Contents lists available at GrowingScience

International Journal of Industrial Engineering Computations

homepage: www.GrowingScience.com/ijiec

Robotic assembly systems planning and scheduling problems: A review

John Andrés Muñoz-Guevara[a*], Eliana Toro-Ocampo[a] and Mario Cesar Vélez-Gallego[b]

[a]Programa de Ingeniería Industrial, Universidad Tecnológica de Pereira, Pereira 660003, Colombia
bDepartamento de Ingeniería de Producción, Universidad EAFIT, Medellín, Colombia

C H R O N I C L E

Article history:
Received March 17 2024
Received in Revised Format June 20 2024
Accepted July 27 2024
Available online July 27 2024

Keywords:
Robotics
Assembly systems
Assembly planning
Assembly scheduling

A B S T R A C T

Evolving market trends, characterized by an increasing demand for personalized products with short life cycles and variable demands, pose a significant challenge to the industry. One of the industry's strategies is to adopt robotic assembly systems to improve productivity and increase system flexibility. The widespread adoption of robots in assembly processes is evident; however, success is not guaranteed with implementation alone. Equally critical is addressing assembly planning and scheduling problems in robotic systems. To facilitate understanding, this review offers, in Section 2, a classification of robotic assembly systems, with an emphasis on a new layout termed the robotic matrix-structure assembly system. Section 3 classifies the planning and scheduling problems applied to the robotic assembly systems. In Section 4, we discuss the approaches and techniques used to formulate and solve the planning and programming challenges. Finally, statistical data are presented to illustrate current research trends and identify gaps for future research.

1. Introduction

One of the most crucial steps in creating a product is planning the assembly; since assembly operations represent more than 20% of manufacturing costs and occupy up to 50% of production time (Pan, 2005). Assembly processes can be performed in different environments, such as manual workstations, automatic systems, robotic cells, or collaborative human-robot cells (Scholz-Reiter & Freitag, 2007). Robotic assembly systems (RAS) are used in the industry to assemble a variety of products with minimal setup and programming times. The assembly costs in an RAS must be kept low for its operation to be profitable. To achieve this objective, researchers have focused on several key problems. These include minimizing the number of tools and fixture changes, reducing the frequency of reorientations during the assembly process, decreasing the cost of equipment operation, and planning the assignment and sequencing of tasks within the RAS. The ultimate goal is to generate programs that minimize the processing time of assembly operations, thereby achieving high levels of efficiency and productivity (Pan, 2005). The basic planning problem in an RAS consists of n products composed of i assembly tasks that must be performed in one or several m assembly workstations with r robots, producing a sequence of products and tasks to meet the desired performance measure. The RAS environment involves various operation-specific conditions. These constraints include movement restrictions to prevent collisions between robots or between robots and humans, degrees of freedom of the robot, precision, and dexterity required to perform specific assembly tasks, availability of tools, cost, speed of work variable, task precedence, simultaneous work on a product, trajectory optimization, and compliance with geometric specifications of operations. This article provides a review of the assembly planning and scheduling problems in RAS with the following structure: The first part delves into the current state and emerging trends of RAS; secondly, it presents the taxonomy of the assembly planning and scheduling problems in different RAS configurations; third, it explores the various approaches developed to address the problems of assembly planning and scheduling in different RAS; and finally, research statistics and future directions are presented.

* Corresponding author
E-mail johandmunoz@utp.edu.co (J. A. Muñoz-Guevara)
ISSN 1923-2934 (Online) - ISSN 1923-2926 (Print)
2024 Growing Science Ltd.
doi: 10.5267/j.ijiec.2024.8.001

This study aims to identify the fundamental components of RAS planning and scheduling and establish a framework for classifying and analyzing the existing literature. To build the research, the following questions were formulated:

- What are the configurations or layouts of the RAS?
- How are RAS planning and scheduling problems classified?
- What types of solution approaches have been developed to solve the RAS planning and scheduling problems?

A bibliometric analysis was conducted given published papers indexed by the Scopus, ScienceDirect, SpringerLink, Taylor & Francis, and IEEE Xplore database. Concretely, we analyzed published records by keywords "robotic assembly line balancing", "robotic assembly cell", "robotic assembly sequence planning, and "robotic assembly path planning" from 1990 to 2023, "robotics assembly task scheduling", "matrix-structure assembly systems", and "matrix manufacturing workshop" from 2010 to 2023. These searches yielded a total of 442 results, many of which were either repetitive or not useful for the review. However, they provided a comprehensive overview of the breadth of the topic and confirmed that no previous literature review had been conducted on the global view of RAS. A total of 112 publications on RAS planning and scheduling problems since 1990 were analyzed. After reading the abstracts, a total of 60 publications were selected as the most relevant for the investigation. A flowchart illustrating the process of finding and choosing pertinent studies for the investigation is presented in Fig 1.

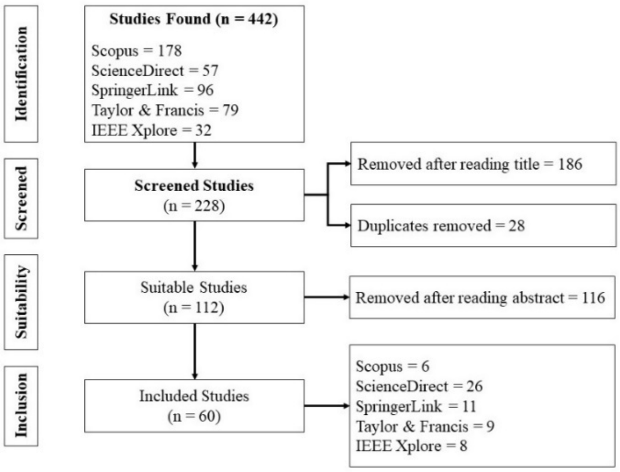

Fig. 1. Flowchart identification and selection of relevant studies

2. Robotic assembly systems

The implementation of flexible assembly systems (FAS) is driven by the need to adapt quickly to the growing demand for customized products characterized by short life cycles and fast delivery times. A FAS is a fully integrated setup comprising a series of assembly workstations connected by a materials handling system and managed by a central computer that allows them to assemble different types of parts (Mohamed et al., 2001). Most studies have concentrated on flexible manufacturing systems (FMS), compared to FAS. Both FAS and FMS are computer-integrated manufacturing systems. However, in the work of (Gultekin et al., 2008), it is established that the FAS and the FMS differ in several aspects:

- In an FMS, work is often performed on one element at a time, while in an FAS, multiple components and parts are assembled simultaneously.
- Compared to the FMS, the FAS requires significantly less time to process each operation. Consequently, compared to the FMS, the FAS has a larger setup time to processing time ratio.
- Comparing the FAS with FMS, the FAS's material handling system is more complicated.

These differences make FAS problems more complex than FMS problems (Abd, 2015). Table 1 lists some of the most important differences between the FAS and FMS. The assembly process is a relatively difficult task in robotic implementations. The wide variety of parts to be assembled, along with the need for different grippers, feeders, and other mechanical devices, can limit the system's flexibility. (Rubinovitz et al., 1993). Flexible robotic assembly systems can be configured in different types: robotic assembly lines, single-robot assembly cells, and multi-robot assembly cells.

Table 1
Differences between FAS and FMS

Features	FAS	FMS
Number of different tasks that can be performed	High	Low
Number of pieces worked per job	Several	A Single
Processing time per part	Short (Seconds)	Long (Minutes/hours)
Material handling	Complex (assembly tasks)	Simple (loading/unloading)
Collaborative work man/machine	High	Low

The configuration of the assembly system depends on different variables, but there are two key variables, the volume of production and the variety of products. (Capacho Betancourt & Pastor Moreno, 2004) define three types of assembly models:

- ***Simple-model:*** A single type of product is assembled, and workstations repeatedly execute the same tasks.
- ***Mixed-model:*** Variants of a basic product are assembled, and production does not involve setup times between one variant and another because the same basic operations are required to produce all variants. Therefore, units of different models are produced in an arbitrary mixed sequence.
- ***Multi-model**:* Different types of products can be assembled in the same system; however, in this case, the joint processes between one type of product (or model) and others vary significantly, which is why batches are produced, and setup times between batches are considered.

2.1. Robotic assembly line

Incorporating robots and other automated machinery is a common practice to promote the concept of smart factories. A robotic assembly line (RAL) utilizes robots to perform assembly-related tasks. This enables production resources to be quickly reorganized, efficiently producing a specific range of products and thereby improving system flexibility (Rubinovitz, 1991). In the early 1960s, robots were employed in the industrial sector. Leading industries such as metalworking, electrical/electronics, and automotive mostly depend on industrial robots in their assembly lines to minimize labor costs and labor-related processing time changes. Robots have made it possible to assemble almost anything, no matter how big or small, thanks to technological advances. High levels of automation on assembly lines can also improve efficiency, productivity, flexibility, reliability, and cost savings. An RAL is used in the following situations.

- It is required to assemble several products with moderate to high production volumes using the same equipment.
- Assembly tasks need to be performed rapidly within a short timeframe.
- Parts may not be able to be manipulated manually, or manual handling may easily cause damage.
- Manual assembly is highly complex and requires a high degree of precision and repeatability. In addition, important considerations in assembly work include ergonomics, safety, and health risks (Chutima, 2022).

