{0,1\}; \quad \forall i, w \tag{16}$$

$$y_{rw} \in \{0,1\}; \quad \forall r, w \tag{17}$$

The mathematical model of the RALBP Type II can be explained as follows. Equation (10) states that, with the given number of workstations, the objective function of the RALBP Type II targets at minimising the cycle time (i.e. largest workstation time of the assembly line). The precedence relationship between tasks is defined in Eq. (11). This constraint ensures that for any two tasks in the precedence diagram, the predecessor task must not be allocated to the workstation located in the succeeding order of its successor task. Equation (12) explains that each task must be allocated to one and only one workstation. Equation (13) states that each robot must be allotted to one and only one workstation. The constraint to ensure that each workstation is equipped with one and only one robot is given in Eq. (14). Equation (15) ensures that the total number of robots exercised in the assembly line must not be higher than the predefined number of workstations. The integrity restrictions are shown in Eqs. (16) and (17). Although the types of mathematical formulations in the RALBP may depend on the optimized objective function (Type I and Type II), three indispensable constraints must be included in the model formulation, i.e. precedence constraint, task-to-workstation assignment constraint and robot-to-workstation assignment constraint.

Gao et al. (2009) pinpointed that the objective function presented in Eq. (10) is non-linear, which is hard to solve by conventional exact optimisation methods. To transform the problem from a non-linear optimisation problem to a linear one, the following modifications are needed as shown in Yoosefelahi et al. (2012) and Çil et al. (2017a). A variable $c \in \mathcal{R}$ (i.e. cycle time to be optimised) is added into the model and used to formulate the objective function in Eq. (18). The decision variable x_{iw} is replaced with x_{irw} to express that task i is executed by robot r allocated to workstation w. Consequently, the constraints in which x_{iw} exists (i.e. Eqs. (11), (12) and (16)) need to be modified to accommodate the new decision variable x_{irw} as shown in Eqs. (20), (21) and (26). The new constraint (Eq. (19)) to ensure that the workstation time must not exceed the cycle time is added into the model. Equation (28) which is the integrity restriction of the cycle time is added into the modified model. The formulation based on mixed-integer programming (MIP) is shown below.

$$\min c \tag{18}$$

subject to

$$\sum_{i=1}^{N_t}\sum_{r=1}^{N_r} t_{ir}.x_{irw} \leq c; \forall w \tag{19}$$

$$\sum_{w=1}^{N_w}\sum_{r=1}^{N_r} w.x_{irw} \leq \sum_{w=1}^{N_w}\sum_{r=1}^{N_r} w.x_{jrw}; \forall (i,j) \in pre(j)) \tag{20}$$

$$\sum_{w=1}^{N_w}\sum_{r=1}^{N_r} x_{irw} = 1; \forall i \tag{21}$$

$$\sum_{w=1}^{N_w} y_{rw} = 1; \quad \forall r \tag{22}$$

$$\sum_{r=1}^{N_r} y_{rw} = 1; \quad \forall w \tag{23}$$

$$\sum_{w=1}^{N_w}\sum_{r=1}^{N_r} y_{rw} \leq N_w \tag{24}$$

$$\sum_{i=1}^{N_t} x_{irw} \leq \delta.y_{rw}; \quad \forall r, w \tag{25}$$

$$x_{irw} \in \{0,1\}; \quad \forall i, r, w \tag{26}$$

$$y_{rw} \in \{0,1\}; \quad \forall r, w \tag{27}$$

$$c \in \mathcal{R} \tag{28}$$

Karp (1972) demonstrated that the simplest version of the ALBP is NP-hard. As a subset of the ABLP, it could be easily inferred that the RALBP also falls into the NP-hard problems. As a result, the computational complexity of the RALBP can be computed from $n!/2^p$, where p is the number of precedence relationships between tasks and n is the number of tasks in the precedence diagram. Additional metrics to measure the complexity of the RALBP are comprised of the following (Rubinovitz et al. 1993; Nilakantan and Ponnambalam 2016; Borba et al. 2018).

(1) Precedence graph structures (*PGS*): Three different structures of precedence graphs could be used in the problem, i.e. chain, bottleneck, and mixed precedence graphs. In the chain precedence graph, tasks that have only one successor and one predecessor must be at least 40% of the total tasks. Furthermore, in the bottleneck precedence graph, a bottleneck task (*BT*) has at least b predecessors and these predecessors have no other successors other than *BT*, and at least one bottleneck

task is identified. However, no enforcement is imposed in the mixed precedence graph.

(2) Cycle time (c): The cycle time and production rate have a reciprocal relationship with each other. If the value of c is low, the number of workstations needed to fulfil the customer demand will be high and much harder to get a good balance and high efficiency of the line.

(3) Number of tasks (N_T): A higher computational time may be needed if a larger number of tasks are available in the precedence diagram. This metric is normally used to indicate the problem size in the conventional ALBP.

(4) Types of robots (N_R): The higher the number of various selectable robot types results in a greater the number of feasible solutions and higher computational efforts. This metric is also used in the RALBP to demonstrate the problem size.

(5) Assembly flexibility (F_{ratio}): This metric indicates the assembly sequence flexibility by measuring the technological constraints presented in the precedence diagram using $F_{ratio} = \frac{2*Z}{k*(k-1)}$; where $P_{ij} = 1$ if task j succeeds task i, and 0 otherwise; k = the number of the assembly tasks; and Z = the number of zeroes in P_{ij}.

(6) Order strength (OS): This metric is similar to the F_{ratio}, but it indicates the number of different orderings being allowed from the given precedence diagram, and can be formulated as $OS = \frac{2*N_{prec}}{k*(k-1)}$; where N_{prec} = the number of precedence relations.

(7) Robotic assembly system flexibility (R_F): This metric measures the flexibility of the system resulting from the availability of different robot types to perform tasks with different task times. This metric is formalised as $R_F = \left\{1 - \frac{M}{[n*(m-1)]}\right\}$; where n = the number of tasks; m = the number of robots; if task i cannot be performed by robot j, the incapable index $IC_{ij} = 1$, and 0 otherwise; the total number of robots that cannot perform tasks $M = \sum_{i=1}^{n} \sum_{j}^{m} IC_{ij}$.

(8) *WEST* ratio: This ratio expresses the average number of tasks in a workstation. Moreover, it indicates the expected achievable solutions under the condition that the number of tasks and robots are equal, and every robot can perform any task.

(9) Relative complexity (R_c): This metric can be computed as $R_c = \frac{Com - MinCom}{MCom}$, where Com is the CPU time to solve the problem, and $MinCom$ is the lowest CPU time of the whole problems under the study.

Proposed structure to classify RALBP

Many classification structures have been suggested to cluster the ALBP literature into heterogeneous areas of research. The traditional scheme that many researchers still use today is given by Baybars (1986), in which the ALBP is classified as simple ALBP (SALBP) and generalised ALBP (GALBP). This guideline was highly appropriate for the 1980s since at that time most research works were emphasised on the formulations and solution techniques for the SALBP. Moreover, the extensions of the SALBP assumptions to reflect more real-world applications as of the GALBP were not substantial. After the 1980s, the research tended to engage more on the GALBP variances, with increasing numbers of new emerging assembly line layouts, such as parallel workstation lines, U-shaped line, etc. As a result, the research in the GALBP becomes progressively heterogeneous mixed with many important contents, which causes Baybars' traditional research classification to no longer be effective.

Recently, many review papers have tried to use some specific attributes of the ALBP as the classification scheme instead, such as task attributes, number of models, etc. (Scholl and Becker 2006; Boysen et al. 2007; Battaïa and Dolgui 2013). Besides, some researchers provided a specific review in the ALBP, e.g. the applications of genetic algorithms in the ALBP (Tasan and Tunali 2008). Nevertheless, while some of these attributes are specific to particular problems, others are generic. Besides, some topics, worthy enough to be review topics of their own, such as disassembly, machining, etc., are also included in the review papers. Such papers without a specific focal point may create confusion to novices since too many bodies of knowledge are bombarded in the review. Moreover, using the constraints and attributes of the assembly lines to classify the problem at the top-level structure may bring about confusion and an inability to differentiate what issues they are referring to since some of them overlap with each other.

Good clustering of research has a profound effect on the effectiveness of the literature review. Be reminded that what we want from review papers is the research gap and direction to pursue future research works. The structure of the literature review should be hierarchical where the highest level of the hierarchy should comprise all the independent and exhaustive research areas. Also, the feature used in partitioning the RALBP into non-overlapping sets ought to be obvious and to appear as the main research highlight (e.g. noticeable in the keywords of the paper or being a part of the paper's name). Assembly line layout structures could serve for this purpose very well since each layout is unique and does not overlap with the others. Besides, when we search for articles with the keyword for a distinctive layout from Google, the papers that are highlighted with this particular layout will show up. The specific attributes of the problem, e.g. tasks, workstations, constraints, the way the system is manipulated, etc. are also found in the property of each particular layout.

The fundamental assembly line layouts are comprised of a multi-manned assembly line (MAL), assembly line with parallel workstations (PWAL), parallel assembly lines (PAL), U-shaped assembly line (UAL), two-sided assembly line (2SAL), and straight-shaped assembly line (StAL). Also, recently several hybrid lines (HLs) are proposed from the mixing structures of two or more fundamental line structures, such as the parallel U-line (PUL), parallel adjacent U-line (PAUL), and parallel multi-manned workstation (PMAL), etc. to achieve the synergistic advantages of several fundamental layouts.

To further subdivided the cluster to a lower level of the hierarchy, the concept of 4 M (i.e. man, machine, material and method) is adopted. Besides being a basic concept for industrial engineers, this method enables the identification and grouping of the causes that affect a particular system. The followings are the attributes under the umbrella of each M.

Man (worker)

- Human involvement: As mentioned earlier, the operations in the assembly line can be manually or automatically executed, or a combination of both methods. Under the context of the RALs, manual assembly lines are irrelevant. Therefore, the RALs could be fully automated or partially automated in which human operators and robots collaborate their assembly works. The issues of skills, learning effect, ergonomics, etc. could be included if human operators are also part of the assembly line.

Machine (robot)

- Robot related constraints: Constraints related to robots involve both hardware (e.g. feeding mechanism, gripper, jig, fixture, material handling, etc.) and software (additional operations apart from assembling workpieces such as setup, tool changing, loading/unloading, etc.). These constraints could influence the way that robots are operated and also the method to allocate tasks into the workstation.
- Real business line: The existence in the real business line of the RALBP is also observed in this paper. This information is crucial for illustrating the success rate that theoretical knowledge can bridge the gap in industrial practices.

Material (part/task)

- Models of product: In an assembly line, the number of product models launched into the assembly line indicates the flexibility of the system. The least flexible line produces only a single model. The most flexible assembly line allows two or more models to arbitrarily intermix assembly on the same line. This line is called a mixed-model assembly line. If two or more models have to be executed in batches due to significant setup cost, this type of line is called a multi-model line.
- Task time variation: Task times (operation times) are fixed and deterministic if the assembly operations are performed by a single type of robot and tool. In the case of multiple types of robots being used in the assembly line, task times are variable and deterministic depending on which robot type is used to execute it. If the assembly system is prone to defects or unreliable robots, task times are stochastic and normally follow some distribution function.
- Task assignment control: The zoning control could prohibit some forms of task-to-workstation assignments. For positive zoning, a set of tasks is compulsory to allocate to the same workstation. In contrast, if a certain set of tasks must be allocated to different workstations, it is called negative zoning. The positional assignment allows the assignment of some tasks to specific positions in the workstation only. Moreover, if the execution of two or more related tasks on different workstations has to be done in parallel, it is called the synchronous assignment.

Method (problem/decision)

- Problems type: Various problem types are normally considered in optimising the RALBP, i.e. *Type I*, *Type II*, *Type O*, *Type E* and *Type F*. The definitions of the RALBP Type I and Type II were given in the previous section. Also, *TypeCost* is included to represent a cost-related RALBP that aims to optimise some objective function, such as the resource (robot) cost, inventory cost, equipment purchasing cost, assignment cost, etc. The remaining types of problems are classified in *Type O* (others) which could include the problems that attempt to optimise the smoothness index, energy consumption, etc.
- Number of objectives: To optimise the RALBP, single or multiple objectives could be considered. The type of the RALBP reflects the objective to be optimised for a single objective problem. For example, given the number of robots available, the RALBP Type II attempts to minimise the cycle time. In some cases, several conflicting objectives need to be achieved simultaneously, known as the multi-objective RALBP. Although more than one objective needs to be optimised, the primary objective is still used to identify the type of problem.
- Problem formulation: This element describes how the RALBP is formulated, such as integer linear programming (ILP), mixed-integer programming (MIP), non-linear programming (NLP), goal programming, etc.

- Solution techniques: Several solution techniques can be used to solve the RALBP, such as exact optimisation technique (e.g. branch-and-bound), heuristic, and metaheuristic (e.g. genetic algorithm, simulated annealing).
- Simultaneous decision: To gain a holistic view of the assembly system, the RALBP can be jointly optimised in conjunction with another decision problem, e.g. sequencing, buffer allocation, etc. or when unconventional operational conditions are taken into account in the problem definition. The simultaneous decision is presented in descriptive wordings. Such concurrent decision making could lead to a new challenge in the research of the RALBP.

Figure 1 shows the proposed classification scheme of the RALBP used in the review of the articles. The column following the RALBP is the layout of the assembly lines and next to that are the characteristics of the problems grouped by 4 M.