A RAL is a system that resembles a flow shop and is composed of several specially designed robotic assembly workstations used to assemble high-volume, low-variety products with stable designs and demand (Rubinovitz et al., 1993). An RAL is similar to a conventional production line. Fig. 2 shows the operation structure of an RAL, where the robots are in a chain or serial line, and the product advances along the line. Each robot oversees performing standardized assembly tasks, with the objective of balancing each robot's workload to unify the operation cycle time.

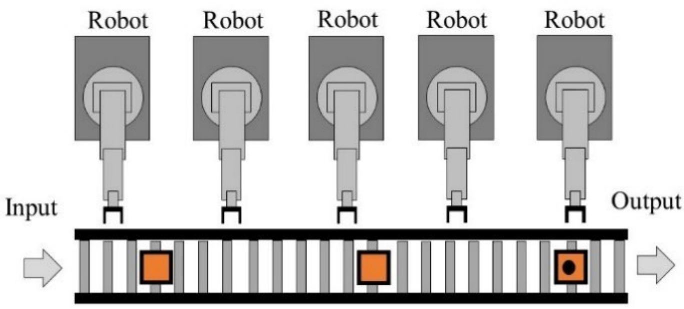

Fig. 2. Robotic assembly line RAL

2.2. Robotic assembly cell

A robotic assembly cell (RAC) is an assembly workstation with one or more industrial robots, part feeders, tools, and assembly equipment, all centered around an assembly table (R. Marian et al., 2003; Mohamed et al., 2001). RAC can assemble a wide variety of products in small-to-medium batches. Designing an RAC with multiple robots has key advantages for manufacturing companies.

- It combines the flexibility of a process-based design with the productivity of a product-based design.
- The assembly process requires different characteristics such as varying ending effects, workload, repeatability, degrees of freedom, and precision.
- The ability of robots to simplify complex part orientations during assembly.

One of the key themes of RAC is the flexibility that robots provide. In a RAC with multiple robots, variants of the same product or different products can be assembled. Additionally, an assembly task can be performed by different robots. However, selecting which robot will perform an assembly task introduces a challenge: how to allocate assembly tasks among robots to minimize the total assembly time. For the following reasons, RAC is considered to be more dexterous and flexible than RAL.

- In a RAL the assembly sequence is fixed, while in a RAC the sequence of the assembly is not restricted.
- RAC is easier to reconfigure and requires less space compared to RAL.
- In a RAC a wide variety of products can be assembled using the same resources, making it more adaptable than a RAL.
- In a RAL the robots are programmed to perform specific assembly tasks, whereas in a RAC the robots can perform different tasks depending on the assembled product.

RACs designed with more than one robot can offer much more flexibility and efficiency when a wide variety of products need to be assembled and reduce setup and processing (Abd et al., 2011b). The use of multiple robots in a RAC allows for greater system flexibility, increases productivity, reduces production costs, and enables a quick and efficient response to unpredictable market changes. However, to achieve these benefits, an intelligent system is necessary to plan assembly tasks in sequences that optimize robot usage, minimize idle time, and maximize efficiency, given their high initial investment (Abd, 2015). Fig 3 shows the configuration of an RAC with multiple robots.

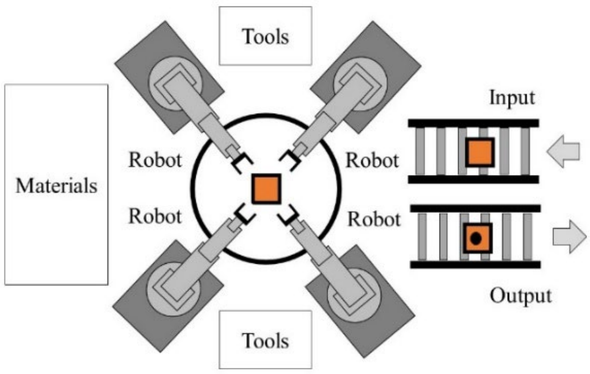

Fig. 3. Robotic assembly cell RAC

Full robotization of assembly cells can increase productivity; however, it is costly. Collaborative robots, or cobots, offer a new alternative for performing assembly tasks within the same cell where humans and robots work together. Cobots are equipped with high-performance sensors and are controlled by intelligent systems that enable effective interaction with their environment and with humans. Since cobots do not require safety fences, they can assist humans in performing highly efficient, flexible, and ergonomic assembly tasks. An assembly cell that incorporates human-robot collaboration is known as RAC-HRC. There are different types of RAC-HRCs depending on the type of collaboration required between humans and robots. Four categories of collaboration can be defined, according to ISO-TS 15066 (Stadnicka & Antonelli, 2019):

- **Independent operation:** worker and cobot operate independently on different workpieces.
- **Synchronized cooperation (collaboration):** worker and cobot operate consecutively on one workpiece.
- **Simultaneous cooperation (collaboration):** worker and cobot operate on the same workpiece, without any physical contact.
- **Assisted cooperation (collaboration):** worker and cobot operate on the same workpiece at the same time, and the process is done by both the cobot and the worker together.

Fig. 4 shows two RAC-HRCs within an assembly line. In the literature investigated, some works address the planning and scheduling of RACs that are part of an assembly line. However, the research focuses only on the assembly cell rather than the assembly line as a whole.

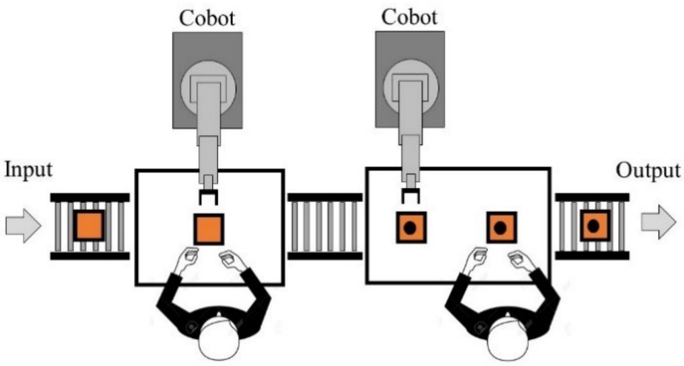

Fig. 4. Robotic assembly cells with human-robot collaboration

2.1. Robotic matrix-structure assembly system

Traditional assembly systems face difficulties inherent to their configuration. For example, assembly lines are efficient when high volumes of the same product need to be assembled but often struggle with mixed-model production scenarios because processing times for assembly tasks can vary. These variations can cause workstations to become blocked or idle, leading to unbalanced resource utilization, reduced productivity, and increased production costs. Addressing this balance issue becomes even more challenging as additional product variants are introduced onto the assembly line (Boysen et al., 2009). Assembly cells are efficient for assembling a wide variety of products in small batches. The challenge lies in creating a system capable of producing large volumes of diverse products while adapting to new market conditions. This requires the system to offer the necessary flexibility and scalability to deliver specialized products with short processing times and low costs (Mayer et al., 2019). In recent years, a new configuration has emerged that combines the benefits of sequential flow lines (efficient for high volumes) with the advantages of assembly cell designs (efficient for a wide variety of products). This new design is called a matrix-structure assembly system (MSAS). The MSAS can assemble a wide range of different products, incorporate new products into the existing line, scale to different production volumes, and accommodate design and process reconfigurations. This flexibility is achieved by allowing adaptive routing of products through the system (Trierweiler et al., 2020). Other advantages of MSAS include scalability and reconfigurability. Scalability can be achieved by duplicating bottleneck resources at the workstation level. Reconfigurability is achieved through the modular design of assembly workstations and the associated resources (Göppert et al., 2021). The current state of research, as of the date of this study, does not find any studies addressing MSAS composed entirely of RACs as assembly workstations. However, recent developments indicate that some specialized robotics and automotive companies are exploring the concept of creating MSAS using only RACs. In this paper will address the description of this type of production, which we refer to as the robotic matrix-structure assembly system (RMSAS).

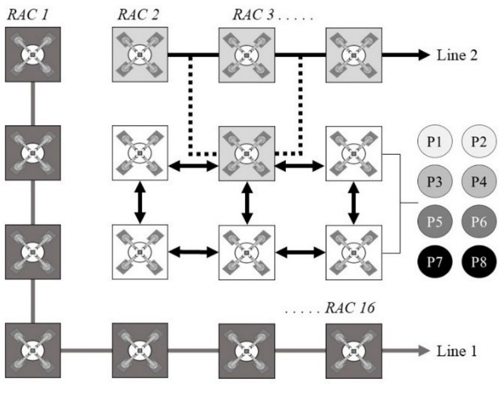

Fig. 5. Robotic matrix-structure assembly system

Fig 5 depicts an RMSAS featuring a 4 × 4 RACs structure. Each RACs can function as an independent assembly workstation or can be integrated into an assembly line. The matrix configuration enables the creation of assembly lines for high-volume production, such as Lines 1 and 2. Simultaneously, it facilitates the use of independent assembly workstations for low-volume production, resembling a job-shop system. Furthermore, this matrix configuration offers the flexibility to introduce a few assembly workstations or RACs into a line to prevent obstructions caused by failures, reconfigurations, and bottlenecks. In an RMSAS, assembly workstations can perform different tasks called work package (WP) for different types of products. The WP can be processed at different assembly workstations, helping to avoid queues and waiting times. The WP is transported autonomously through the RMSAS. The product route is not always the same, as the selection of the next assembly workstation can be based on predefined optimization objectives, such as the shortest distance, the shortest delivery time, the shortest time/distance ratio, or the degree of system utilization. An RMSAS works under the following principles: individual cycle time, redundant assembly workstations, multiple WPs per assembly workstation, flexible product routing, and equipment utilization rate (Greschke et al., 2014). Assembly workstations in an RMSAS are divided into smaller individual subsystems where the cycle times of each subsystem do not need to match the average cycle time of the entire system. This is because the workstations function as partially autonomous systems with varying cycle times. However, to achieve high levels of productivity, it is crucial that the total system utilization be as high as possible. The overall system utilization is derived from the utilization of all the individual subsystems. To achieve maximum utilization, it is essential to minimize waiting times at each individual task step (Schukat et al., 2022). The most important planning decision relates to the selection of workstations for each task of each product. This situation occurs every time a task of a product is completed on an assembly workstation. The product should then select the next assembly workstation for the next task. This decision can be based on different criteria, such as the shortest distance between assembly workstations or the shortest processing time. To select an appropriate dispatch rule, the performance of several of them is evaluated (Mueller & Schmitt, 2020).

The main elements of an RMSAS are the WP and the buffers. Buffers can have limited capacity to store products before processing, which helps to reduce idle time at assembly workstations. The WP is a set of tasks to be performed on each type of product. The assembly workstations are designed to perform some of the WP of each product, and the same WP has several alternatives to be processed in different assembly workstations. The *Pi* product requires a finite sequence of WPs processing steps. $Pi = [WP_{i1}, WP_{i2}, \ldots, WP_{imi}]$ where *mi* is the total number of WP for the *Pi* product. Since multiple product types can be produced on an RMSAS, assembly workstations must be able to process all WPs of all product types. Consequently, at least one assembly workstation must be able to process a required WP. One assembly workstation can create a WP suite for different products. The assignment of WP to the assembly workstations is based on the available assembly equipment of the workstations. WPs assigned to a particular assembly workstation are a subset of all existing WPs of all product types (Schönemann et al., 2015). Fig 6 shows the design of an RMSAS where the WPs of products P1 and P2 can be assembled in more than one assembly workstation or RAC with flexible product routing. To eliminate bottlenecks, assembly workstations can be duplicated and operated in parallel; it is determined which assembly workstations need to be duplicated as highly efficient production segments can remain in the line layout (Göppert et al., 2021). In conclusion, the configuration of the RAS depends on the production volume and variety of products to be assembled.

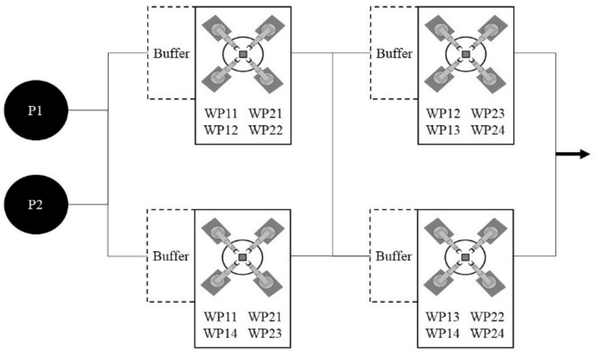

Fig. 6. Work packages flow into the RMSAS

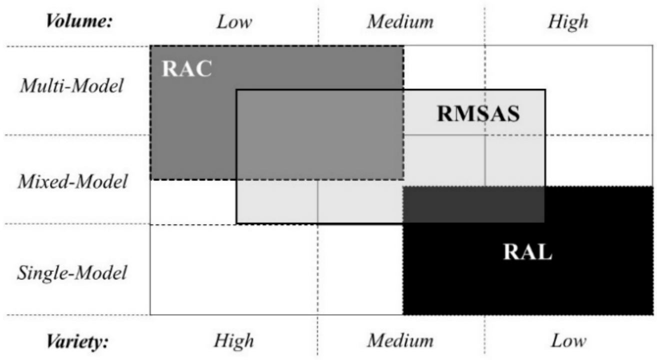

Fig. 7. Matrix volume-variety and RAS

Fig 7 shows the different RAS and their implementation according to the conditions of volume and variety of production. The RAC is highly efficient when it is required to assemble a wide variety of products with low production volume, known as multi-model systems. On the other hand, RAL is efficient when it is required to assemble a high production volume and medium to low variety, which makes it suitable for simple-model or mixed-model assembly. For its part, the RMSAS, due to its matrix architecture, can be efficient for both mixed-model and multi-model systems with medium-high volume and variety, and Table 2 shows the main differences between the RAL, RAC, and RMSAS systems.

Table 2
Differences between RAL, RAC, and RMSAS

Features	RAL	RAC	RMSAS
Volume	High	Low	Medium
Variety	Low	Wide	Wide
Process	Divided	Integrated	Redundant
Task in workstation	Simplified Specified Standardized	Complex Versatile Not standardized	Complex Versatile Not standardized
Cycle time	Short (3-60) seconds	Long (3-20) minutes	Medium (3-10) minutes
Parts flow	Line One route	Circular Network	Variable Various routes
Material handling system	Short (roller conveyor)	Rigid (roller conveyor)	Versatile (AGV)
Workstations per process	Single	Single	Redundant
Number of robots	1-100	1-4	1-100
Number of assembly parts	1-6	1-50	1-50
Tool changes	Single tool Multiple clamp	Switchable tool Automatic	Switchable tool Automatic
Objective function	Balanced flow	Minimize cycle time	Maximize utilization

3. Taxonomy of assembly planning and scheduling problems

One of the most crucial components of operational planning in the manufacturing industry is assembly planning. An assembly plan defines how to make a complete product from separate parts, considering various factors such as the geometry of the parts and the final product, the precedence of assembly tasks, the availability of resources and tools, processing time, etc. Assembly planning encompasses three main sub-problems: assembly path planning (APP), assembly sequence planning (ASP), and assembly line balancing (ALB) (Ghandi & Masehian, 2015). Another problem associated with assembly planning is assembly task scheduling (ATS), where there is a set of different products made up of several tasks/jobs that must be ordered in a sequence that respects precedence and assigns the tasks to the available workstations or operator, to obtain an acceptable performance measure.

3.1. Assembly path planning

APP generates paths $\tau_1, \ldots, \tau_n$ from an initial position to a final position for all the parts $P_1, \ldots, P_n$ that make up a final product A, without collisions between assembled parts or with obstacles $O_1,\ldots,O_n$ in the workspace $W \in \Re 2$ or $\Re 3$. The APP plans robot trajectories to move a part within a 3D space with six degrees of freedom among stationary obstacles. It is an *NP-hard* problem. Therefore, the APP is just as complex as general movement planning problems. The APP allows, first of all, to provide better feedback to the designers of the products and their parts. Secondly, good planning of assembly routes guarantees the profitability of the process. Lastly, a poor assembly route plan can significantly increase manufacturing costs and reduce productivity, as long assembly routes consume more time and energy (Ghandi & Masehian, 2015). According to (Lotter, 2013), assembly tasks are defined as the sum of all processes used to join geometrically defined bodies. These processes are classified into handling operations (store, move, supply, classify, position), control (inspect, measure), union (assemble, fill, join, adhere), adjust (squeeze, hold, separate), and specials (seal, roughing, cleaning, unpacking, packing). The APP focuses on finding a set of sequences for the movements of some or all parts of the product from an initial position to its final configuration. The notion of space configuration proposed by (Lozano-Pérez et al., 1983), can be used to formulate the APP as a motion planning problem. In the case of a system M involving n movable objects mi (for example, the parts of assembly). The composite configuration space C is the set of all configurations, that is to say, $C = \prod_{i=1}^{n} c_{mi}; i = 1,\ldots,n$. A q is a minimal set of parameters that define the location of a moving part in space. Given the initial configuration q_{dis}, the problem is to find a feasible path in C from q_{dis} to a final assembled configuration q_{ass}. The objective function for the APP problem is related to the path of the assembly, such as minimizing the length of the path, number of tools required, number of part orientations changing, inserts between parts, or maximizing path smoothness or safety of the path (Ghandi & Masehian, 2015).

To model the assembly problem, in addition to the parts, characteristics such as the dimension, geometry, and constraints of the parts and other components required to perform the assembly tasks must be considered. These characteristics are defined below:

Dimensions. A Euclidean workspace can be used to simulate an assembly process in two or three dimensions, represented by $\Re 2$ o $\Re 3$ (Halperin et al., 2000).
Components. Refers to the parts that make up the product, the assembler (human or robot), and the assembly tools (Rakshit & Akella, 2014).
Geometry. The geometry of the parts can be rigid or deformable. Deformable parts can be further classified into two types: articulated or flexible (Zhang et al., 2008).

Movements. Solving the APP can streamline the paths and trajectories of the robots' end effectors, resulting in significant resource savings and a faster, more efficient assembly process. The movements and trajectories can be translational, rotational, or helical (Chang & Li, 1995).

Constraints and objective function. There are geometric, physical, and mechanical constraints to consider. The physical properties of parts include friction, gravity, and forces, while mechanical constraints involve deformation under tension, torsion, or compression. Objective functions often prioritize time-based metrics, such as minimizing cycle time or total assembly time. Other objectives may include maximizing equipment efficiency or utilization, smoothing the path, and minimizing costs, the number of moves, or the complexity of established paths (Cortés et al., 2008).

Scale. Scale problems can be gross or fine. In a gross problem, the clearance between the parts is much larger than the size of the parts, which generally simplifies the assembly process. In a fine problem, the spaces between parts are so tight that even small positional errors can prevent a collision-free assembly process (Carlson et al., 2013).