Literature on RALBP

According to the proposed structure of the review presented in "Classification of RALBP" section, the layout types which are the most obvious features of the assembly lines are employed in dividing the RALBP. In the content of the review, italics are used to highlight the main contribution of each paper to help readers to catch vital points more easily. The details of the review are presented as follows.

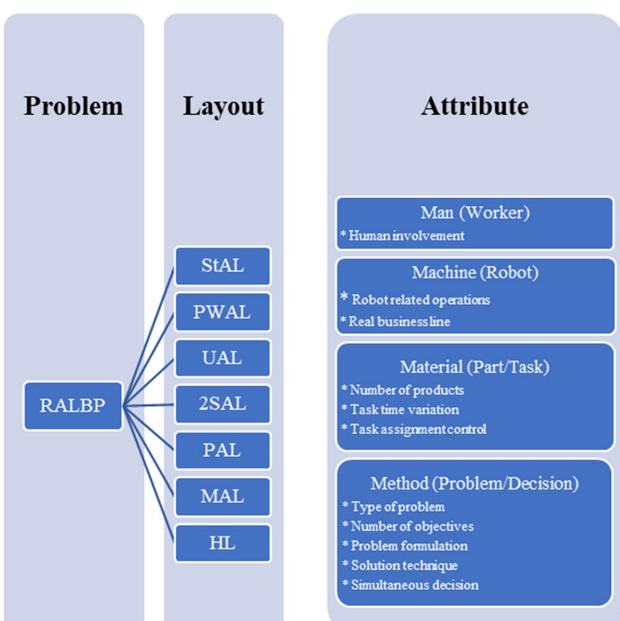

Fig. 1 Classification scheme used in this paper

StAL

Rubinovitz and Bukchin (1991) were the *foremost pioneers* in the research area of *the RALBP*. They formulated a *linear programming model* for the *RALBP Type I* under the *StAL* layout. With a given cycle time, the formulated model targeted the minimising of the number of workstations by uniformly allotting assembly workloads to workstations while assigning the most effective robot type from the given robot pool to each workstation to complete them. Their model assumed that the line produced a single product and only one robot was equipped in every workstation to perform the allotted tasks. Besides, task times were variable and deterministic, and their values depended on the selected robot.

Rubinovitz et al. (1993) described that, in the design and balancing of the RAL, two objectives had to be achieved simultaneously, i.e. optimal balance of the assembly line without violating production constraints and efficient robot allocation to each workstation. Task times were assumed to be deterministic and their actual values depended on the specific robot and equipment selected to perform them at a given workstation. The assembly environment was in the StAL layout in which a single product was produced. *The search limiting heuristic rules incorporated with branch-and-bound (B&B) algorithm* was proposed to solve large and complex real-world problems within an acceptable computation time. The algorithm generated a search tree by allocating tasks and robots to workstations and terminated after an optimal solution was found. The problem complexity graphically presented by the relations between the total nodes and the RALBP parameters (i.e. cycle time, task work element, robot type, and assembly flexibility) were observed. Also, they reported that the proposed algorithm could solve the RALBP for up to 100 tasks with three available robot types under various assembly flexibilities and robotic assembly system flexibilities.

Kim and Park (1995) considered the RALBP Type I in a serial robotic assembly line in which the additional constraints of the *limited tray storage space for parts and capacity of robot hand tool* were also taken into account. Tasks, parts, and tools were assigned to the assembly line, which produced a single product to minimise the number of robotic cells. An integer linear programming (ILP) model was formulated, and the strong cutting plane algorithm was proposed as the solution method. The proposed algorithm was tested with *two real-wold problems* (31 and 34 tasks) from a VCR-deck (camcorder) robotic assembly line and also eight problems (8, 11, 12, 14, 17, 21, 22, and 45 tasks) from the open literature with various parts and tools. It was found that the proposed approach could discover a lower bound on the optimal solution. Also, for the real industry case, the number of robots achieved by the proposed method was less than the currently used number under the conditions of the same tray and slot capacities and the same cycle time.

Hong and Cho (1997) considered the problems of the *robotic assembly sequencing problem jointly with the ALBP*. These two problems attempted to optimise different objectives. For the assembly sequence problem, the minimum assembly cost that satisfied assembly constraints was needed; whereas, for the specified cycle time, the ALBP tried to yield the minimum number of workstations. They demonstrated that the minimum number of workstations might not be realised if the line balancing problem was not considered while creating cost-effective assembly sequences. Two real industry cases, i.e. *electrical relay* (*10 parts*) *and automobile alternator* (*13 parts*), in which only one product type was assembled by a serial line, were used to verify the argument. The energy function is taken into account as a penalty for the assembly constraints (i.e. precedence and connectivity constraints) as well as the costs of assembly and idle time. A simulated annealing (SA) algorithm used the energy function to generate assembly sequences that considered the line balancing. The performance of SA was evaluated against the expert system that considered only the assembly cost in generating the assembly sequences. It was shown that the energy function and the number of workstations achieved by the expert system were sometimes higher than the SA. In other words, the minimum number of workstations may not be realised unless the cost-effective assembly sequences are jointly created by taking into account the line balancing problem.

Levitin et al. (2006) studied the RALBP *type II* aiming at *minimising the cycle time*. A single product was assumed to be produced by the StAL. The genetic algorithm (GA) incorporated with two different procedures, i.e. recursive and consecutive assignment procedures, to assign robots with different capabilities to workstations was proposed. Besides, the workpiece exchange local search procedure was also embedded in the algorithm. The performance of the GA was compared with the B&B algorithm under different characteristics of the problems, i.e. flexibility ratio (F-ratio), work element to station number ratio (WEST-ratio), number of different robot types (N_r), robot flexibility (RF) and robot expected time variability (RETV). The results showed that GA with the consecutive assignment procedure yielded a better average cycle time than the recursive one. Also, the GA with a consecutive assignment procedure was compared against the B&B one in minimising the number of workstations. The B&B could attain optimal solutions for only a small subset of problems, while the GA was superior to the heuristic version of B&B, especially for large and high complex problems.

Gao et al. (2009) considered the RALBP type II. The StAL was balanced for a single finished good. As a result of a new product model being launched, the assembly line was adjusted *using the same number of workstations and currently available robots* to accommodate the change. Hence, to achieve the minimum cycle time, the line balancing decisions involved the allocation of tasks to a predefined number of workstations and the allotment of the existing robots to each workstation. The integer non-linear programming (INLP) approach was used to formalise the problem complexity. As a result, for large-scaled problems, the GA hybridised with local search procedures (hGA) was proposed. The searchability of the hGA could avoid being trapped in the local optima and so lead to better solution quality and reduce the computation time. Five local search procedures, i.e. three on robot assignments and two on task sequences, were operated under the principle of variable neighbourhood search (VNS). The performance of hGA was tested against GA with the two different robot-to-workstation assignment procedures(recursive and consecutive assignments), developed by Levitin et al. (2006). The problems used to evaluate the relative performances differed in the number of tasks (25, 35, 53, 70, 89, 111, 148, and 279), the number of robots/workstations, and WEST ratios. The results showed that the Ilog OPL software could solve only very small-sized problems (25 tasks with three and four robots) and the same optimal solutions were also obtained by hGA. For large-sized problems, hGA gave satisfactory solutions in an acceptable time and outperformed the other tested algorithms.

Nilakantan and Ponnambalam (2012) studied the RALBP type II. A single product was produced by the assembly line. The local exchange procedure embedded in the *particle swarm optimisation* (PSO) algorithm was proposed to obtain an improved solution quality of minimising the cycle time. The test problems were comprised of 25, 35, 53, 70, 89, 111, and 148 tasks with four different number of workstations. The performance of the PSO algorithm was evaluated against the GA with recursive (GA + recursive), with consecutive (GA + consecutive), or hybrid GA. The results indicated that the PSO algorithm outperformed the others with a satisfactory CPU time.

Yoosefelahi et al. (2012) studied the RALBP Type II to optimise *multiple objectives* simultaneously, i.e. *minimising the robot costs, robot setup costs, and cycle time*. The StAL was balanced for a single product. The robot setup times and purchasing costs were also taken into account. Also, *a discount would be given if more than one robot of the same type was purchased*. The mixed-ILP (MILP) model was formulated to demonstrate the complication of the RALBP. Since the problem was NP-hard, three metaheuristics were proposed, i.e. constraint multi-objective evolutionary strategy (CMOES), Pareto archive evolutionary strategy (PAES), and hybrid multi-objective evolution strategy (HMOES), which was the combination of both aforementioned algorithms. Two test-bed problems were employed to test their relative performances, i.e. Problem I (three robots, five workstations and 10 tasks) and Problem II (five robots, five workstations and 35 tasks). The numerical results based on the *Pareto optimal sense* (i.e. the number of non-dominated solutions, maximum spread, and spacing) indicated that HMOES outperformed CMOES and PAES. They also discussed big manufacturers that tend to purchase only a single standard robot to execute all the tasks to ease maintenances and save costs. The Pareto fronts of the cycle time and robot cost illustrated that benefits that could be achieved from using different robot types, e.g. less cycle time, better quality, more productivity, etc.

Daoud et al. (2014) studied the RALBP *Type E* where *pick-and-place robots* were used as the material handling system in *an automated packaging line dedicated for dairy food products* to seize product components and assemble them at various points on the conveyor. The optimized objectives were comprised of maximising the line efficiency (measured by *the number of product components seized by the robots*) and balancing the tasks between robots. The assembly line produced only one type of product. Due to high execution time, an exact solution method was inappropriate for this case study (i.e. time to reach a final solution must be less than one second). Therefore, three hybrid metaheuristics were proposed, i.e. hybrid GA and guided local search (GA-GLS), hybrid particle swarm optimisation and GLS (PSO-GLS), and hybrid ant colony optimisation and GLS (ACO-GLS) algorithms. The proposed algorithms employed the GLS to enhance their searchability by avoiding the local optima to gain a high-quality solution. The proposed algorithms were compared with their original versions (ACO, PSO, and GA) and a full enumeration method (FEM). To benchmark the performance of the algorithms, four test cases varying in terms of the number of robots, location points, and the number of layers in the final assembled product, were setup. It was shown that the exact FEM method failed to reach the final solution within a given CPU execution time requirement. All three proposed hybrid algorithms passed the industrial requirement by obtaining the final solutions within one second. All the hybrid algorithms were better than their original versions, but theACO-GLS performed the best and could reach the optimal solutions found by FEM in almost all tested problems.

Nilakantan et al. (2015a) proposed a 0–1 integer programming model for the RALBP Type II which attempted to optimise the total energy consumption and cycle time simultaneously. The RAL was assumed to produce a single product. To deal with the problem, two models were developed, i.e. time-based model (key effort was on minimising the cycle time) and energy-based model (key effort was on minimising the total energy consumption). Both proposed models were well pertinent to the automobile body shop. The PSO was applied as the solution technique. Thirty-two instances, classified into small (25, 35, 53, and 70 tasks) and large (89, 111, 148, and 279 tasks) sizes were used as test problems. It was revealed that the energy-based model obtained lower total energy consumption than the time-based model. In contrast, the time-based model outperformed the energy-based model in terms of reduced cycle time.

Nilakantan et al. (2015b) considered the RALBP Type II. In their study, the line produced a single product and task times were based on the assigned robot types. The zero–one integer programming (0–1 IP) formulation was developed to model the RALBP. The test instances were formed by varying the number of tasks (25, 35, 53, 70, 89, 111, 148, and 297 tasks) and available workstations/robots. It was found that the CPLEX optimisation solver was able to reach the solutions only for the instances with 25, 35, and 53 tasks. This indicated the NP-hard characteristic of the problem. As a result, two bio-inspired algorithms, i.e. hybrid cuckoo search and particle swarm optimization (CS-PSO), and PSO algorithms, were proposed for large-sized instances. These two algorithms were compared against the hybrid GA. The statistical analysis revealed that the solutions found by PSO and CS-PSO were close to those of the CPLEX and they outperformed hGA. Between them, CS-PSO performed better than PSO.

Çil et al. (2016) simultaneously considered *three objectives* in the RALBP Type II, i.e. minimising the total robot utilisation cost, minimising the number of workstations, and minimising the cycle time. Mixed-model products were produced by the robotic assembly line. The *pre-emptive goal programming* (*GP*), which ranked the goals according to their importance and optimised them hierarchically, was formulated. Since the priority level in the pre-emptive GP had an impact on the goal realisation, all combinations of them were evaluated. The small-sized examples (25 tasks and five robots) with 3! (6) different orders of priority levels were solved by the proposed technique using the GAMS software. The results showed that Goal 1 (minimising cycle time) conflicted with Goals 2 (minimising the number of workstations) and 3 (minimising robot utilisation cost). In contrast, Goals 2 and 3 had no conflict with one another.

Çil et al. (2017a) studied the RALBP Type II in which *mixed-model products* were produced by the assembly line. For a given number of workstations, the objective to be optimised was to *minimise the sum of cycle times over all models*. The task times were variable deterministic, and their values were based on the assigned robots. The MILP formulation was proposed for the small—(25 and 35 tasks) and medium-sized (53 tasks) problems. Adopting heuristics was acceptable for large-sized problems since the RALBP was NP-hard. Therefore, the beam search (BS) algorithm, a heuristic which is a structural adaptation of the B&B technique, using different priority rules as the local search method [i.e. random search (RS), descending minimum positional weights (MaxPW−), shortest task time (STT), maximum follower (MF), mixture of maximum follower and longest task times (MF&L), and mixing priority rules (MPR)], were developed. It was observed that BS with MPR outperformed the other versions; whereas, BS with STT and MaxPW− performed very unsatisfactory.