Sequenceability. Refers to the relationship between the movements of subassemblies and grippers (or "hands"). The simplest problems involve two-handed plans, also known as binary or sequential plans. In this context, the work table where the assembly is placed is considered one of the "hands." (Wilson et al., 1995).

Monotonicity. Refers to the need for intermediate assembly operations that involve at least part of the assembly. Non-monotonic assembly path plans do not require the identification of these intermediate positions (Ghandi & Masehian, 2015).

Linearity. Refers to the restriction that only one part can be assembled into the rest of the assembly at a time. It does not allow for the simultaneous assembly of more than one part in at least one stage of the assembly operation (Morato et al., 2013).

Coherence. Refers to the restriction of creating a subassembly of parts before inserting them into the final assembly. To assemble previously joined parts, it is necessary to secure or maintain the stability of the subassembly before completing the final assembly.
In Fig. 8(a) there are no precedence constraints for the assembly, which makes the space of feasible sequences very large, while in Fig. 8(b), an assembly with linear sequences is presented where precedence constraints make the space of feasible sequences smaller. In Fig. 8(c), a coherence sequence is shown in which subassembly must be performed before final assembly.

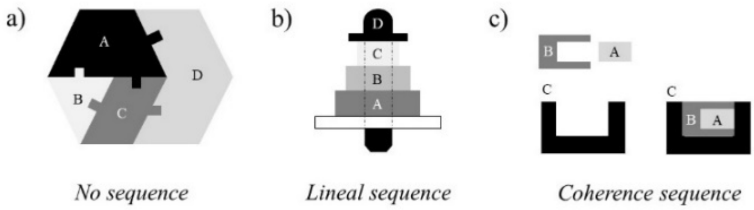

Fig. 8. Assembly sequence types

3.2. Assembly sequence planning

An assembly A is a product composed of n individual parts $A = \{C_1, C_2, \dots, C_n,\}$ that are assembled with a finite number of m assembly tasks $T = \{T_1, T_2,, \dots, T_m,\}$, where $m \geq n$. Each component C has given relative locations such that they do not overlap and where each part n forms a subset of the whole. The ASP has the following information:

- **I1 contact/connection**: on T defines a set of join relations C, if $(Ti, Tj) \in C$ means that Ti and Tj have a connection; that is why $(Ti, Tj) \in C \Leftrightarrow (Tj, Ti) \in C$.
- **I2 precedence:** on T defines a set of precedence relationships P. A binary relation of two operations $(Ti < Tj) \in P$ means that Ti must be done before Tj. If $(Ti, Tj) \in P$, then $(Tj, Ti) \notin P$.
- **I3 optimization criteria:** on A defines a set of optimization criteria TMk, $k \in N$.
- **I4 optimization function:** The objective function aims to either maximize or minimize F.

Given a set S of assembly sequences for the manufacture of a product, $S = \{S_1, \dots, S_i, \dots, S_n / S_i \in T\}$. A feasible assembly sequence S only satisfies conditions *I1* and *I2*. If an assembly sequence S satisfies conditions *I1*, *I2*, and *I3*, it means finding the optimal assembly sequence (Wolter et al., 1992). The assembly sequence determines the order of performing the assembly tasks that join the nnn pieces of a product, following a geometric description of their positions in the final assembly. The ASP consists of three steps (Jiménez, 2013).

Precedence constraints. Precedence constraints specify conditions that determine the execution or non-execution of some sequences established between tasks. The constraint determines that a set task can only be performed if its previous task has already been performed. Human experience is a crucial factor in defining precedence constraints and reducing the complexity of the process. (Cottrez & Van Brussel, 1989). Precedence constraints may affect the use of gripping tools and accessories. This is known as resource constraints. In the study of (Delchambre, 2012), two types of precedence constraints are defined: hard constraints and technological constraints. Hard constraints depend on the geometry of the parts and their position in the final assembly. Technological constraints are determined by an operator and define an assembly sequence to facilitate the process. Assembly sequencing begins with the graphical representation of assembly tasks. These representations can be made using graphs, vector elements, labels, or similar methods. The graphical representation of the n parts, subassemblies, or tasks is defined as unique elements and represented by nodes or labels. The connections established between nodes determine the precedence constraints of the assembly tasks. Among the most common representations is the directed graph, where the nodes are stable partitions of the assembly of parts, and the arcs correspond to the feasible assembly tasks (Alfadhlani et al., 2019). The AND/OR graph is one of the most commonly used graphs to represent possible assembly sequences. In this graph, the root node represents the complete assembly, while the other nodes represent the parts or subassemblies. The arcs indicate feasible assembly tasks viewed from the bottom up. Each node is linked to several alternative AND part combinations, and there are several different OR combinations to perform a subassembly. Another widely used graphical representation is the bill of materials (BOM). The BOM lists all parts, subassemblies, and the final product, along with the required quantities. The BOM can be represented by a structured list or by a tree chart with hierarchical level codes (Nof & Drezner, 1993). Fig 9 shows some of the most used graphs to represent ensemble sequences. Graph (a) is a network of operations where each node specifies an assembly task and the arcs determine the precedence, graph (b) is a BOM graph where the nodes identify the parts, subassemblies or materials and the arcs indicate between the possibility of assembling them. For example, node A represents the final product after assembling subassemblies B and C, while to assemble B, parts d, e, and f are required, and (c) is an AND/OR graph which illustrates all possible assembly sequences between parts A, B, and C.

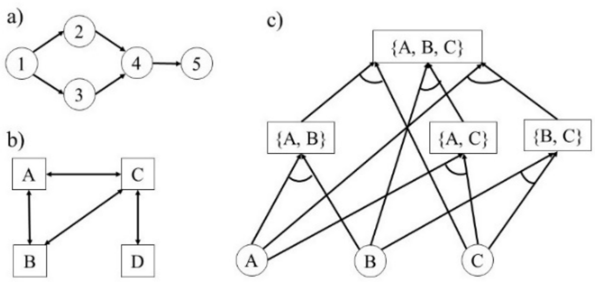

Fig. 9. Graphs for the representation of precedence relationships

Feasible sequence. The amount of assembly combination of the n parts is between $n-1$ and $n(n-1)/2$; for assembly without sequence, the number of combinations of possible assemblies can be of the order of $n!$ (De Fazio & Whitney, 1987). The size of the solution space for all possible sequences indicates the complexity of finding an optimal sequence through an exhaustive search. If the number of assembly tasks is equal to the number of parts, the number of sequences is given by the permutations of parts ($n!$). Introducing precedence constraints makes the sequences linear and monotonic. Without the linearity constraint, the solution space expands to $(2n - 2)!/ (n-1)!$.Allowing non-monotonic sequences makes the number of possible sequences infinite (R. M. Marian et al., 2003). The complexity of the ensemble sequence is generally measured in terms of:

- Number of parts that make up the assembly.
- Number of directions in which the parts must move.
- Number of reorientations or changes of direction.
- Number of tools and tool changes.
- Number of nonlinear steps.
- Depth of an assembly sequence.

Other criteria can be used to define an assembly sequence. For example, in the case of large parts, they can be grouped to favor sequences where large parts undergo fewer movements, reducing processing time and energy consumption. Favoring tasks where more complex movements involve easily manipulated parts is known as manipulability criteria. Another example is assembly uniformity, where similar parts are grouped to minimize the number of tool changes required.

Sequence selection. To find a feasible assembly sequence or select the best assembly sequence within the entire space of possible sequences, it is necessary to apply search or optimization algorithms to satisfy one or more performance criteria. Graphing exhaustive search methods is the simplest strategy that guarantees completeness, but it is impractical except for very simple assemblies. In the study of (Ghandi & Masehian, 2015), some of the most used criteria to select sequences for the ASP are mentioned (the information is presented in Table 3). The choice of the optimal sequence assembly for the ASP has an *NP-hard* complexity (R. M. Marian et al., 2003).

Table 3
Objective functions for ASP

Minimize	Maximize
Total assembly time	Efficiency
Cycle time	Utilization
Assembly cost	Unform workload
Tool changes	Geometrical constraints
Changing assembly directions	Assembly stability
Assembly complexity	
Distance traveled by the tool	
Change of the type of assembly	
Similarity of connection	

3.3. Assembly line balancing

ALB problem consists of the identification, selection, and sequencing of assembly tasks in a system with assembly workstations in a chain. Assembly systems are composed of a finite set of assembly workstations and tasks, which are assigned a processing time and a set of precedence relationships that, based on product design, specify the sequence of tasks assembly. Basically, the ALB consists of assigning the tasks to the ordered sequence of the assembly workstations in order to achieve a balanced load of tasks that allows unifying the cycle time of each assembly workstation that allows a high level of efficiency and productivity, all this satisfying the precedence relationships (Capacho Betancourt & Pastor Moreno, 2004).

An assembly line consists of several assembly workstations $j=1...m$ arranged along a conveyor belt or similar mechanical material handling equipment, where assembly parts are joined by operations called tasks $i=1...n$, which have a processing time ti, then the assembled parts are moved from one assembly workstation to another to finish the task. At each assembly workstation, certain operations are performed with respect to a cycle time C. The decision problem is to optimally assign the tasks of assembly i to assembly workstations j with respect to some objective. This is known as the assembly line balancing problem (ALBP). ALBP has been extensively studied. The classical classification of the ALBP distinguishes two main types: simple assembly line balancing problems (SALBP) and general assembly line balancing problems (GALBP). Given their combinatorial nature, these problems are very difficult to solve optimally, especially GALBP problems. In the case of industrial problems, their resolution is complicated due to the large number of tasks that make up the production process and a large number of constraints present in real problems (Scholl & Becker, 2006). The terms used in the ALBP are.

- **Task:** It is an indivisible work unit i that is associated with a processing time ti. The total work required to assemble a product is divided into a set of n tasks.
- **Precedence relationships:** They are defined by the constraints on the order in which tasks can be executed in the assembly process. Thus, a task cannot be processed until all immediately preceding tasks have been processed. Graphs are used to represent precedence relationships.
- **Workstation:** It is part j of the assembly system where tasks i are executed. They may consist of an operator (human or robot), certain types of machinery, and specialized equipment or process mechanisms.
- **Cycle time**: The cycle time C is the time available at each workstation j to complete assignment tasks i for a unit of product. It can be the maximum time or the average time available for each work cycle.
- **Workload:** It is the set of tasks i that are assigned to workstation j called X_{ij}.
- **Workstation time:** It is the sum of the times of all the tasks i assigned to a workstation j, which is called Sj.
- **Idle time:** It is the difference between the cycle time C and the workstation time Sj.

RAL planning problems are addressed under the robotic assembly line balancing problem (RABLP) methodology. The RABLP is a problem of efficient assignment of tasks and assignment of robots to assembly workstations. RALBP appeared for the first time in 1991, when Rubinovitz and Bukchin presented a simple ALBP incorporating new constraints for a robotic assembly process. In the context of RAL, the definition of RALBP, in addition to solving the problem of assigning a task to a workstation, must also assign a robot to a workstation. The robots can be identical or different. Processing times for assembly tasks performed by robots are not fixed; the time required for any task can vary depending on the robot's performance. Optimizing the RALBP involves solving two subproblems: 1) assigning tasks to workstations to optimally balance the assembly line, and 2) assigning the most efficient robot to execute the tasks at each workstation (Chutima, 2022).

In an RAL, the variation in the processing time of an assembly task is minimal and nearly consistent with its expected values. Therefore, in the RALBP, it can be assumed that task processing times are deterministic. However, task processing times may vary depending on the types of robots, tools, and assembly equipment chosen to perform the tasks at each workstation (Nof & Drezner, 1993). The following are the assumptions that are typically used to model the RALBP:

- The RALBP addressed the balance problem for a unique model of a single product.
- An assembly task cannot be subdivided between two or more workstations.
- The precedence relationships between tasks are known and fixed.

- Task processing times are deterministic, and values depend on the robots chosen to execute the tasks.
- A workstation can be used to perform any task if the robot assigned to the workstation can complete it.
- Each workstation can accommodate only one robot, resulting in the same number of workstations and robots in the RAL.
- All types of robots are always ready for use without capacity limitations or breakdowns.
- The robot movement, tool change, and setup times are independent of the sequence and negligible or are already included as part of the task time.
- The loading, unloading, and transport times of the workpieces are negligible.
- The costs of the robots are not considered.

As a result, the RAL is generally used to generate a quick return on investment. To achieve this, line balancing becomes an effective tool to eliminate bottlenecks on the assembly line, reduce waste, and develop a continuous flow of parts and subassemblies. The following questions must be answered when making RALBP decisions:

- How should tasks be assigned to assembly workstations?
- What types and how many robots should be installed?
- How should robots be allocated to assembly workstations?

There are two basic types of RALBP. In Type I or RALBP-I, with a given cycle time, the number of workstations must be minimized by assigning tasks to the workstations and selecting the most appropriate robot to perform the assembly tasks. In Type II or RALBP-II, there are a given number of fixed workstations, and the objective is to minimize cycle time by assigning tasks and allocating the appropriate number of available robots to each workstation. (Yoosefelahi et al., 2012). Other types of RALBP include RALBP-F, which generates a viable solution for defined workstations and a given cycle time; RALBP-E, which seeks to maximize line efficiency by minimizing cycle time and the number of workstations simultaneously; and RALBP-C, which aims to minimize the cost of RAL design. Problems that do not fall into the aforementioned categories will be classified as RALBP-O or others. To formulate the fundamental mathematical model of the RALBP-I and RALBP-II, the nomenclatures described in Table 4 are defined (Chutima, 2022).

The RALBP can be further subdivided at a lower level of the hierarchy with the concept of 4M. (man, machine, material, and method) is adopted (Chutima, 2022). Fig. 10 shows the 4M constraints for analyzing different RALBP settings.

Table 4
RALBP nomenclature

i,j	Variable representing the assembly tasks (*i,j = 1, …, Nt*)
r	Variable representing the robots (*r=1,…Nr*)
w	Variable representing the workstation (*w=1,…W*)
W	Maximum number of workstations possible (*W≤Nt*)
Nw	Given number of workstation (Tipe II), or minimum number of workstation (Tipe I)
Nt	Given number of tasks
Nr	Given number of robots
Sw	Group of tasks allocated to workstation *w*
t(Sw)	Total task time of workstation *w* (or workstation time)
C	Cycle time
Ct	Given cycle time
Pre(j)	Task j's direct predecessor
tir	Time required by robot *r* to execute task *i*
δ	A very large positive number
Xiw	1, if task *i* is allocated to workstation *w;* 0 otherwise
Xirw	1, if task *i* is executed by robot *r* allocated to workstation *w*, o otherwise
Yrw	1, if robot *r* is allocated to workstation *w;* 9 otherwise

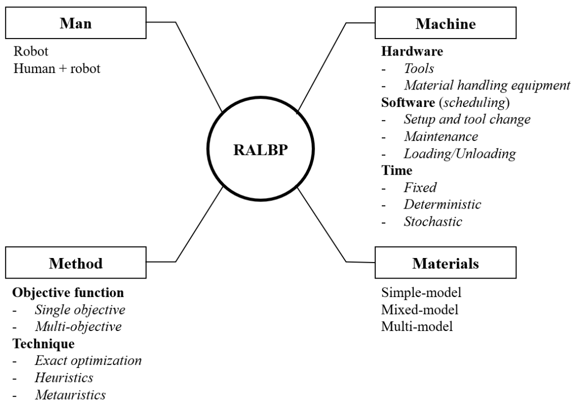

Fig. 10. RALBP subdivision of problems associated with the 4M

Another way of classifying the RAL is according to the configuration of its architecture. The fundamental assembly line layouts are comprised of a robotics parallel workstations assembly line (RPWAL), robotics parallel assembly lines (RPAL), robotics U-shaped assembly line (RUAL), robotics two-sided assembly line (R2SAL), and robotics straight-shaped assembly line (RStAL). In recent years, with the installation of collaborative robots in assembly lines, a new planning problem called the assembly line balancing problem with human-robot collaboration (ALBP-HRC) was born. In this problem, apart from solving ALB, it is necessary to address other challenges, such as assigning sets of tasks to a robot or a worker and determining if these tasks should be performed simultaneously in parallel or collaboratively (Weckenborg et al., 2020).

3.4. Assembly task scheduling

ATS is a major short-term problem decision in assembly planning. In an assembly process, a product is considered as a complete unit that assembles all the components of the final product. The ATS deals with finding, in relation to a set of resources, an adequate sequence for the execution of a set of established tasks and the times in which their different tasks must be started in order to reach an optimal or acceptable value of one or more performance measure (Brucker, 2006). Scheduling optimizes the allocation of limited resources for the execution of assembly tasks. These resources encompass workstations, robots, tools, material handling equipment, and the parts to be assembled. Each task involves a series of assembly operations to be conducted at assembly workstations. In general, a planner looks for a schedule that allows him to minimize assembly time, meet delivery deadlines, reduce work-in-process inventories, and maximize the use of assembly workstations and equipment (Pinedo, 2012); (French, 1982); and (Blazewicz, 2013). ATS solves the problems of task sequencing in the RAL, RAC, and RMSAS. In the RAL, when it becomes necessary to assemble variants of a basic product, a mixed-model environment is generated. In this setting, three main problems must be addressed: task assignment, model sequencing, and robot allocation (Li et al., 2018). The recent trend has led manufacturers to switch to RAL configuration for low-volume assembly of custom products; this is called mass customization. The strategic change became effective due to the diversified needs of customers along with the individualization of products. Ultimately, this triggered the investigation into the RAL balancing and sequencing problem for custom products on the same line in a mixed scenario, which is characterized as a robotics mixed-model assembly line balancing and sequencing problem (RMALBSP) (Uddin & Lastra, 2011).

When the RAC is made up of a set of resources working in parallel, such as two or more robots or HRC, and the assembly tasks require a single operation from one of the resources, it is necessary to solve the scheduling problem. This problem is known as multiprocessor task scheduling (MTS) and is a generalization of parallel machine scheduling where a single job can be executed simultaneously by a set of parallel resources (Chen & Lee, 1999). For example, in an RAC a product composed of *n* assembly tasks must be assembled, each task can be assembled by a single robot or by a set of *m* robots working simultaneously. Task 1 can be assembled by robot 1 and robot 2 together, by robot 1 and robot 3 together, or simply assembled by robot 1 alone. The processing time for each task depends on the group of robots assigned to assemble the task. A group is formed when a set of robots is working on a particular task, but a robot may not belong to a fixed group all the time. The objective is to assign these *n* tasks to *m* robots in order to minimize the total assembly time (Framinan et al., 2014). Different variants of MTS problems are described in the literature; either all resources are identical, or resources are not identical. The MTS is described over a set of resources *R* and a set of tasks *J*. A mode *m* is composed of a subset of resources *Rm* such that $Rm \subseteq R$. A task *j* must be executed in one of the possible modes *Mj*, each requiring a specific processing time *Pjm* (Ferreira et al., 2021).