Nilakantan et al. (2017a) studied the RALBP Type E to simultaneously optimize *maximising the RAL efficiency* and *minimizing the total carbon footprint*. The power consumptions of robots were used to compute the carbon footprint in terms of the energy consumption during performing operations and that on standby (idle between operations). The StAL was used to assemble a single product model. The problem was formalised with a mixed-integer linear programming (MILP) formulation. Since the problem was NP-hard, they proposed the multi-objective co-operative co-evolutionary (MOCC) algorithm for large-sized problems. The extending operator of MOCC was modified to achieve a better result by avoiding getting stuck in a local optimum. Two groups of problems were employed to evaluate the proposed algorithms, i.e. small (25–53 tasks) and large (70–297 tasks) sizes. MOCC was tested against two single objective algorithms, i.e. SA-1 (simulated annealing aiming at minimising line efficiency) and SA-2 (simulated annealing aiming at minimising total carbon footprint). Moreover, three other multi-objective optimisation algorithms, the artificial bee colony (ABC), restarted simulated annealing (RSA), and fast-elitist non-dominated sorting genetic (NSGA II) algorithm, were also compared with MOCC. The unary epsilon and hypervolume ratio metrics were the Pareto compliant measurements used to gauge the effectiveness of the algorithms. The results showed that the Pareto archive of MOCC dominated SA-1 and SA-2. Also, MOCC achieved the best average results in the hypervolume ratio and unary epsilon indicators in comparison with RSA, NSGA II, and ABC.

Nilakantan et al. (2017c) addressed the RALBP under two different objectives, i.e. minimising cycle time (Type II) and minimising total assembly line cost (Type Cost) with two layouts, i.e. StAL and UAL, in which a single product was assembled. Two zero–one IP models (i.e. time-based model and cost-based model) were formulated for each layout. Since the problem was NP-hard, the differential evolution (DE) algorithm was proposed for large-sized problems. Two problem sizes were used in comparing the results from different layouts, i.e. small size (25, 35, 53, and 70) and large size (89, 111, 148, and 297), both with the different number of robots. The best results obtained from DE showed that better total assembly line costs were achieved by the cost-based model. In contrast, lower cycle times were gained from the time-based model. Furthermore, the total assembly line cost and cycle time of the UAL were better than the StAL.

Pereira et al. (2018) proposed the *cost-oriented RALBP* that explicitly optimised *fixed and variable costs*. The RAL was balanced for a single product. Only one single robot selected from various available robot types was installed in each workstation. Besides, several workstations could be equipped with the same robot type and some robots in the pool may not be used. Each robot was operated with a fixed cost. The binary IP model was formulated to concurrently allocate tasks and best-suit robots to workstations. *The RALBP was proved to be strongly NP-hard*, even with the simple case of two workstations. Two lower bounds were proposed, i.e. cheap-to-calculate and expensive-to-calculate. Also, an elitist memetic algorithm (MA) was developed for large-sized instances. The effectiveness of the MA was evaluated against the GA, multi-start local search (MLS), and RS under a various number of tasks (25, 35, 50, 53, 70, 89, 100, 111, 148, and 297), graph structures, task time distribution, order strength of precedence graph, number of robots, and cost structures. The solution quality found by MA was nearly equal to the best-found solutions. Also, MA performed better than MLS, GA and RS. Moreover, MLS and GA obtained significantly better solutions than RS.

Borba et al. (2018) proposed *a lower bound* for the RALBP aiming at minimising the cycle time. A single product was produced by the RAL. The chain decomposition was used to investigate the graph dependencies. As exact algorithms, the MILP model and branch-bound-and-remember (BBR) algorithm with problem-specific dominance rules were developed. Also, the iterative beam search (IBS) heuristic with the same problem-specific dominance rules and lower bounds were proposed. The performances of the proposed algorithms were evaluated against the CS-PSO and PSO algorithms on various instances (35, 53, 70, 89, 111, 148, and 297 tasks). The experiments demonstrated that good solutions could be obtained even with the proposed exact solution method. Moreover, the proposed heuristic could find good solutions in an acceptable time.

Li et al. (2018b) were the first to address the RALBP which *simultaneously balanced and sequenced mixed-model products* to minimise makespan. As a result, they had to deal with one more subproblem of the *mixed-model* products sequencing on the assembly line, in addition to the task- and robot-allotment subproblems like the conventional RALBP. They opted for the simultaneous approach in optimising the mixed-model RALBP and sequencing problem rather than the hierarchical approach because the former was significantly better. A combined precedence diagram was used when balancing decisions were made. Besides, task times were variable deterministic which depended on the product models and allocated robots. For the small-sized problem (11, 25, 35, 53, 70, and 89 tasks with different numbers of robots and product mixes), the MILP formulation was proposed as the problem-solving method using the CPLEX solver. For large-sized problems (111, 148, and 297 tasks with the different number of robots and product mixes), the restarted co-evolutionary GA (RCoGA) and RSA algorithms were presented. The restart mechanism was embedded into the proposed algorithms to quickly search for a good solution and was capable of escaping from local optima. The results (measured from the relative percentage increase; RPI) showed that the proposed algorithms were efficient in obtaining better average RPI values than their original versions of the SA, original co-evolutionary algorithm (CoGA1), and co-evolutionary algorithm without a restart (CoGA2), as well as the benchmark algorithms of the ABC and GA).

Janardhanan et al. (2019) considered the RALBP type II in which the *sequence-dependent robot setup times* were also incorporated. In their study, the setup times caused by tool changes depended on the previous and following tasks to be performed by robots. Besides, the setup (forward setup) could occur directly after the preceding task was completed and before the following task was performed in the same cycle. They also assumed that task times were deterministic and fixed, setup times were variable deterministic depending on the preceding task, and loading and unloading times were negligible. The StAL was balanced for a single finished good. The MILP formulation was proposed to handle small-sized problems (11–70 tasks), while for large-sized problems (89–297 tasks), the migrating birds' optimisation (MBO) algorithm was used. The performance of MBO was tested against ABC, PSO, GA, discrete cuckoo search (DCS) and SA algorithms through the relative percentage deviations (RPD). The CPLEX solver could reach optimum solutions only for the small-sized problems, which indicated the complexity of the problem. Moreover, the statistical analyses showed that MBO obtained better average RPD values than the others for all three scenarios (i.e., low-variable, high-variable, and no setup times).

Weckenborg et al. (2019) investigated the deployment of *collaborative robots* in the RALBP type II. The salient characteristic in this study was that human workers and light-weight robots were allowed to work on the same workpiece in collaboration or to perform different tasks in parallel in a shared common workstation. Under the working environment of human–robot collaboration, workers' health could be improved since repetitive tasks and ergonomically stressful tasks would be executed by robots. This study assumed that the StAL was used to produce a single finished good with fixed deterministic task times. However, the task times depended on the process alternative (human, robot, or collaborative human–robot). They proposed the MIP formulation to tackle the small-sized problems (20 tasks); whereas, a hybrid GA was developed for medium—(50 tasks) and large—(100 tasks) sized instances. Owing to the complexity of the problem, the standard solver could reach optimal results for only a few system configurations. On the other hand, the hybrid GA yielded promising results for both the solution quality and CPU time. Also, by deploying collaborative robots, a substantial increase in the efficiency of manual work and productivity improvement in manual assembly lines were noticeable.

Dalle Mura and Dini (2019) proposed a software based on GA for balancing assembly lines in which humans and robots worked collaboratively. The skills of workers and collaborative robots were directly considered in the RALBP. The weighted sum of the normalised objectives was considered, including minimising the assembly line cost, minimising the number of skilled workers, and minimising the variance of workers' energy workloads. The proposed software was tested on the scooter chassis assembly line comprised of 45 tasks, three skill levels of workers and two dexterity levels of robots. The results showed that the energy expenditure of workers was reduced by collaborative robots. Also, tasks that needed the same skills and equipment were grouped in the same workstations.

Weckenborg and Spengler (2019) developed a MILP model for the assembly line, in which human workers and collaborative robots may be allocated to the workstations under ergonomics consideration. The formulation attempted to minimise the cost per cycle by considering the cycle time and workstation and resource cost per minute. The high-pressure cleaner assembly line was adopted as an example case. The cost-efficient configurations were realised from the proposed approach.

PWAL

Tsai and Yao (1993) proposed an integrated study of the capacity planning and RALBP for a serial-type flexible RAL where *more than one robot of the same type* was allowed to operate in each workstation. A single product was produced by the RAL. The capacity planning process was comprised of the two stages of (1) preliminary planning the robot capacity with an *ILP* model, and (2) capacity adjustment with *simulation*. To minimise the weighted average standard deviation of all workstation output rates (WASDOR), the initial number and type of robots assigned to each workstation were determined in the capacity planning stage. The output rates of robots obtained from the first stage as well as product demands, available budget, and purchasing cost, were transformed into the simulation model (SLAM II). In the *proposed heuristic rules*, the absolute weighted average different index (AWADI) and waiting time of the workstation were used to specify the potential bottleneck. If the deviation of all workstation output rates could be improved, the recommendation of the new robot combinations would be given by the *workstation adjustment procedure*. On the most critical bottleneck, the adjustment was made one by one in each consecutive simulation run. The proposed procedure was evaluated through ten case studies with six workstations, three selectable robot types, five variants of a product family and demands, different task times and robot procurement costs. The simulation outputs revealed that the average queueing times and WASDOR were improved significantly after the adjustment stage was terminated.

UAL

Nilakantan and Ponnambalam (2016) were the first to address the RALBP Type II in the UAL layout. The task-to-workstation assignments in the UAL differed from the StAL since they could be assigned after either all of their predecessors (forward assignment) or successors (backward assignment) were allocated to the earlier or the same workstations. The UAL in their study was designed to assemble a single product model. The 0–1 IP formulation was proposed for small-sized (25 and 35 tasks) problems. They demonstrated that this problem was hard to solve by traditional exact optimisation methods through the optimisation software (LINGO). Out of 32 test problems, LINGO found optimal solutions in an acceptable time only for four small-sized problems (25 tasks). As a result, metaheuristics were an effective approach to tackle practical-sized problems. For medium—(53, 70, and 89 tasks) and large-sized (111, 148, and 297 tasks) problems, the PSO algorithm embedded with a consecutive allocation heuristic, which attempted to allocate to each workstation with the maximum number of tasks, was developed. They revealed that the cycle times of the UAL were smaller than the StAL. The proposed algorithm was also tested on an actual case (i.e. engine cradle manufacturer) in the automotive industry, where 35 tasks had to be completed by a line equipped with five robots. They showed that the cycle times of the UAL obtained from the proposed PSO algorithm were lower than those of the StAL solved by the hGA algorithm.

Nilakantan et al. (2016) put forward the serious current issue about the environmental impacts and energy cost rise leading to an interest in energy-efficient factories. Hence, for a given number of workstations, they studied the RALBP Type E to *minimise the total energy consumption* (corresponding to *maximise the line efficiency*) of all installed robots. A single product model was produced by the UAL. Two metaheuristics, the DE and PSO algorithms were proposed and the test problems were classified into small (25, 35, 53, 70, and 89 tasks) and large (111, 148, and 297 tasks) sizes. The statistical analysis (Wilcoxon signed-rank test) revealed that DE outperformed PSO in gaining a higher line efficiency and lower cycle time. Conversely, the solutions from PSO had a lower energy consumption than DE.

Rabbani et al. (2016) considered *multi-objective* RALP Type II which attempted to *simultaneously minimise four objectives* (cycle time, sequence-dependent setup cost, robot setup cost, and robot purchasing cost). Mixed-model products, in which two types of tasks were considered, i.e. common task for several product models and special task for one product model, were assumed in this study. Also, the purchase costs of different robots and setup times for tasks and robots were taken into account. For small-sized problems, the MILP formulation was proposed. The GAM software with the CPLEX solver was used for searching for solutions. For large-sized problems, the multi-objective PSO (MOPSO) and NSGA II were employed to find *Pareto optimum solutions*. The relative effectiveness between NSGA II and MOPSO was compared through the four *Pareto* measurement metrics of the mean ideal distance, spacing, diversity, and a number of Pareto solutions. The results indicated that NSGA II outperformed MOPSO in general.

Nilakantan et al. (2017b) studied the RALBP Type II on two different layouts (*StAL and UAL*) to produce the same single product. Different robot types were distributed among the workstations, one each, to execute the allocated tasks. The 0–1 IP technique was used in modelling the small-sized problems of both layouts. Two PSO algorithms were developed for the StAL (one with consecutive and the other with recursive allocation procedure). The proposed PSOs were then evaluated against the hGA. Various benchmarked problems both created randomly and those known to exist in the industry (i.e. auto engineer cradle, hot tank, electronics, refrigerator, small utility vehicles, and engine assemblies), with a different number of tasks, i.e. small—(25 and 35 tasks), medium—(53, 70, and 89 tasks), and large—(111, 148, and 297 tasks) sized problems, and a different number of robots/workstations were employed. The result indicated that the PSO with a consecutive allocation procedure yielded better solutions than the PSO with a recursive allocation procedure or the hGA. Furthermore, the PSO with consecutive allocation procedure was also applied in the UAL. It was clear that under the UAL layout with the same problems, the proposed PSO yielded better cycle times than those obtained in the StAL layout.

2SAL

Aghajani et al. (2014) were the pioneers in researching the RALBP Type II. The 2SAL was balanced for mixed-model finished goods. The *setup times for each robot and sequence-dependent setup times between tasks* were taken into account. Due to the 2SAL layout, interferences between tasks allocated at the opposite sides (Left and Right) of the line could occur, which might then cause idle time if the mated station needed to wait for its companion station to complete the preceding task. For the small-sized problems

(9, 12, 16, and 24 tasks), the MIP technique was used to formulate the model, while the SA algorithm was exercised to handle the large-sized problems (65 and 148 tasks). No significant difference between the outputs obtained from SA and the GAMS software was noticed in the small-size problems. Conversely, for the large-sized problems, SA was more effective than the GAMS software in searching for near-optimal solutions.

Li et al. (2016a) discussed the robotic assembly line in which two robots were operated facing each other in the same mated-station and both of them worked on different sides of the same large-sized workpieces at the same time, namely 2SAL. The assembly line produced a unique single product model. The MIP was formulated for the RALBP type II. Since the problem was NP-hard, the co-evolutionary PSO (C-PSO) algorithm was developed, in which the concept of the co-evolutionary algorithm was incorporated with the PSO algorithm. The algorithm was operated under three improvement mechanisms, i.e. restart mechanism, modification of global best, and local search for the global best solution. The test instances were comprised of small—(9, 12, 16, and 24 tasks) and large—(65, 148, and 205 tasks) sized problems to compare the C-PSO against co-evolutionary GA (C-GA), SA, ABC, GA, and PSO algorithms using the RPI metric. The experimental results showed that optimal solutions were only found by the CPLEX optimisation solver for the small-sized problems. However, the C-PSO could reach optimal or near-optimal solutions and outperformed the other algorithms for almost all tested problem sizes.

Li et al. (2016b) pointed out that the global environmental awareness and high energy consumption cost necessitated research on reducing the total energy consumption. Therefore, they studied the RALBP type II in the 2SAL to simultaneously minimise the cycle time and *the energy consumption on all stations*. The 2SAL was balanced for a single finished good. Moreover, the energy consumption was calculated from the total of the robot power consumption during the production and standby modes. The MIP formulation was used in modelling the problem with these two conflicting objectives. The RSA algorithm, which utilised local search and restarting mechanisms, was proposed since the problem was NP-hard. The performance of RSA was evaluated against two single-objective SA, i.e. SA aiming at minimising the cycle time (SA-CT) and SA aiming at minimising the total energy consumption (SA-TE), as well as SA that did not use the restart procedure (SA1), SA that chose a solution at random from the Pareto solutions in the restart procedure (SA2), and NSGA II. The benchmark problems consisted of small—(9, 12, 16, and 24 tasks) and large-s (65, 148, and 205 tasks) sized problems, where those with 65 and 205 tasks were real industry cases obtained from truck assembly lines. The three metrics were employed to assess the performance of the different algorithms were the Pareto solutions, convergence, and spread. The results demonstrated that the Pareto fronts of RSA was better than the other benchmarked algorithms in terms of both the convergences and spread criteria.

Li et al. (2018a) addressed the application of RALBP in the *2SAL* where two robots were allocated in a mated-station to execute tasks on a large-sized item simultaneously. The 2SAL produced a single product model, task times were variable deterministic depending on the assigned robot type, and task assignment complied with a positional constraint (i.e. L-, R-, or E-type). The DCS algorithm was proposed for the problem. Since two subproblems had to be considered, i.e. task-to-workstation and robot-to-workstation allocations, DCS was modified into the cooperative co-evolutionary cuckoo search (CoCS) algorithm. The benchmark problems used to test the effectiveness of DCS and CoCS were small—(9, 12, 16, and 24 tasks) and large—(65, 148, and 205) sized. DThe CS and CoCS were evaluated against the SA, ABC, GA, PSO, and CoPSO algorithms in terms of the RPD. The results indicated that DCS and CoCS significantly outperformed the other algorithms, while CoCS was better than DCS in the overall average RPD. Also, new upper bounds were discovered by CoCS and DCS in many instances.

PAL

Çil et al. (2017b) addressed the RALBP type II under the PAL layout aiming at minimising the cooperative cycle time under a given number of workstations located in the PAL. This could be translated into two decisions, i.e. assigning proper robots to workstations and balancing neighbouring lines. In their research, different single products were assumed to be produced in each line of the PAL. Moreover, robots were allowed to work on the adjacent sides of the PALs. To illustrate the problem complexity, the MILP formulation was formalised. For the large-sized problems (PAL with two neighbouring lines), the beam search technique was used as a basis for developing the three metaheuristics of cutting BIBS (CBIBS), best search method based on IBS (BIBS), and iterative beam search (IBS) algorithms. Their effectiveness was evaluated against the DE, CS-PSO, and PSO algorithms. The results showed that BIBS outperformed IBS and CBIBS in searching for a high solution quality. In contrast, CBIBS could reach final solutions faster than the other two proposed algorithms. Moreover, the proposed algorithms were far more effective than DE, PSO, and CS-PSO.

MAL

Lopes et al. (2017) studied the RALBP of the *MAL* Type II which used *resistant robotic spot welders* in the body shop (body-in-white) of Renault's car factory in Brazil. More than one robotic welder was allowed to work simultaneously in each workstation (multi-manned workstations). The performances of robots may not be the same due to their models and installed welding tool sizes. Apart from the welding time, the *movement and positioning times of the robot* (i.e. workpiece handling and welding gun positioning times) and interferences between robots (i.e. multiple robots in each workstation) were also considered since the welding time was short compared to the cycle time. Forty-two robots distributed symmetrically among 13 workstations were operated in the factory to perform more than 700 welding points. Although the line produced four mixed-model products, the problem was simplified by optimising four single models separately instead. Also, no precedence relationships between welding points were assumed. Over 700 welding points needed to be spotted on each car model. The *MILP technique* was used in modelling the problem. For system constraints, the interference constraints, movement times, assignment restrictions, and robot-wise variations on parameters were taken into account. The results showed that the assembly line could produce around 6.6% more vehicles than its current yield.

Table 1 summarises the main contribution of the RALBP literature presented in chronological order, while the research contributions from the RALBP literature in conjunction with the temporal sequence, to demonstrate the major additional contributions from its previous direct predecessor, is shown in Fig. 2.

Research findings

The RALBP literature was comprehensively reviewed in "Literature on RALBP" section. In this section, some interesting aspects and their associated statistics are articulated as follows.

- Since the first issue of the RALBP published by Rubinovitz and Bukchin (1991), up to now (nearly 30 years) only 33 articles have been published in international scientific journals and international conferences. The number of papers chronologically published by year is shown in Fig. 3. The average number of published papers per year is around 1.2 (during 1991–2019) but in the past five years (during 2015–2019) the number had jumped to 4.4 papers per year and accounted for 67% of the total number of released papers. There were no research publications on the RALBP during the eight consecutive years from 1998–2005, partly because the RAL technology was still expensive at that time compared with the wages and availabilities of human workers. In contrast, the return of this research area during the last few years may be the result of worldwide awareness in the concept of Industry 4.0 as well as the affordable prices of industrial robots and its supported technologies.
- Since the manual straight-shaped assembly line was first invented in 1913 by Henry Ford, new layouts of the assembly lines have been developed constantly to increase productivity gains and save manufacturing costs. The RALs also adopt the aforementioned scheme to increase their system effectiveness. Up until now, six layouts of the RALs have been noticeable in the RALBP literature (Fig. 4), i.e. StAL (Rubinovitz and Bukchin 1991), PWAL (Tsai and Yao 1993), UAL (Nilakantan and Ponnambalam 2016), 2SAL (Aghajani et al. 2014), PAL (Çil et al. 2017b), and MAL (Lopes et al. 2017). Note that the name(s) in the parenthesis after the layout type is the name of the person or group of persons who first pioneered the research in such layout (cf. Figure 2). The most studied layout was the StAL (23 papers), followed by UAL (five papers) and 2SAL (four papers). Note that since Nilakantan et al. (2017b, c) referred to two layouts (StAL and UAL) in each of their papers for relative performance comparisons between StAL and UAL, the total frequency count from different layouts is 35 (Fig. 4), not 33 (total number of the papers). Since the first publication in 1993, we have not found any further papers on the PWAL. The reason may be that investing in a large robot capable of completing all required assembly tasks is more economical and effective than buying many small robots to work in parallel. However, this strategy is still practical when customer demands do not increase dramatically and the company is not ready to upgrade the whole production system. Similar to the PWAL, no papers on the PAL and MAL have been found since 2017. However, it is expected that new papers under these two layouts will be released shortly since these two layouts are new interesting research areas in the RALBP.
- The mixed-model RAL is an important manufacturing technique that permits several dissimilar product models to be assembled on the same assembly line without changeovers. This practice is applied in many industries today to respond to a vast variety of customer demands. It was observed that this predominant practice was hardly adopted in the RALBP since more than 80% of the research still assumed that the RALs were balanced for a single finished product (Fig. 5). Only a few research papers applied this assumption in their problem definitions, i.e. StAL (three papers), UAL (one paper), 2SAL (one paper), and MAL (one paper), whereas none has been found in the PAL and PWAL. This indicates that most researches in the RALBP still focus on the utilisation of the RAL for fixed automation where the production sequence of products is fixed rather than flexible, which is the true need in today's world. Implementing the

Table 1 Contribution of the research publications based on the chronological order of publications

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Weckenborg and Spengler (2019)	StAL	Human and robot can execute tasks at the same workpiece in parallel or collaboratively	–	High-pressure cleaner	Single	Variable deterministic depending on the process alternatives	–	Type Cost	Min (*cost per cycle*)	MIP	EX (MIP)	Task allocation & robot/worker collaboration allocation
Dalle Mura and Dini (2019)	StAL	Human and robot can execute tasks at the same workpiece in parallel or collaboratively	–	*Scooter chassis* (*Real case*)	Single	Variable deterministic depending on the process alternatives	–	Type *Cost*	1. Min (Assembly line cost) 2. Min (number of skilled workers) 3. Min (energy workload variance)	–	MH (GA)	Task allocation & robot/worker collaboration allocation and equipment selection
Weckenborg et al. (2019)	StAL	*Human and robot can execute tasks at the same workpiece in parallel or collaboratively*	–	–	Single	Variable deterministic depending on the process alternatives	–	Type II	Min (cycle time)	MIP	EX (MIP), MH (GA)	Task allocation & *robot/worker collaboration allocation*
Janardhanan et al. (2019)	StAL	–	Sequence-dependent setup time	–	Single	Variable deterministic depending on the allocated robot type	–	Type II	Min (cycle time)	MIP	EX (MIP), H (MBO)	Task allocation & robot allocation
Borba et al. (2018)	StAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	Type II	Min (cycle time)	MIP	EX (MIP), EX (*BBR*), H (IBS)	Task allocation & robot allocation

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Li et al. (2018a)	2SAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	Positional constraint (L-, R-, or E-type)	Type II	Min (cycle time)	–	MH (DCS), MH (CoCS)	Task allocation & robot allocation
Li et al. (2018b)	StAL	–	–	–	*Mixed*	Variable deterministic depending on the product model and allocated robot type	–	*Type O*	*Min* (*makespan*)	MIP	EX (MIP), MH (RSA), MH (RCoGA)	Task allocation & robot allocation & *mixed-model sequencing*
Pereira et al. (2018)	StAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	*Type Cost*	*Min* (*sum of fixed and variable costs*)	0–1 IP	EX (0–1 IP), MH (MA)	Task allocation & robot allocation
Çil et al. (2017)	StAL	–	–	–	*Mixed*	Variable deterministic depending on the allocated robot type	–	Type II	*Min* (*sum of model cycle times*)	MIP	EX (MIP), H (BS)	Task allocation & robot allocation
Çil et al. (2017b)	*PAL* (*first*)	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	Type II	*Min* (*joint cycle time*)	MIP	EX (MIP), MH (IBS), MH (BIBS), MH (CBIBS)	Task allocation (balancing of neighbouring lines) & robot allocation

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Lopes et al. (2017)	*MAL* (*first*)	–	*Accessibility limitation, movement time and interference between robot constraints*	*Robotic spot welders in the body shop of Renault's factory* (*automotive industry*) *in Brazil* (*Real case*)	Mixed	Variable deterministic depending on the allocated robot type	*Positional constraint*	Type II	Min (cycle time)	*MIP*	EX (MIP)	*Assignment of welding procedures to robots, robot part routing, and robot coordination*
Nilakantan et al. (2017a)	StAL	–	–	–	Single	Variable deterministic and depended on the allocated robot type	–	Type E	1. Min (*total carbon footprint*) 2. Max (line efficiency) (*result comparison using Pareto fronts*)	MIP	MH (MOCC)	Task allocation & robot allocation & *carbon footprint*
Nilakantan et al. (2017b)	StAL & *UAL*	–	–	Auto engineer cradle, hot tank, electronics, refrigerator, small utility vehicles, and engine assemblies	Single	Variable deterministic and depended on the allocated robot type	–	Type II	Min (cycle time)	0–1 IP	EX (0–1 IP), MH (PSO)	Task allocation & robot allocation
Nilakantan et al. (2017)	StAL & UAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	Type II & Type Cost	1. Min (cycle time) 2. Min (total assembly line cost) (*time-based and cost-based models*)	0–1 IP	MH (DE)	Task allocation & robot allocation & carbon footprint