The ATS in the RMSAS is like scheduling problems in FMS, where basic scheduling consists of sequencing tasks (products or WPs) at system input and in assembly workstation buffers. The assignment and scheduling of the tasks depend on whether they are carried out in a single assembly position or more than one. When tasks are performed at a single assembly workstation, transportation between assembly workstations is not considered. On the contrary, if the tasks must be carried out in more than one assembly workstation, it is necessary to program the task route that consists of programming the material handling equipment, the most used being the automatic guided vehicle (AGV). The scheduling procedures for assembly tasks in assembly workstations and the scheduling of vehicle transport operations are strictly interrelated. Dispatch algorithms for scheduling must consider the various interactions between assembly workstations and AGVs and use current information about the assembly process and system status. Some of this information is related to the product, such as processing and transportation times, precedence relationships between tasks, product assembly routes, etc. (Mayer et al., 2019). Scheduling problems within an RMSAS can be categorized as follows:

- **Job sequencing problem:** selecting the product from the queue.
- **Task selection problem:** selecting the next task of the product to assemble.
- **Workstation selection problem:** selecting the next assembly workstation to assemble a task.
- **AGV scheduling problem:** selecting the AGV to transport or the route of the AGV.

The number of variables within an ATS is very large, and it can have many more problems, such as the number of assembly workstations, the number of robots in the cell, the availability of tools, the number of AGVs, among others (Chan & Chan, 2004).

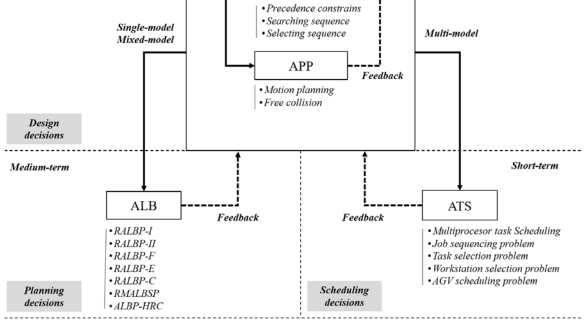

Fig. 11. Hierarchical structure of planning and scheduling problems in robotics assembly systems

The sequence of assembly tasks and the routing of each task are defined by resource availability and other circumstances such as the availability of robots and tools, the efficiency or utilization of assembly workstations, the distance to be traveled, or failures at workstations. The absence of an overall cycle time and a flexible flow system with independent assembly workstations eliminates the need to balance the system, allowing the production of highly individualized products within the same assembly system (Hofmann et al., 2018). Table 5 shows the description of the assembly planning and scheduling problems, and Fig 11 shows the hierarchical structure of planning problems in an RAS.

Table 5
Description of planning and scheduling problems for robotic assembly systems

Assembly problem	Sub-problem	Constraints		Objective function
APP	Motion planning	Dimension Components Geometry Movements Scale	Sequenceability Linearity Monotonicity Coherence Collision	**Minimizing:** path length, number of required tools, number of orientation changes of parts, interpart penetrations, etc. **Maximizing:** path smoothness, path safety
ASP	Precedence constraints Generation of feasible sequence Sequence selection	Assembly relations, fixed precedence, technological precedences, collisions, number of tools, number of parts, geometrical of final product, number of redirects, position parts, parallel work		**Minimizing:** tool change, assembly direction change, assembly complexity, assembly tool travel distance, assembly cost, energy consumption, etc. **Maximizing:** assembly stability, workload smoothness, satisfying geometric constrains.
ALB	RALBP-I RALBP-II RALBP-F RALBP-E RALBP-C RMLBSP ALBP-HRC	Robot related operation, available time, robot load capacity, failures, maintenance, investment cost, number of products, task time variations, task assignment control, setup time, number of objectives		**Minimizing:** cycle time, total assembly time, number of workstations, number of robots, assembly cost, idle time, number of robots, etc. **Maximizing:** line efficiency, workload smoothness, utilization
ATS	Multiprocessor task scheduling Job sequencing Task selection Workstation selection AGV scheduling	Number of workstations, number of robots, buffer capacity number of AGV, transportation, availability of workstations, setup time, assembly tools		**Minimizing:** Cycle time, makespan, transportation time, idle time, route length, wait time, queue time, number of AGV, assembly cost, etc. **Maximizing:** utilization, workstation efficiency

4. Solution approaches

Several solution approaches have been proposed for the assembly planning and scheduling problems due to its *NP-hard* complexity. Table 6 shows a classification of the different solution approaches, traditional and advanced solution approaches. Traditional approaches include analytical and heuristic approaches. Analytical approaches can find optimal solutions but are generally applicable only to small problems due to their high computational cost. Analytic approaches are inflexible, inefficient, and slow to solve the most robust planning problems. Heuristic approaches are capable of solving large problems; since they are mainly based on simple dispatch rules that generate good solutions but do not guarantee obtaining the optimal solution (Abd, 2015).

Table 6
Solution approaches for RAS assembly planning and scheduling

Traditional Approaches				Advanced Approaches			
Heuristic Approach		Analytical Approach		Simulation Approach		Artificial Intelligence	
Priority Rules	PR	Linear Programming	LP	Petri Network	PN	Genetic Algorithm	GA
Dispatching Rules	DR	Integer Programming	IP	CAD Simulation		Ant Colony	AC
Tabu Search	TS	Mixed Integer Linear Programming	MILP	Simulation Models	SM	Fuzzy Logic	FL
Local Search	LS	Dynamic Programming	DP	Discrete Event Simulation	DES	Bee Colony	BC
Simulating Annealing	SA	Branch and Bound	B&B	Agent-Based Simulation	ABD	Migratory Birds	MB

Recently, advanced methods such as simulation and artificial intelligence have been applied for two reasons: first, the solutions obtained are more promising, and second, advanced methods reduce the time required to find solutions compared to traditional methods (Abd, 2016). The planning of tasks in FAS aims to improve the efficiency of the system. Although flexibility is the key term that affects the performance of these systems, the documents analyzed so far do not consider it. Consequently, the problems considered in the current literature are too limited where many variables are ignored to simplify the models and the solutions provided are suboptimal. Recent studies have explored problems related to repair operations, controllable processing times, design issues, and bicriteria models within the context of FAS planning (Gultekin et al., 2018). To solve the planning problems in the FAS, the studies have used different types of performance measures to evaluate the results of the planning, the measures based on time, utilization, and cost, are the most used, the performance measures based on time, the one that has most attracted the attention of researchers (Allahverdi et al., 2018). The research studies based on the different approaches are presented below.

4.1 Traditional approach

Analytical and mathematical methods do allow finding the optimal solution to small problems. (Rubinovitz, 1991) were the first to formulate a linear programming model for RALBP, which balances tasks across assembly workstations and assigns the most efficient robot from the available set to each workstation. The objective is to minimize the number of workstations required on the line for a given cycle time. (Rubinovitz et al., 1993) propose a heuristic algorithm using the borders search using the branch and bound method (B&B) for a fixed line of a single product. In (Lin et al., 1995) implemented a heuristic algorithm to minimize the assembly cycle time, allowing for collision avoidance and simultaneous operations in an assembly cell with two robots. The heuristic was divided into three steps: initial insertion sequence, balancing and reallocation, and collision avoidance between robots. (Pelagagge et al., 1995) developed a heuristic that characterizes assembly tasks to solve coordination and collision avoidance problems. The assembly area was divided into two categories: outer and inner, with the inner area defined as the critical area. This approach finds acceptable solutions with a high level of robot utilization. (Del Valle & Camacho, 1996), proposed a constructive heuristic based on an analysis of the AND/OR graph for the assignment of tasks in a multi-robot assembly cell, considering tool change times to minimize the total assembly time. The study was applied to a RAC that must assemble different types of products. (Su & Fu, 1998) developed the dynamic picked-and-place model to analyze robot travel in the assembly process, the authors generally used fixed coordinates of the insertion points and the traveling agent algorithm to sequence the insertion points. However, the routing of the robot's travels must be based on relative coordinates because the coordinates of the insertion point and the charger are constantly changing, that is, the robot, the assembly table and the loader move simultaneously at different speeds. This study presents an simulated annealing based algorithm that can orchestrate the insertion sequence and allocate magazine slots for better performance than the heuristic approach. In the work of (Sawik, 1998), an integer linear programming formulation and a heuristic algorithm are presented to solve three problems within a flexible assembly system: machine loading, assembly routing, and assembly plan selection. (Bukchin & Tzur, 2000) propose an exact B&B algorithm to solve moderate problems in designing a flexible assembly line when multiple equipment alternatives are available. The objective is to minimize the total cost. For handling larger problems, a heuristic procedure is incorporated. (Khouja et al., 2000) describe a two-stage statistical clustering procedure for designing RAC. In the first stage, a fuzzy clustering algorithm is used to group similar tasks to balance the workload assigned to robots and achieve a defined cycle time. In the second stage, appropriate robots are selected for the task groups. (R. Marian et al., 2003) proposed a heuristic approach to solve the assembly task scheduling problem in a RAC to maximize cell performance. The heuristic has two modules: offline and online. In the offline module, an optimal or near-optimal assembly sequence is generated for each product. In the online module, a priority rule is determined for the assembly tasks of multiple products at each time to optimize the utilization of the available RAC resources. (Abd et al., 2011a) proposed an algorithm for RAC scheduling in a multi-product assembly environment. Various dispatch rules were applied, and four performance measures were suggested to evaluate RAC productivity. A case study is presented to illustrate the application of the proposed algorithm, demonstrating that the schedules achieved high efficiency. (Yoosefelahi et al., 2012), a mixed integer linear programming (MILP) model is provided for a RALB-II problem, where this model is considered to minimize three objective functions: the cycle time, the robot setup cost, and the robot cost. Due to the *NP-hard* complexity, an evolutionary metaheuristic algorithm is used. (Stadnicka & Antonelli, 2019) propose a combination between dispatch rules and Lean tools to carry out the design and scheduling of assembly cells where humans and collaborative robots interact to improve the efficiency of the process. The objective of the problem is to simultaneously determine a schedule of assembly tasks between workstations and select the assembly plan and assembly routes for a mix of products, balancing the workloads of the workstations and minimizing the total transportation time. In the study of (Weckenborg et al., 2020), a MILP formulation is presented to solve a RALBP with collaborative robots. The problem aims to minimize cycle time and balance the assignment of assembly tasks executed simultaneously, either in parallel or in collaboration by humans and robots. The model determines both the assignment of collaborative robots to workstations and the workload distribution between humans and robots. The following are some studies with a traditional approach applied to RAC planning. (Li et al., 2021) developed a multi-objective MILP model to minimize both the cycle time and the total purchase cost of collaborative robots for RALBP. They employed a multi-objective optimization algorithm to generate a set of high-quality Pareto solutions.