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Çil et al. (2016)	StAL	–	–	–	Mixed	Variable deterministic depending on the allocated robot type	–	*Type II*	*1. Min (cycle time)* *2. Min (number of workstations)* *3. Min (robot utilisation cost)*	GP	EX (*GP*)	Task allocation & robot allocation
Li et al. (2016a)	2SAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	Positional constraint (L-, R-, or E-type)	Type II	Min (cycle time)	MIP	EX (MIP), MH (C-PSO)	Task allocation & robot allocation
Li et al. (2016b)	2SAL	–	–	Truck assembly lines of an automotive company (referred to the literature)	Single	Variable deterministic depending on the allocated robot type	Positional constraint (L-, R-, or E-type)	Type II	1. Min (cycle time) 2. *Min (energy consumption) (result comparisons using Pareto fronts)*	MIP	MH (RSA)	Task allocation & robot allocation
Nilakantan & Ponnambalam (2016)	*UAL (first)*	–	–	Engine cradle line of an automotive manufacturer (referred to the literature)	Single	Variable deterministic and depended on the allocated robot type	–	Type II	Min (cycle time)	0–1 IP	EX (0–1 IP), MH (PSO)	Task allocation & robot allocation

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Rabbani et al. (2016)	UAL	–	–	–	*Mixed*	Variable deterministic depending on the allocated robot type	–	Type II	1. *Min* (*robot purchasing cost*) 2. *Min* (*setup robot cost*) 3. *Min* (*sequence-dependent setup cost*) 4. Min (cycle time) (*result comparison using Pareto fronts*)	MIP	EX (MIP), MH (NSGA II), MH (MOPSO)	Task allocation & robot allocation
Nilakantan et al. (2016)	UAL	–	–	–	Single	Variable deterministicdepending on the allocated robot type	–	*Type E*	*Max* (*line efficiency*) *by min* (*total energy consumption of the robots at each workstation*)	–	MH (PSO), MH (DE)	Task allocation & robot allocation
Nilakantan et al. (2015a)	StAL	–	–	Automobile body shop	Single	Variable deterministic depending on the allocated robot type	–	Type II	1. Min (cycle time) 2. Min (total energy consumption)	0–1 IP	MH (PSO)	Task allocation & robot allocation & *energy consumption*
Nilakantan et al. (2015b)	StAL	–	–	Engine cradle line of an automotive manufacturer	Single	Variable deterministic depending on the allocated robot type	–	Type II	Min (cycle time)	0–1 IP	EX (0–1 IP), MH (PSO), MH (CS-PSO)	Task allocation & robot allocation

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Aghajani et al. (2014)	*2SAL* (*first*)	–	*Sequence-dependent setup times between tasks and setup times for each robot*	–	*Mixed*	Variable deterministic depending on the allocated robot type	Positional constraint (L-, R- or E-type)	Type II	Min (cycle time)	MIP	EX (MIP), MH (SA)	Task allocation & robot allocation
Daoud et al. (2014)	StAL	–	–	*Automated packaging line dedicated to dairy food products using pick-and-place robots* (*real case*)	Single	Variable deterministic depending on the allocated robot type	–	*Type E*	*Max* (*line efficiency*) *calculated by the number of products seized by robots*	–	MH (ACO-GLS), MH (PSO-GLS), MH (GA-GLS)	Task allocation & robot allocation
Nilakantan and Ponnambalam (2012)	StAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	Type II	Min (cycle time)	–	MH (*PSO*)	Task allocation & robot allocation
Yoosefelahi et al. (2012)	StAL	–	*Some robot type may not be able to perform a certain task*, *robot setup times*	–	Single	Variable deterministic depending on the allocated robot type	–	*Type II*	1. Min (cycle time) 2. *Min* (*robot setup costs*) 3. *Min* (*robot costs*) (*result comparison using Pareto fronts*)	*MIP*	MH (CMOES), MH (PAES), MH (HMOES)	Task allocation & robot allocation

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Gao et al. (2009)	StAL	–	*Fixed number of robots*	–	Single	Variable deterministic depending on the allocated robot type	–	Type II	Min (cycle time)	INLP	EX (INLP), MH (hGA)	Task allocation & robot allocation
Levitin et al. (2006)	StAL	–	–	–	Single	Variable deterministic depending on the allocated robot type	–	*Type II*	Min (cycle time)	–	MH (*GA + constructive assignment procedure*), MH (*GA + recursive assignment procedure*)	Task allocation & robot allocation
Hong and Cho (1997)	StAL	–	–	*Electrical relay and automobile alternator industrial products* (*2 real cases*)	Single	Fixed deterministic	–	Type I	Min (number of workstations)	–	MH (SA)	*Assembly sequence* & line balancing
Kim & Park (1995)	StAL	–	*Limited storage space for parts and robot hand slot capacity*	*VCR-deck robotic assembly line* (*real case*)	Single	Fixed deterministic	–	Type I	Min (number of robot cells)	*IP*	EX (IP), H (strong cutting plane)	Task allocation & *part and tool required to perform tasks assignments*
Tsai & Yao (1993)	*PWAL* (*first*)	–	*More than one robot of the same type was allowed to operate in each workstation*	–	Single	Variable deterministic depending on the allocated robot type	–	*Type O*	*Min* (*weighted average standard deviation of all workstation output rates*)	*IP*	EX (IP), *Simulation*	*Robotic capacity planning*

Table 1 (continued)

Author	Layout	Man	Machine (Robot)		Material (Task)			Method (Problem/Decision)				
		Human involvement	Robot related constraints	Real business line	Number of products	Task time variation	Task assignment control	Problem type	Number of objectives	Problem formulation	Solution technique	Simultaneous decision
Rubinovitz et al. (1993)	StAL	–	–	–	Single	Variable deterministic and depended on the allocated robot type	–	Type I	Min (cycle time)	–	*H (B&B based on the frontier search strategy and search limiting heuristic rules)*	Task allocation & robot allocation
Rubinovitz and Bukchin (1991)	*StAL (first paper)*	–	–	–	*Single*	*Variable deterministic depending on the allocated robot type*	–	*Type I*	*Min (cycle time)*	*IP*	–	*Task allocation & robot allocation*

RAL with fixed automation could hinder the adaptability and responsiveness, which are the intrinsic features available in the RAL, resulting in poor customer service and satisfaction.

- From Table 1, apart from the papers presented by Kim and Park (1995) and Hong and Cho (1997), which assumed fixed and deterministic task times, the remainders have used the assumption that task times were a deterministic variable and that their values were based on the robot type selected to execute them. The former assumption is true when only one type of robot is available in the production system. In contrast, if more than one type of robot is ready to engage in assembly tasks, due to their different capabilities, the latter assumption of deterministic variable task times will be justified. For these reasons, most articles have adopted the latter assumption in their problem definitions. Moreover, the zoning constraint, which could be encountered in the real industry, in terms of positive zoning, negative zoning, and synchronous operational constraints of tasks, has never been considered in the RALBP modelling at all.
- The objectives to be optimised in the RALBP are diverse and hence the classification scheme (Type) of the objectives is needed to group the research in the same direction into the same category. Figure 6 shows that the majority of the RALBP have been conducted on Type II (23 times), followed by Type Cost (five times), Type I (four times), Type E (three times) and Type O (twice). Thus, most papers on the RALBP have been focussed on Type II rather than Type I, as appeared in the conventional ALBP. The practical reason may reflect the high capital investment of robots where after one product reaches the end of its life cycle the factory has to reprogram the existing robots for producing another new product. As a result, the ALBP Type II that targets at optimising the cycle time, given the number of workstations and robots, was widely adopted. All five Types of problems were attempted in the StAL. The problems of Type II, Type E, and Type Cost were found in the UAL. However, the research in the 2SAL, PAL, and MAL just focused on Type II only. Also, most papers on the RALBP attempted to optimise only a single objective (Fig. 7). For the multi-objective optimisation, the number of objectives ranged from two to four and it was found in the StAL, UAL, and 2SAL only. The concept of the Pareto optimum was used in Yoosefelahi et al. (2012), Li et al. (2016b), Rabbani et al. (2016), and Nilakantan et al. (2017a) to demonstrate the trade-offs among several objectives being optimised simultaneously (Table 1), whereas the remaining papers applied the weighted sum method for multi-objective optimisation (Table 2).
- Mathematical models were used in the model formulation of the RALBP to describe the relationships among vari-

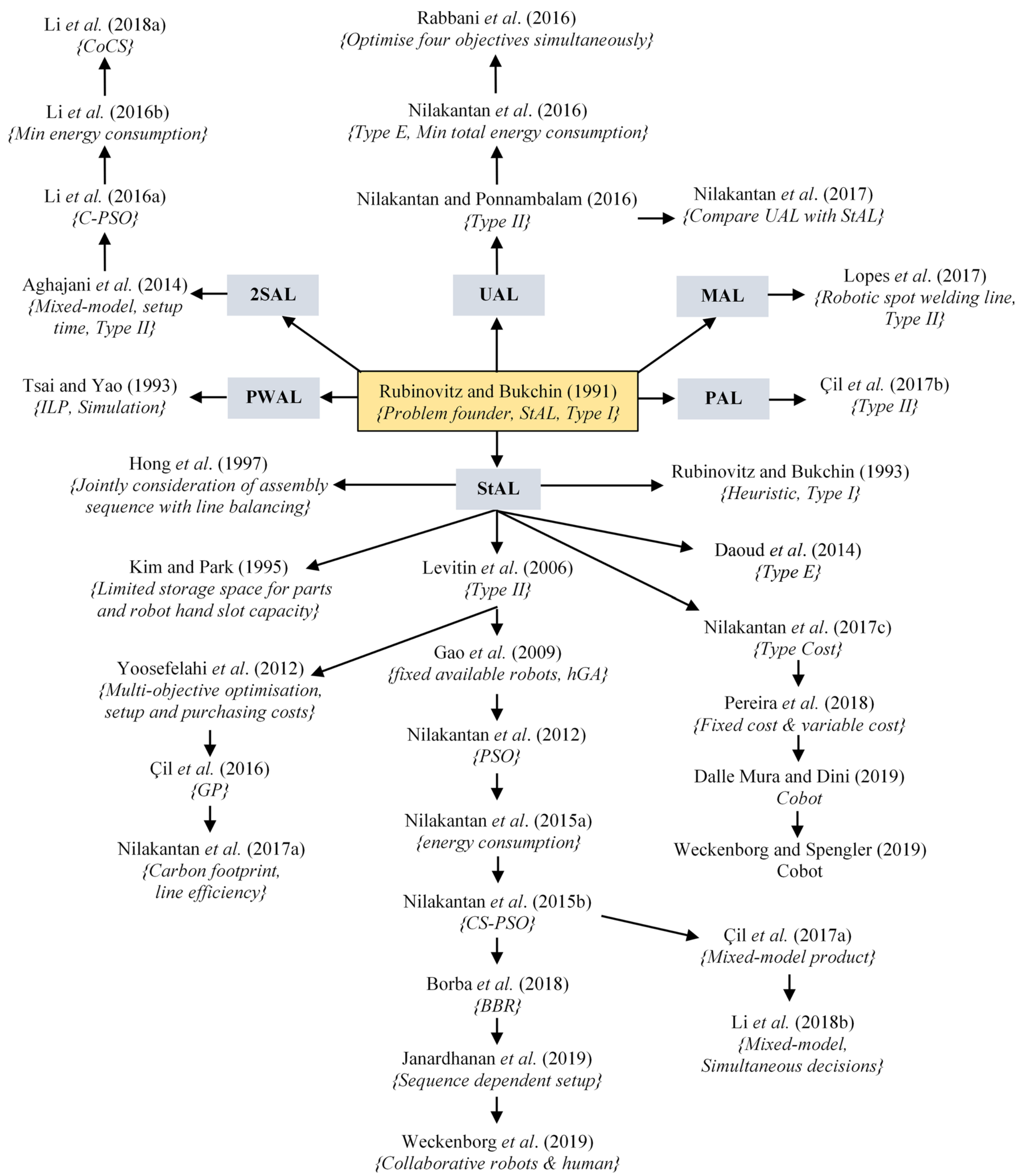

Fig. 2 Precedence diagram of the research contributions

ous variables in the problem at hand. The most-used analytical technique to formulate the RALBP was MIP (15 papers), followed by IP (10 papers), LNIP (one paper), and GP (one paper). Figure 8 shows that various formulation techniques were employed in the StAL. In contrast, only one analytical technique was used in the PWAL (IP), 2SAL (MIP), PAL (MIP), and MAL (MIP). To formulate a mathematical model of the RALBP, the first step was to set up the objective function that needed to be optimised. Some objective functions in the RALBP were inherited from the conventional ALBP, whereas others only existed in the RAL. To ease the future research direction, all objective functions employed in the RALBP literature based on different layouts are summarised in Table 3. Note that the information in Table 3 was extracted from the holistic view of Table 1 and then rearranged to demonstrate the relationship between the number and types of objectives applied in the RALBP and the layout types. This presentation structure would be useful in the search for research gaps since it could suggest which objective

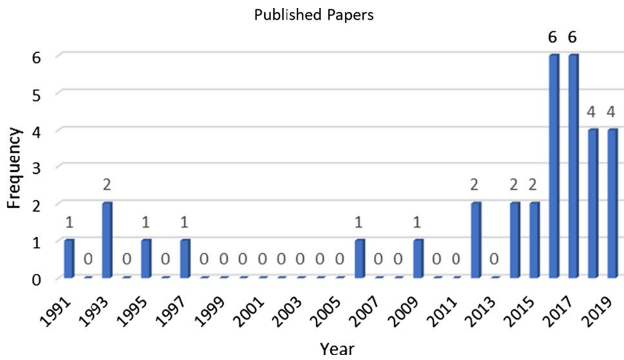

Fig. 3 RALBP research papers published each year since 1991

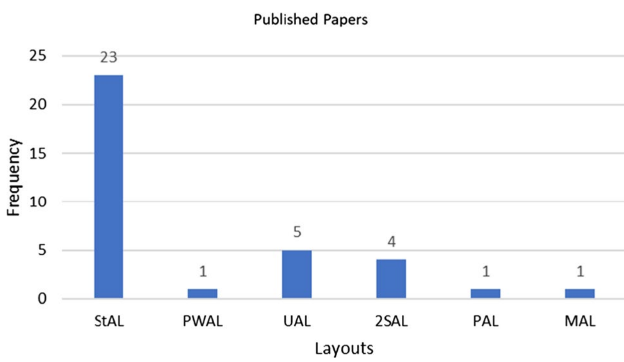

Fig. 4 Number of papers published in each layout since 1991

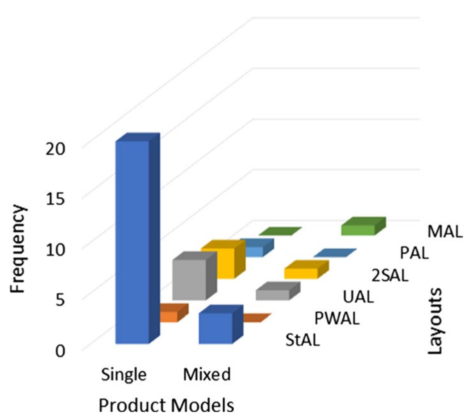

Fig. 5 Number of product models produced in each layout

has never been attempted under which layout. As a result, to bridge such gaps, the mathematical models in the untouched areas of the RALBP would be formulated in the future. As mentioned in "Classification of RALBP" section, the basic model of the RALBP (Type I) was typically formulated with IP in which the two decision variables used were (1) the assignment of a specific task to a certain workstation and (2) the assignment of a specific robot to a certain workstation. The type of mathematical

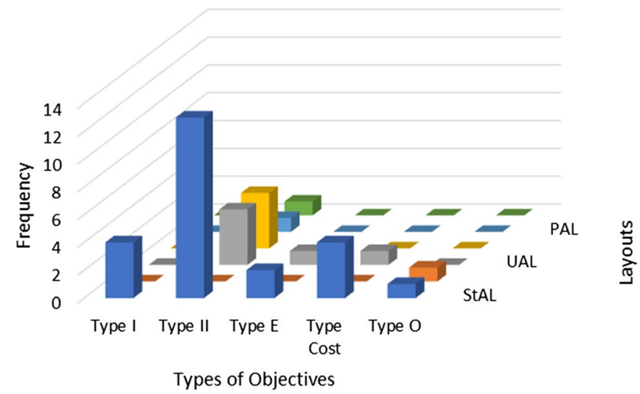

Fig. 6 Types of objectives used in each layout

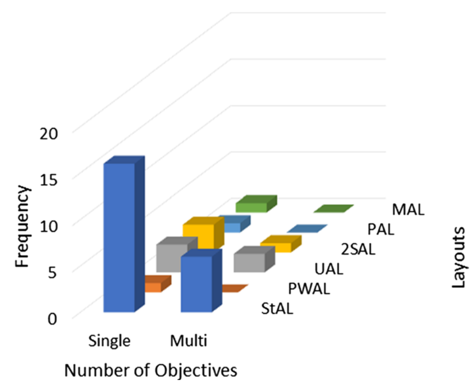

Fig. 7 The number of objectives to be optimised in each layout

formulation could be altered if the objective to be optimised in the RALBP was changed. For example, if we need to minimise the cycle time of the RALBP Type II, the most popular problem of the RALBP, the MIP formulation would be applied. However, two crucial decisions as used in Type I still be necessary for the formulation of Type II's models. Also, to extend the scope of the RALBP to encompass wider operational perspectives, e.g. energy efficiency, carbon footprint, line efficiency, etc., additional decision variables would be needed in the model formulation. Table 4 summarises the decision variables used in the MIP formulation and the reasons why they were applied. This information is useful as a starting point in the model formulation.

- The solution techniques used to solve the RALBP presented in ascending order counted by the frequency found were as follows (Fig. 9), i.e. metaheuristics (39), exact solution (20), heuristic (4) and simulation (1). The exact solution techniques claimed their effectiveness only on small-sized problems and they were normally used to demonstrate the complexity of the problems. As a result, for the problem with practical sizes, metaheuristics dominated the other methods. Figure 10 shows that PSO was

Table 2 Research gaps of the RALBP

Layout	Human involvement	Robot related constraint	Real business line	Mixed model	Task time variation	Positional zoning	Positive and negative zoning	Synchronous constraint	Type I	Type II	Type E	Type Cost	Type O	Multiple objectives	Novel metaheuristics	Simultaneous decision
	Man	Machine		Material					Method							
StAL	○	○	○	○	●	NA	●	●	○	○	○	○	○	○	●	○
PWAL	●	●	●	●	●	NA	●	●	●	●	●	●	○	●	●	●
UAL	●	○	●	○	●	NA	●	●	●	○	○	○	●	○	●	○
2SAL	●	○	●	○	●	○	●	●	●	○	●	●	●	○	●	●
PAL	●	●	●	●	●	NA	●	●	●	○	●	●	●	●	●	●
MAL	●	○	○	○	●	●	●	●	●	○	●	●	●	●	●	○
HL	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

○ = moderately necessary; ● = highly necessary; NA = not applicable

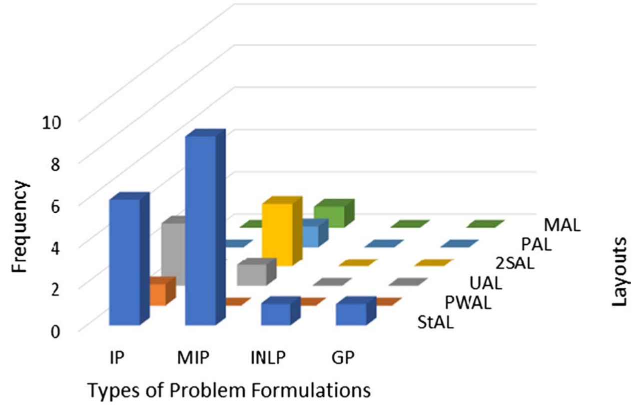

Fig. 8 Analytical techniques used in the RALBP formulations

the most-use metaheuristic for the RALBP, which was found in seven papers, followed by GA (three papers). Among several heuristics, seven of them were exercised in multi-objective search space and evaluated under non-dominated Pareto fronts, i.e. MOCC, RSA, NSGA II, MOPSO, CMOES, PAES, and HMOES. Albeit their differences in details of operations, metaheuristics were run under a similar basic algorithmic structure, i.e. (1) solution encoding (chromosome representation), (2) fitness function definition, (3) initial population creation, (4) population fitness evaluation, (5) new population creation, (6) old population replacement by the new one, (7) repeat steps (4)–(6) until the terminating condition was reached, and (8) decoded the best chromosome to obtain a solution.

- Cooperative robots (cobots) has recently emerged as a new trend in the RALBP. The intended utilisation of cobots was to increase productivity in manual assembly lines. This new problem was configured by allowing a worker and a cobot to share a common workstation to execute tasks at the same workpiece simultaneously in a collaborative or parallel manner. Since the execution times of the human–robot collaboration were faster than human workers, various synergistic advantages from manual and automated production systems could be gained, especially increased productivity and effectiveness. On the other hand, workers could also be relieved from repetitive ergonomically stressful tasks. Due to possible collaborative works between human workers and cobots in shared workstations, the sequence-dependent start times (similar to the MAL) between the two resources must be strictly observed while making line balancing decisions, which then enhances the complexity of the problem compared to the original RALBP. Since this research area was quite new, only three published papers were revealed (Table 1), i.e. Weckenborg et al. (2019), Weckenborg and Spengler (2019), and

Table 3 Objectives applied in optimising the RALBP based on the line layouts

Layout	Objective	Author
StAL	Max (line efficiency)	Nilakantan et al. (2017a), Daoud et al. (2014)
	Min (assembly line cost)	Dalle Mura and Dini (2019)
	Min (cost per cycle)	Weckenborg and Spengler (2019)
	Min (cycle time)	Janardhanan et al. (2019), Borba et al. (2018), Nilakantan et al. (2015b, 2017a, b), Çil et al. (2016), Nilakantan and Ponnambalam (2012), Yoosefelahi et al. (2012), Gao et al. (2009), Levitin et al. (2006), Rubinovitz et al. (1993), Rubinovitz and Bukchin (1991)
	Min (energy workload variance)	Dalle Mura and Dini (2019)
	Min (makespan)	Li et al. (2018b)
	Min (number of robot cells)	Kim and Park (1995)
	Min (number of skilled workers)	Dalle Mura and Dini (2019)
	Min (number of workstations)	Hong and Cho (1997), Çil et al. (2016)
	Min (robot costs)	Yoosefelahi et al. (2012)
	Min (robot setup costs)	Yoosefelahi et al. (2012)
	Min (robot utilisation cost)	Çil et al. (2016)
	Min (sum of fixed and variable costs)	Pereira et al. (2018)
	Min (sum of model cycle times)	Çil et al. (2017)
	Min (total assembly line cost)	Nilakantan et al. (2017)
	Min (total carbon footprint)	Nilakantan et al. (2017a)
PWAL	Min (weighted average standard deviation of all workstation output rates)	Tsai & Yao (1993)
USAL	Max (line efficiency)	Nilakantan et al. (2016)
	Min (cycle time)	Nilakantan et al. (2017a, b), Nilakantan and Ponnambalam (2016), Rabbani et al. (2016)
	Min (robot purchasing cost)	Rabbani et al. (2016)
	Min (sequence-dependent setup cost)	Rabbani et al. (2016)
	Min (robot setup cost)	Rabbani et al. (2016)
2SAL	Min (cycle time)	Li et al. (2016a, b, 2018a), Aghajani et al. (2014)
	Min (energy consumption)	Li et al. (2016b)
PAL	Min (joint cycle time)	Çil et al. (2017b)
MAL	Min (cycle time)	Lopes et al. (2017)

Dalle Mura and Dini (2019). Because of the existence of human workers in the system, the ergonomic risk assessed by the energy expenditure of workers was also taken into account. Besides, it appeared that only Dalle Mura and Dini (2019) considered the different skills of workers and different dexterities of robots. Two mathematical models were proposed, one to optimise the cycle time (Type II) and the other to optimise the cost per cycle (Type Cost). Because the problem was too complex to solve by traditional mathematical approaches, GA was employed to handle real-life problems. This research area left a vital conclusion about the benefits of deploying collaborative robots to help strengthen the work efficiency of the manual assembly lines.

- When tasks were assigned to workstations in the RAL, parts and tools necessary to execute them must also be supplied accordingly. This imposed additional constraints to the RALBP. For example, the available robot hand slots could restrict the number of tools to be allocated to a robot. Various tools could be used to complete the same task but with different effectiveness and operational cost. Moreover, the storage space (e.g. tray capacity) of a robot was normally limited which restricted the number of parts to supply to the robot. Apart from the aforementioned workplace-related constraints, robots themselves could impose additional constraints into the model, e.g. fixed number of available robots, robot setup time/cost, robot movement time, and interference, etc. It appeared that only seven out of 33 papers integrated some of these constraints into their models. This reflects the fact that most of the RALBP research has not yet taken into account the actual characteristics and environment of the robotic assembly system in model development.
- Under the environment of mixed-model RALs, more than one model of products was concurrently assembled on the same line. However, the model sequencing problem, which was strongly interrelated with the mixed-model RALBP, had a significant impact on the

Table 4 Additional decision variables applied for the MIP formulation

Decision variable	Author	Reasons for use
Cycle time	Yoosefelahi et al. (2012), Li et al. (2016a, b), Nilakantan et al. (2017a), Lopes et al. (2017), Weckenborg et al. (2019), Janardhanan et al. (2019)	To minimise the cycle time
Cycle time of the specified model	Aghajani et al. (2014), Rabbani et al. (2016), Çil et al. (2017)	To minimise the cycle time of the mixed-model RALBP
Finish time of task of a specified model	Aghajani et al. (2014), Li et al. (2016b)	To represent the precedence relationship on the mated-stations in the 2SAL
Total energy consumption	Li et al. (2016b), Nilakantan et al. (2017a)	To minimise the total energy consumption on all workstations
Energy consumption on a given side of a specified mated-station	Li et al. (2016b)	To calculate the energy consumption on a given side of a specified mated-station
Line efficiency	Nilakantan et al. (2017a)	To minimise the line efficiency
Carbon footprint by energy consumption	Nilakantan et al. (2017a)	To minimise total carbon footprint by energy consumptions on all workstations
Energy consumption on a specified workstation	Nilakantan et al. (2017a)	To calculate the energy consumption on a specified workstation
Operation energy consumption on a specified workstation	Nilakantan et al. (2017a)	To calculate the operation energy consumption on a specified workstation
Standby energy consumption on a specified workstation	Nilakantan et al. (2017a)	To calculate the standby energy consumption on a specified workstation
Robot cycle time	Lopes et al. (2017)	To calculate the cycle time of a specified robot
Joint cycle time	Çil et al. (2017b)	To minimise the joint cycle time of the PAL
Completion time of all the tasks	Li et al. (2018b)	To minimise makespan
Finishing time of the task	Li et al. (2018b)	To represent the precedence relationship
Total operation time of tasks on a given station for a specific model	Li et al. (2018b)	To evaluate the total operation time of tasks on a given station for a specific model
Start time of a specific task in its respective station	Weckenborg and Spengler (2019)	To represent the precedence constraint