4.2 Simulation approach

A simulation approach involves imitating the actual operations of a process using software. However, few studies have been carried out with simulation approaches to address RAS planning and scheduling problems. These simulation methods are accompanied by heuristic and metaheuristic methods that allow decisions to be made as the variables of the problem change when various scenarios are simulated. Due to the complexity of these systems, it is not practical to find an optimal solution in an environment that changes quickly, but it is necessary to develop an integrated planning system capable of administering the dynamic and stochastic nature of production systems. (Caprihan et al., 2013). (Glibert et al., 1990) in this study, a multi-robot assembly cell was modeled using Robcad software, and various scenarios were simulated to reduce the total assembly time. Two methods were employed: the synchronous method, which enables online programming, and the asynchronous method, which generates offline programs and achieves better assembly times. (Hsu & Fu, 1995) developed a new methodology for modeling and programming a RAC. A multi-robot assembly cell was constructed using CimStation software, which enables collision detection between robots during the execution of scheduled assembly tasks. Two steps were undertaken to integrate planning with simulation. First, a graphical AND/OR approach was proposed to generate feasible assembly sequences; second, an optimal task sequence was determined by applying a search algorithm. (Basran et al., 1997) developed an agent-based framework for scheduling a multi-robot assembly cell. The study divided the assembly task into two stages: part picking and part assembly. Agents used the Contract Net protocol for the dynamic assignment of tasks to robots. A simulation approach was employed to validate the proposed framework, but collision avoidance between robots was not addressed in this work. (Lee & Lee, 2002) developed a strategy for scheduling and coordinating tasks in a multi-robot assembly cell using a supervisor-controlled logic system. They employed a Petri net representation to prevent collisions between robots in a shared area and to minimize the total task time. Various types of tasks performed by the robots were considered, including motion, tool changes, picking, and assembly. (Abd et al., 2012) used SIMPROCESS simulation software to examine the performance of various dispatching rules and a proposed fuzzy logic-based sequencing rule. The effectiveness of the proposed rule was evaluated using four performance measures: maximum tardiness, average tardiness, percentage of tardiness jobs, and percentage improvement. (Tsarouchi et al., 2017) propose a method for tasks planning of human-robot and design of the workplace. A model for the representation of humans and robots is proposed as an active resources team, while teams such as worktables and accessories are considered passive resources. human-robot workload is structured in a three-level model. A multiple-criteria decision framework for the formulation of alternative designs and task assignment is used. For the estimation of the criteria values, both analytical models and simulation are used which allows evaluating of the different alternatives. Petri nets is a powerful tool to model systems that can be characterized as concurrent, asynchronous, distributed, parallel, deterministic, or stochastic. When assembly operations share multiple resources, the Petri nets are combined with a variety of heuristics dispatching rules to solve the assignment resources and scheduling jobs, and the development of efficient programming algorithms for specific problems is less difficult; this type of approach is called simheuristic (Tuncel & Bayhan, 2007). Stochastic Petri nets are used to describe the uncertain events in the systems such as machine failures, repair time, processing time, release time of jobs, and machines. In addition, deterministic conditions such as buffer capacity, block machine, transition times, setup times, dead points, among others. This allows dynamic job scheduling. In dynamic scheduling problems, the integration of Petri nets with heuristic dispatching rules has been stimulated by the desire to obtain a good solution in a shorter time (Chincholkar & Chetty, 1996).

4.3 Artificial intelligence approaches

Artificial intelligence approaches, or expert systems, aim to transfer human expertise to computer systems. These systems analyze complex problems and offer viable, high-performance solutions. (Levitin et al., 2006) developed a genetic algorithm (GA) to solve large and complex RALBP problems. This algorithm provides a solution for grouping the N tasks to be performed at W workstations and assigning one of the R available robots to each workstation to achieve a minimum cycle time. The algorithm is based on a genetic approach, utilizing the principles of evolution. Unlike the B&B method, it does not experience a surge in storage requirements as the problem size increases. (Daoud et al., 2014) utilized ant colony optimization, particle swarm optimization, and GA to maximize the efficiency of a RALBP. These algorithms were combined with a guided search method to enhance quality and avoid local optima. An exact enumeration method was developed to evaluate the quality of the approaches, and discrete event simulation was employed to assess system performance. (Janardhanan et al., 2019) presented a RALBP study aimed at minimizing cycle time with sequence-dependent setup times. To address the problem's high complexity, they implemented a metaheuristic migratory birds algorithm, presented as a hybrid genetic algorithm. Through extensive computational experiments, the algorithm demonstrated promising results in both computational time and solution quality. (Maoudj & Bouzouia, 2019) developed a multi-agent dispatching system to schedule and control a RAC to minimize overall assembly time. They addressed the problem by using autonomous control agents. Three types of agents were employed: supervisory agents, local agents, and remote agents. These agents utilize common dispatch rules to negotiate and coordinate their individual decisions, achieve their local objectives, and provide an optimized global solution. Furthermore, due to the dynamic nature of assembly systems, they considered external factors such as unexpected robot failures or random arrivals of products and parts. (Çil et al., 2020) formulated a MILP model to solve small mixed-model assembly line balancing problems with collaborative robots, aiming to minimize the sum of the cycle times for each product model. For larger planning problems, they implemented and enhanced bee colony and artificial bee colony algorithms. (Çil et al., 2020) investigated the mixed-model assembly line balancing problem with collaborative robots, aiming to minimize the sum of the cycle times for various models. A MILP model is formulated to solve smaller problems optimally. For larger problems, the bee colony algorithm and the artificial bee colony algorithm were implemented. The proposed algorithms demonstrated superior competitive performance compared to nine other algorithms, including simulated annealing, GA, particle swarm optimization, the original bee colony algorithm, and three variations of the artificial bee colony algorithm. (Li et al., 2021) formulated a multi-objective MILP model to minimize both the cycle time and the total purchase cost of collaborative robots for a RAL. To address larger problems, they implemented a multi-objective migratory bird optimization algorithm, which demonstrated competitive performance compared to multi-objective GA, multi-objective simulated annealing, and two multi-objective artificial bee colony algorithms. Table 7 shows in chronological order the most relevant research publications referring to the problem of assembly planning in RAS.

Table 7
Chronological order of relevant publications in robotics assembly systems planning and scheduling