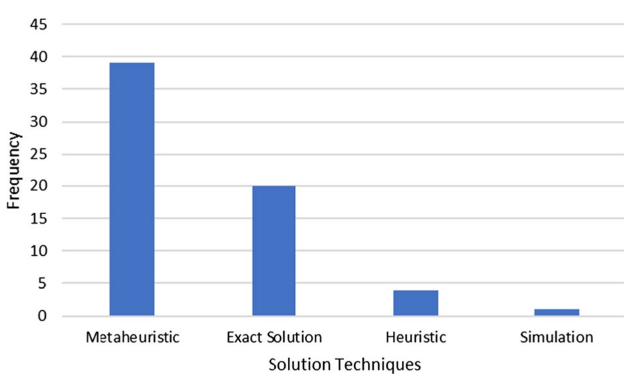

Fig. 9 Solution techniques utilized to solve the RALBP

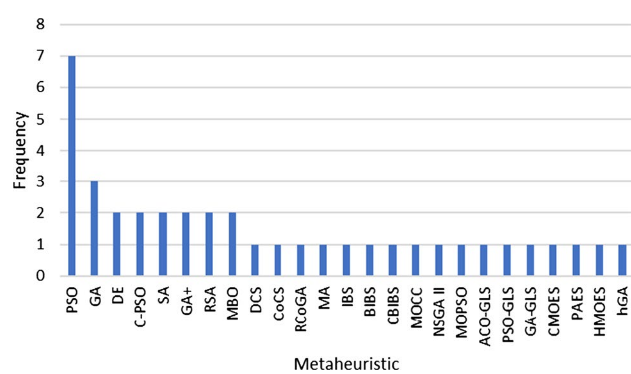

Fig. 10 Metaheuristics used to solve the RALBP

line effectiveness. The model sequencing decision determined the sequence of product models to be executed by the mixed-model RAL. Normally, these two problems were solved independently in a hierarchical manner, firstly the line balancing and then the model sequencing. However, a more effective approach was to solve them simultaneously to achieve a synergistic objective. As a result, three sub-problems needed to be optimised in the robotic assembly line balancing and sequencing problems, i.e. robot assignment, task assignment, and model sequencing. Although this problem was widely addressed in the traditional ALBP, only one paper (Li et al. 2018b) tackled this problem in the StAL.

- Energy cost is a high proportion of the cost structure in modern manufacturing systems where automation is used to replace human workers. Manufacturers would gain a higher competitive ability if they could increase the energy efficiency in their production systems. In the RAL, the most energy-consuming device was the robots, whose energy source was normally based on electricity. As a result, the selection of an appropriate robot to perform the tasks with the lowest energy consumption became a vital energy-saving measure. Three papers addressed this issue explicitly in their respective model formulations of StAL (Nilakantan et al. 2015b), UAL (Nilakantan et al. 2016), and 2SAL (Li et al. 2016b). These three papers assumed that robots were the main energy consumption of the RALs. Also, robots consumed energy both during operation and on standby, with energy consumption during the standby period being approximately 10% of the consumption during operation. The total energy consumption of each robot was computed from the duration that the robot was in operation multiplied by the operation energy consumption per time unit plus the duration that the robot was on standby multiplied by the standby energy consumption per time unit. The total energy consumption of the RAL was computed from the summation of the energy consumption of all the robots. As a result, to minimise the total energy consumption of the RAL, under a given cycle time, the total duration that robots were in operation had to be minimised by assigning the most suitable robot to perform the allocated tasks with the least total task time in each workstation.
- Only one paper has addressed the issue of environmental awareness in the RALBP with the StAL configuration (Nilakantan et al. 2017a). Under the context of the RAL, the amount of carbon emission released into the atmosphere (carbon footprint) was assumed to be created solely by robots powered by electricity. The carbon footprint of the RAL was measured by the total amount of electrical energy consumed by all workstations multiplied by the conversion factor of 0.5488 $kgCO_2/kWh$ (i.e. average electricity carbon footprint of the main power grids). In each workstation, once again the energy usage of a robot was divided into the in operation and on standby modes. The common parameter used in calculating the operation and standby energy consumptions were the total operation times of the tasks that were assigned to the specific robot in the workstation. To minimise the carbon footprint in the RAL, the total carbon footprint released by all robots had to be minimised under a given cycle time.
- To probe the practical value of the RALBP being researched, the number of research papers that embraced a real industrial environment for determining various settings and conditions in the problem definition and testbed instances seems to be a reliable indicator. It turned out that only five papers from a total of 33 published papers employed real industrial cases as a basis for modelling and evaluating their solution techniques. Four of the real cases belonged to the StAL and the remaining one was associated with the MAL (Table 1). The business lines referenced to in the RALBP literature included the body shop in the automotive industry, automated packaging line for dairy food products, electrical relay assembly line, automobile alternator assembly line, VCR-deck line, and scooter chassis assembly line. There are many more real-life applications in industry that employ RAL to perform operations, which is something that researchers in this field should pay more attention to and share their knowledge with others.

Future research directions

Regarding the number and diversity of the recently published articles on the RALBP, it is clear that this research area is still of interest to researchers and practitioners. Many research gaps are waiting to be explored to further enhance the effectiveness of the robotic assembly system and/or to bring the research activities to be closer to that in real practices in the industry. The following points are research guidelines on the RALBP that merit deliberation.

- Almost all research in the RAPBP hypotheses that the system is fully automated and operated without human involvement. This assumption is generally true for large companies that are ready for high financial investment and pursuing the strategy to be a leader over competitors. However, for small and medium enterprises (SME), such huge investment may not be affordable. As a result, robots may only be gradually introduced at some particular sections of the line as necessary to smoothen the production flow by mitigating the bottleneck problem and/or to perform repetitive tasks which may be physically stressful to human workers or tasks involving high ergonomic risk, such as an awkward posture, bending, etc. In this case, human workers still outnumber robots in the assembly line. As a result, in today's factory, we increasingly find robots working alongside with workers to help them execute monotonous, complex, or unsafe operations. Hence, an interesting research area is a study of the semi-RALBP with collaborative work between human workers and robots, in which single workers are assigned to work in some workstations of the RAL (i.e. StAL, PWAL, UAL, and PAL). The benefits resulting from the use of robots to enhance human work, which could lead to a higher productivity gain and lower occupational illness and medical expense, in comparison with

the investment required to buy robots and the costs of operating them in some workstations should be studied further.
- The cobot environment, where one or more workers and robots are assigned to work collaboratively in parallel in the same workstation of the RAL is another interesting RALBP. Three recently published papers adopted this concept in the StAL (Weckenborg et al. 2019; Weckenborg and Spengler 2019; Dalle Mura and Dini 2019). Hence, there is still the need to research this in other layouts (2SAL, MAL, and HL). Since the cobot assembly line balancing problem is a new research area in the RALBP, further research to encompass real industrial needs is necessary. Furthermore, the key intention of utilising cobots is to relieve manual repetitive tasks that induce physical stress in the workers, and so more ergonomics factors, such as fatigue, situation awareness, posture, etc. could be added in the problem formulation to enhance the model validity. Skills, age, and other physical characteristics (e.g. body size and shape, fitness, handicap status) of workers could be taken into consideration as well. Due to the coexistence between workers and cobots, social physiological aspects should be considered in effectively assigning tasks between these two entities. In addition to the ergonomics aspects, the energy consumption and conservation, as well as the environmental friendliness associated with cobots, is worthy of further consideration. Also, the investigation scope should be extended to multi-objective optimisation that reflects the aforementioned aspects.
- To gain more acceptance from practitioners that the RALBP research is truly applicable in industry, the real case studies and research surveys that articulate more detailed characteristics and configurations of the RAL, rather than the simplified ones, seem highly necessary. Not only could these publications demonstrate the validity of research findings, but also, they could dissipate the cutting-edge concepts and useful guidance to practitioners. Moreover, academics could use these publications as a guideline for incorporating new relevant and practical constraints into their future models. Concerning the findings stated previously that around 90% of the RALBP publications used simplified fictitious problems in the study, more publications to bridge this gap are essential, especially in the PWAL, UAL, 2SAL, PAL, and HL. Besides, the workplace-related constraints should be taken into account more in the operational environment to closely reflect practical works in practice, such as fixed eligible number of robots, sequence-dependent robot setup time/cost, robot movement time and interference, tool and accessory (jig and fixture) changing, limited storage space of parts and tools, part transportation between workstations, sequence-dependent task setup time/cost, etc. Undoubtedly, the parts and tools are necessary elements in the RAL. Taking into account parts and tools in the RALBP could enhance the sensibility and acceptability of the model formulation. Furthermore, interactions among these accessories and the robots that manipulate them to execute assembly operations should be explicitly modelled. For example, for the limited number of tools to be used, some tools could be plugged in with a specific robot type only, where the operational performance of robots depends on selected tools, tool setup time, and cost. Then, tradeoffs between tool costs and productivity gains could be observed in the model. Apart from the conventional concepts in production, such as just-in-time, lean management, human factor, or handicapped and multi-skilled workers (if human workers are allowed to work as a part of the semi-RAL); the current world's issues that are recently receiving great attention in the industry, e.g. green production, energy conservation, renewable energy, carbon footprint, etc. should be integrated into the formulation of the RALBP in the future. The layouts that have never considered these constraints before are PWAL, PAL, and HL.
- The assembly lines of the RALBP are mostly assumed to produce a single product to simplify the formulation of the problem, especially when a new layout, configuration or constraint is just taken into account and augmented into the problem. This assumption is acceptable when the line is programmed to produce a low-unit-cost, standardised product in mass. The single-model RAL does not provide agility and responsiveness to meet demand variations and is unable to adapt to short-term changes in volume and product mix. However, the rapidly changing requirements of customers and short product lifecycle that results in the production concept of economies of scale is superseded by economies of scope. To respond quickly to such a volatile market, multi- and mixed-model assembly lines that allow a family of products that share-alike materials, work content, and processes to be assembled on the same line become a more prevalent production approach. As a result, some necessary software and hardware should be installed to upgrade the RAL with fixed automation to become a flexible one to take advantage of the economies of scope. In terms of future research, more attention should be geared towards multi- and mixed-model RALBP, especially with the PWAL, PAL, and HL layouts.
- Under the context of RALs, deterministic robot-dependent task times are normally assumed in the model formulation. However, task time variations resulting from unexpected uncertainties, such as robot/equipment breakdown caused by poor maintenance or defects from raw materials, tool/equipment wear and tear, etc. have yet to be addressed in future research. Moreover, as suggested

earlier, if the human workers and robots are allowed to work in parallel or independently, various human factors that occur while performing manual operations, such as a lack of motivation, fatigue, learning effect, etc. could also be encompassed in the model. Consequently, as appropriate, another approach to represent uncertain task times could be assumed, e.g. task time representation based on some forms of the distribution function, stochastic variable, or fuzzy variable.

- The task assignment constraint is relevant only to the 2SAL and MAL layouts. In 2SALs, the positional task assignment constraint (L-, R-, or E-side task) must be strictly observed since it controls the effective direction of operations on the workpiece. Although this constraint is optional for MALs, often the appropriate direction of work not only affects the work efficiency but also reduces the interference and interruption between workers who work together at the same large workpiece. So far, only one paper has addressed the RALBP under the MA (Lopes et al. 2017). But this issue was still not considered in their model. It would be more interesting if subsequent researches take this constraint into account. Also, the task assignment constraints of positive zoning, negative zoning, and task synchronisation should be embedded into the MAL's configuration to reflect the real-world operations more closely.
- Most RALBP articles are focused on the Type II problem regardless of the layout types. Apart from the StAL, other layouts should pay more attention to the Type I, Type E, Type Cost and Type O problems. The reason that very little research effort is given to these research areas may be due to the complex non-linearities of the model. Besides, the cost-oriented objective that indicates the long-term benefits of the line balancing decisions should be observed to illustrate the correlation among task time, cycle time, flexibility, and robot/equipment cost. New concerns in today's industry that could cause global climate changes, such as energy consumption, carbon footprint, and energy conservation, or pollution, such as air (pm 2.5), noise, etc., should be considered more in the model formulation of the RALBP.
- Most real problems in the industry need to satisfy more than one objective at the same time. However, nearly all researches on the RALBP are still optimised based on a single objective. As a result, more research should be conducted on the multi-objective RALBP to observe trade-off benefits among these conflicting objectives in the complex solution space. Under multi-objective optimisation, if clear domination among different objectives is observed, hierarchical optimising of these objectives one-by-one is acceptable. However, in most cases, the interested objectives are conflicted with one another and no clear domination is evident, resulting in a compromised solution being satisfactory. A Pareto-based optimisation approach is an effective approach in dealing with multi-objective optimisation problems. It is suggested that the Pareto-optimal solution technique be applied to the multi-objective RALBP, in particular PWAL, PAL, MAL, and HL.
- Two important contents in the RALBP literature are the problem formulation and solution technique. Mathematical models, especially IP and MIP, are widely used in the problem formulation to formally define the relationships among the objective function, parameters and variables of the RAL. The solution obtained from solving the mathematical model enables appropriate explanation and prediction of the system behaviour. As summarised in the previous section, the objectives to be optimised in the RAL are diverse. Also, the number of involved parameters and variables depends on the optimised objective and system configuration (i.e. layouts and associated constraints). Due to the richness in the body of knowledge, there is a high demand for a review article that is specifically written about the application of mathematical models in the RAL. Since the RALBP is NP-hard, numerous research articles have revealed that optimising this problem by complete or implicit enumeration (e.g. B&B, dynamic programming) was only practical for very small-sized problems. For the real-world RALBP, metaheuristics seem to be the most promising way to tackle such a hard problem since they are simple and fast to implement. As a result, it is interesting if more effective cutting-edge metaheuristics are proposed for the RALBP. Statistics from the survey showed that PSO, SA, GA, and DE were the most-used metaheuristics in this area. Recently, several effective novel metaheuristics have emerged, e.g. firefly optimisation algorithm (FOA), cuckoo optimisation algorithm (COA), co-evolutionary algorithm (CoEA), etc. These algorithms should be adapted, and their relative performance should be tested against each other. Moreover, hybrid metaheuristics which draw the prominent points of two or more metaheuristics in searching for good quality solutions need further development. In practice, we often observed that two or more objectives are of interest simultaneously. Therefore, multi-objective optimisation metaheuristics are an effective tool to handle the problem. This kind of metaheuristics could be implemented under its standard form or even in the hybrid form in which two or more of them are synergised together. It is observed from the review that hybrid metaheuristics normally perform better than the non-hybrid ones, especially regarding their exploitation and exploration performances. Also, some of them are embedded with an adaptive mechanism to prevent premature convergence by having a mechanism to escape

from local optima. Similar to the problem formulation, a review article on the application of metaheuristics applied to the RAL merits enough to be published as a paper.

- The standard benchmark problems of the RALBP should be established and made available on-line to the research community. These problem instances should encompass all vital characteristics of different layouts. Besides, they should be diverse and varying in their complexities with different sizes (small, medium and large). The complexity of the problems should be indicated through the number of tasks, available types of robots, precedence relationships, cycle time, F_{ratio}, OS, R_F, R_c, and $WEST_{ratio}$. These standard problems could be employed to fairly benchmark the relative effectiveness of newly proposed cutting-edge algorithms since they will exercise on the same standard platform leading to more credible, consistent, and reliable comparisons. Effective lower bounds are highly needed for all layouts of the RAPBP since this information could be a good baseline when the optimum solution is not available. Besides, a public website should be established to keep the source codes of the algorithms proved to be effective in solving the RALBP and offer a free download so that the benchmark test with them can be done more easily.
- Publications on case studies and research surveys that illustrate real-world problem definitions, data, constraints, conditions, requirements, and approaches to handle them during the actual implementation of the RALBP solution techniques are in great demand. The rationality of research findings and practical guidance could also be revealed in these publications. Not only could academics be highlighted with the real RAL environment, but also this information would be taken into account while modelling the RALBP in the future. To gain more acceptance from practitioners, more research must be geared toward answering the actual RALBP rather than these simplified ones currently being done. Also, interviews with practitioners on the applicability of the existing solutions derived from the research would be beneficial, since valuable advice could be given.
- All HLs often claim that these new configurations would bring greater benefits to the production system than those conventional ones. However, none of the RALBP research has been conducted in the production environment. As a result, it is interesting to extend the study of the RALBP to the HLs to further support the claim under both fully automated (operated purely by robots) and semi-automated (human workers and robots collaborate their works) assembly lines. Also, related mathematical models, heuristics and metaheuristics, need to be developed for the RALBP with different HLs.

From the previous discussion, the research gaps in the RALBP that will lead to potential research in the future are pinpointed in Table 2. The symbol ○ demonstrates that the research needs in this aspect are moderate; whereas, the symbol ● indicates the absolute necessity.

Conclusion

The trend in utilising human workers in today's manufacturing world has been changing towards Industry 4.0 due to increased labour wages and the need to improve the productivity and flexibility of the system. Many manufacturers prudently accept this challenge by replacing the manual assembly line with the more effective RAL. One of the key design decisions that need to be addressed throughout the life cycle of the RAL is to optimise the RALBP to achieve a smoother workflow and gain high productivity. The RALBP focuses on the task-to-workstation assignment and selects the most effective robot to execute tasks at each workstation, since individual robots may have different capabilities, to realise a given tactical goal. The RALBP is an extended form of the conventional ALBP where the assembly line is equipped with robots and several automated types of equipment. These robots are authorised to take on all manual operations that set to be executed by human labours. Since the ALBP was established before the RALBP for many decades, it is not unsurprising that the number of research publications and the popularity of the RALBP is not as much as the ALBP. Nevertheless, in recent years, this topic has gained significant attention from researchers.

In this paper, the whole collection of the RALBP articles are compiled. A comprehensive review is presented to demonstrate the academic progress and trends of the subject domain that has been developed continuously. A new classification structure, based on the layout configurations, which is more suitable to the great diversity of the current body of knowledge in this research area, is proposed to classify the RALBP in a way that its subproblems are exhaustive and do not overlap with each other. The 4 M concept is employed to further subdivide each subproblem into several basic attributes of the system. The research contributions, which are chronologically organised, are illustrated and summarised in the Table and graphical formats to help track the review content easily. Finally, the research gap which highlights the direction for future work is addressed. Hopefully, the aforementioned gap between practical and research perspectives will be closed soon.

References

Aghajani, M., Ghodsi, R., & Javadi, B. (2014). Balancing of robotic mixed-model two-sided assembly line with robot setup times. *The*

International Journal of Advanced Manufacturing Technology, 74(5–8), 1005–1016.
Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics, 142*(2), 259–277.
Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science*, *32*(8), 909–932.
Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research, 168*(3), 694–715.
Borba, L., Ritt, M., & Miralles, C. (2018). Exact and heuristic methods for solving the robotic assembly line balancing problem. *European Journal of Operational Research, 270*(1), 146–156.
Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research, 183*(2), 674–693.
Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics, 111*(2), 509–528.
Çil, Z. A., Mete, S., & Ağpak, K. (2016). A goal programming approach for robotic assembly line balancing problem. *IFAC-PapersOnLine, 49*(12), 938–942.
Çil, Z. A., Mete, S., & Ağpak, K. (2017a). Analysis of the type II robotic mixed-model assembly line balancing problem. *Engineering Optimization, 49*(6), 990–1009.
Çil, Z. A., Mete, S., Özceylan, E., & Ağpak, K. (2017b). A beam search approach for solving type II robotic parallel assembly line balancing problem. *Applied Soft Computing, 61,* 129–138.
Dalle Mura, M., & Dini, G. (2019). Designing assembly lines with humans and collaborative robots: A genetic approach. *CIRP Annals, 68*(1), 1–4.
Daoud, S., Chehade, H., Yalaoui, F., & Amodeo, L. (2014). Solving a robotic assembly line balancing problem using efficient hybrid methods. *Journal of Heuristics, 20*(3), 235–259.
Gao, J., Sun, L., Wang, L., & Gen, M. (2009). An efficient approach for type II robotic assembly line balancing problems. *Computers & Industrial Engineering, 56*(3), 1065–1080.
Hong, D. S., & Cho, H. S. (1997). Generation of robotic assembly sequences with consideration of line balancing using simulated annealing. *Robotica, 15*(6), 663–673.
International Federation of Robotics (2018). Executive summary world robotics 2018 industrial robots. https://ifr.org/downloads/press2018/Executive_Summary_WR_2018_Industrial_Robots.pdf. Accessed 1 August 2019.
Janardhanan, M. N., Li, Z., Bocewicz, G., Banaszak, Z., & Nielsen, P. (2019). Metaheuristic algorithms for balancing robotic assembly lines with sequence-dependent robot setup times. *Applied Mathematical Modelling, 65,* 256–270.
Karp, R. M. (1972). Reducibility among combinatorial problems. In *Complexity of computer computations* (pp. 85–103). Boston, MA: Springer.
Kim, H., & Park, S. (1995). A strong cutting plane algorithm for the robotic assembly line balancing problem. *International Journal of Production Research, 33*(8), 2311–2323.
Levitin, G., Rubinovitz, J., & Shnits, B. (2006). A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research, 168*(3), 811–825.
Li, Z., Dey, N., Ashour, A. S., & Tang, Q. (2018a). Discrete cuckoo search algorithms for two-sided robotic assembly line balancing problem. *Neural Computing and Applications, 30*(9), 2685–2696.
Li, Z., Janardhanan, M. N., Tang, Q., & Nielsen, P. (2016a). Co-evolutionary particle swarm optimization algorithm for two-sided robotic assembly line balancing problem. *Advances in Mechanical Engineering, 8*(9), 1687814016667907.
Li, Z., Janardhanan, M. N., Tang, Q., & Nielsen, P. (2018b). Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line. *Engineering Optimization, 50*(5), 877–893.
Li, Z., Tang, Q., & Zhang, L. (2016b). Minimizing energy consumption and cycle time in two-sided robotic assembly line systems using restarted simulated annealing algorithm. *Journal of Cleaner Production, 135,* 508–522.
Lopes, T. C., Sikora, C. G. S., Molina, R. G., Schibelbain, D., Rodrigues, L. C., & Magatão, L. (2017). Balancing a robotic spot-welding manufacturing line: An industrial case study. *European Journal of Operational Research, 263*(3), 1033–1048.
Nilakantan, J. M., Huang, G. Q., & Ponnambalam, S. G. (2015a). An investigation on minimizing cycle time and total energy consumption in robotic assembly line systems. *Journal of Cleaner Production, 90,* 311–325.
Nilakantan, J. M., Li, Z., Tang, Q., & Nielsen, P. (2017a). Multi-objective co-operative co-evolutionary algorithm for minimizing carbon footprint and maximizing line efficiency in robotic assembly line systems. *Journal of Cleaner Production, 156,* 124–136.
Nilakantan, J. M., Nielsen, I., Ponnambalam, S. G., & Venkataramanaiah, S. (2017b). Differential evolution algorithm for solving RALB problem using cost-and time-based models. *The International Journal of Advanced Manufacturing Technology, 89*(1–4), 311–332.
Nilakantan, J. M., & Ponnambalam, S. G. (2012). An efficient PSO for type II robotic assembly line balancing problem. In *2012 IEEE international conference on automation science and engineering* (CASE) (pp. 600–605). IEEE.
Nilakantan, J. M., & Ponnambalam, S. G. (2016). Robotic U-shaped assembly line balancing using particle swarm optimization. *Engineering Optimization, 48*(2), 231–252.
Nilakantan, J. M., Ponnambalam, S. G., & Jawahar, N. (2016). Design of energy efficient RAL system using evolutionary algorithms. *Engineering Computations, 33*(2), 580–602.
Nilakantan, J. M., Ponnambalam, S. G., Jawahar, N., & Kanagaraj, G. (2015b). Bio-inspired search algorithms to solve robotic assembly line balancing problems. *Neural Computing and Applications, 26*(6), 1379–1393.
Nilakantan, J. M., Ponnambalam, S. G., & Nielsen, P. (2017b). Application of particle swarm optimization to solve robotic assembly line balancing problems. In *Handbook of neural computation* (pp. 239–267). Academic Press.
Oztemel, E., & Gursev, S. (2020). Literature review of industry 4.0 and related technologies. *Journal of Intelligent Manufacturing, 31*(1), 127–182.
Pereira, J., Ritt, M., & Vásquez, Ó. C. (2018). A memetic algorithm for the cost-oriented robotic assembly line balancing problem. *Computers & Operations Research, 99,* 249–261.
Rabbani, M., Mousavi, Z., & Farrokhi-Asl, H. (2016). Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem. *Journal of Industrial and Production Engineering, 33*(7), 472–484.
Ramachandram, D., & Rajeswari, M. (2004). Neural network-based robot visual positioning for intelligent assembly. *Journal of Intelligent Manufacturing, 15*(2), 219–231.
Rubinovitz, J., & Bukchin, J. (1991). Design and balancing of robotic assembly lines. In *Proceedings of the fourth world conference on robotics research*, Pittsburgh, PA, 1991.
Rubinovitz, J., Bukchin, J., & Lenz, E. (1993). RALB–A heuristic algorithm for design and balancing of robotic assembly lines. *CIRP Annals, 42*(1), 497–500.
Salveson, M. E. (1955). The assembly line balancing problem. *Journal of Industrial Engineering, 6*(3), 18–25.
Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research, 168*(3), 666–693.

Tasan, S. O., & Tunali, S. (2008). A review of the current applications of genetic algorithms in assembly line balancing. *Journal of Intelligent Manufacturing, 19*(1), 49–69.

Tsai, D. M., & Yao, M. J. (1993). A line-balance-based capacity planning procedure for series-type robotic assembly line. *International Journal of Production Research, 31*(8), 1901–1920.

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., & Spengler, T. S. (2019). Balancing of assembly lines with collaborative robots. *Business Research, 13,* 93–132.

Weckenborg, C., & Spengler, T. S. (2019). Assembly Line Balancing with Collaborative Robots under consideration of Ergonomics: A cost-oriented approach. *IFAC-PapersOnLine, 52*(13), 1860–1865.

Yoosefelahi, A., Aminnayeri, M., Mosadegh, H., & Ardakani, H. D. (2012). Type II robotic assembly line balancing problem: An evolution strategies algorithm for a multi-objective model. *Journal of Manufacturing Systems, 31*(2), 139–151.