Reference	Robotic Assembly System	Assembly Planning		Assembly Model			Objective	Solution Technique
		Problem	Sub-problem	Single	Mix	Multi		
(Glibert et al., 1990)	RAC	ASP	Searching sequence	✓			Minimize total assembly time	On-line: free collision, and Off-line:B&B
(Rubinovitz et al., 1993)	RAL	ALB	RALBP-I	✓			Minimize number of workstations, and robots	B&B, and Frontier Search
(Chang & Li, 1995)	RAL	APP	Movements	✓			Maximize Maintainability	C-space C-obstacle
(Hsu & Fu, 1995)	RAC	ATS	Tasks sequencing	✓			Minimize total assembly time	And/Or graph, Priority Rules, and Genetic Algorithm
(Del Valle & Camacho, 1996)	RAC	ASP	Searching, and selecting	✓			Minimize makespan	And/Or graph, and Algorithm A*
(Su & Fu, 1998)	RAC	APP	Movements	✓			Minimize assembly travel distance	Simulated Annealing
(Sawik, 1998)	RAC	ATS	Jobs and routing		✓		Minimize maximum workload, and	MILP, and Heuristic Algorithm
(Kojima & Hashimoto, 1999)	RAC	ASP	Searching and selecting sequence	✓			Minimize number of robots	CAD Simulation
(Bukchin & Tzur, 2000)	RAL	ALB	RALBP-C	✓			Minimize total equipment costs	MILP, Heuristic Algorithm, and B&B
(Halperin et al., 2000)	RAC	APP	Dimension	✓			Minimize path	Motion Space Approach
(Thomas & Wahl, 2001)	RAC	ASP	Selecting sequence	✓			Minimize assembly cost	And/Or graph, and CAD Simulation
(R. M. Marian et al., 2003)	RAC	ASP	Searching and selecting	✓			Minimize assembly travel distance	Genetic Algorithm
(Rosell, 2004)	RAC	ASP	Searching sequence	✓			Minimize assembly travel distance	Petri Nets, and Priority Rules
(Levitin et al.,	RAL	ALB	RALBP-II	✓			Minimize cycle time	Genetic Algorithm
(Hui et al., 2009)	RAC	ASP	Searching and selecting sequence	✓			Minimize path	Genetic Algorithm, Ant Colony, and CAD Simulation
(Joseph & Sridharan, 2011)	RAC	ATS	Jobs and routing sequencing			✓	Mean flow time	Dispatching Rules, and Simulation Model
(Abd et al., 2011a)	RAC	ATS	Jobs sequencing			✓	Minimize makespan	Dispatching Rules
(Yoosefelahi et al., 2012)	RAL	ALB	RALBP-II	✓			Minimize cycle time, robot setup costs, and robot costs	MILP, and Genetic Algorithm
(Abd et al., 2012)	RAC	ATS	Jobs sequencing			✓	Minimize maximum of tardiness, and mean tardiness percentage of tardy products	Fuzzy Sequencing Rule, and CAD Simulation
(Izui et al., 2013)	RAC	ASP	Searching and selecting sequence	✓			Minimize assembly time, minimize area, and Maximize feasibility	Genetic Algorithm
(Carlson et al., 2013)	RAL	APP	Geometrical variation	✓			Maximize path smoothness	Path Planning Algorithm, and CAD Simulation
(Rakshit & Akella, 2014)	RAC	APP	Stability – force of gravity and friction	✓			Maximize stability	CAD Simulation, and Motion Stability
(Daoud et al., 2014)	RAL	ALB	RALBP-E	✓			Maximize the efficiency	Ant Colony, Particle Swarms Optimization, and Genetic Algorithm
(Schönemann et al., 2015)	MSAS	ATS	Task and routing sequencing			✓	Maximize utilization	Simulation Model
(Michniewicz & Reinhart, 2016)	RAC	ASP	Searching sequence	✓			Minimize assembly tool travel distance	CAD Simulation, and Cyber Physical Systems
(Abd et al., 2016)	RAC	ATS	Jobs sequencing			✓	Minimize makespan, total tardiness, and number of tardy jobs	Fuzzy Sequencing Rule, and Taguchi Optimization

Table 7
Chronological order of relevant publications in robotics assembly systems planning and scheduling (Continued)

Reference	Robotic Assembly System	Assembly Planning		Assembly Model			Objective	Solution Technique
		Problem	Sub-problem	Single	Mix	Multi		
(Michniewicz et al., 2016)	RAC	APP	Geometric constraints	✓			Minimize path	CAD Simulation
(Çil et al., 2017)	RAL	ALB ATS	RALBP-II, and RALBSP		✓		Minimize sum cycle time	Beam Search, and Heuristic Algorithm
(Pellegrinelli et al., 2017)	RAC	APP	Motion planning, and collision free	✓			Minimize cost	Motion Planning, and Simulation
(Tsarouchi et al., 2017)	RAC	ASP	Searching and selecting sequence	✓		Total completion time, floor	Mathematical Model, and CAD Simulation	
(Andrzejewski et al., 2018)	RAC	ASP	Searching sequence, and location of the parts	✓		Minimize cycle time	Genetic Algorithm	
(Weckenborg & Spengler, 2019)	RAL	ALB	ALBP-HRC	✓		Minimize cost	MILP	
(Janardhanan et al., 2019)	RAL	ALB ATS	RALBP-II, and RALBSP		✓		Minimize cycle time	MILP, and Migratory Birds
(Maoudj & Bouzouia, 2019)	RAC	ATS	Operation scheduling, and allocation robots			✓	Minimize makespan	Dispatching Rules, and Dispatcher Multiple Agent System
(Mayer et al., 2019)	MSAS	ATS	Job and routing sequencing, AGV			✓	Minimize makespan, and route length	Agent-Based Simulation
(Stadnicka & Antonelli, 2019)	RAC	ATS	HRC, and Operation Scheduling		✓		Eliminate waste	Lean Tools
(Li et al., 2019)	RAL	ALB	RALBP-II (Two-sided)	✓			Minimize cycle time	MILP and Genetic Algorithm
(Casalino et al., 2019)	RAL	ALB ATS	RALBP-II, and Task sequencing	✓			Minimize idle time	Petri Nets
(Hagemann & Stark, 2020)	RAC	ASP	Robotic assembly system design			✓	Minimize cycle time	Simulated Annealing
(Weckenborg et al., 2020)	RAL	ALB ATS	ALBP-HRC	✓			Minimize cycle time	MILP and Hybrid Genetic Algorithm
(Mueller & Schmitt, 2020)	MSAS	ATS	Job sequencing, and			✓	Minimize throughput time, and maximize utilization	Precedence Graphs, and Discrete Event Simulation Model
(Rabbani et al., 2020)	RAL	ALB	ALBP-HRC		✓		Minimize number of workstations, and minimize cost	MILP, Multiobjective Particle Swarms Optimization
(Li et al., 2021)	RAL	ALB	ALBP-HRC	✓			Minimize cycle time, and robot purchasing cost	MILP, Multiobjective Migrating Bird Optimization
(Ferreira et al., 2021)	RAC	ATS	HRC, and operation scheduling		✓		Minimize total work time	Constraint Programming Model, and Genetic
(Zhang et al., 2021)	RAL	ALB ATS	RALBPS		✓		Minimize makespan, and energy consumption	Hybrid Multiobjective Dragonfly Algorithm
(Koltai et al., 2021)	RAL	ALB	ALBP-HRC	✓			Minimize number of workstations, and minimize cycle time	MILP
(Boschetti et al., 2021)	RAL	ALB ATS	ALBP-HRC	✓			Minimize makespan	MILP

Table 7
Chronological order of relevant publications in robotics assembly systems planning and scheduling (Continued)

Reference	Robotic Assembly System	Assembly Planning		Assembly Model			Objective	Solution Technique
		Problem	Sub-problem	Single	Mix	Multi		
(Nourmohammadi et al., 2022)	RAL	ALB	ALBP-HRC	✓			Minimize cycle time, and number of human and robot	MILP, and Simulated Annealing
(Schukat et al., 2022)	MSAS	ATS	Job sequencing and routing, workstation selecting, and AGV scheduling			✓	Minimize throughput time, and maximize utilization	Agent-Based Simulation
(Cai et al., 2022)	RAC	ASP	Robot collaborative hybrid assembly cell (HRCHAC)	✓			Minimize unit product assembly time, and maximize total task matching	MILP, and NSGA-II Algorithm
(Stecke & Mokhtarzadeh, 2022)	RAL	ALB ATS	ALBP-HRC, and MTS		✓		Minimize cycle time, and ergonomic risk	MILP, and Constrained Programming Model
(Yadav & Agrawal,	RAL	ALB ASP	R2SALB		✓		Minimize number of workstations	MILP
(Şahin & Tural, 2023)	RAL	ALB	Stochastic RALB	✓			Minimize cycle time	MILP
(Wang & Zhang, 2023)	RAL	ALB ATS	ALBP-HRC	✓			Maximize utilization	MILP, Cobb–Douglas Production Function, and Chance-Constrained Programming Model
(Z. Li et al., 2023)	RAL	ALB	ALBP-HRC		✓		Minimize cycle time	MILP, Bee Colony Algorithm, and Migrating Bird Optimization Algorithm
(Z.-K. Li et al., 2023)	MSAS	ATS	AGV scheduling			✓	Minimize cost	Nearest-Neighbor-Based Heuristic, and Discrete Invasive Weed Optimization Algorithm
(Y. Li et al., 2023)	MSAS	ATS	AGV scheduling			✓	Minimize transportation costs	MILP, Genetic Algorithm, and Nearest-Neighbor-Based Heuristic
(Miao et al., 2023)	RAC	ASP	Searching sequence, Selecting			✓	Minimize total work time	Autonomous Constraint Generation, CAD Simulation

5. Research findings

Below are some important facts and statistics that allow you to visualize the information more clearly.

- Regarding the three RAS configurations described in Section 2, it was found that the RACs have attracted the most attention from researchers due to the importance of planning and scheduling these systems to improve cell efficiency and productivity assembly. It should be noted that some publications address the planning of the RAC that is part of an assembly line, but the analysis focuses on the cell and not on its relationship with the assembly line. For their part, RALs have also attracted the attention of researchers mainly in the last decade, mainly motivated by the increase in robots on assembly lines in production plants around the world. There is less research on MSAS because it is a new system proposed at the beginning of the second decade of this century, although its configuration is based on the idea of having agile and modular systems, which was studied previously, MSAS was only proposed as such in 2014. It is important to note that this system has caught the attention of researchers in the last five years, making it a promising field for further research. On the other hand, there are no specific investigations in the RMSAS. The number of papers chronologically published by year is shown in Fig. 12.
- The configuration of the RAS depends on the type of assembly model, whether simple, mixed, or multi-model. Research has been mainly focused on investigating the planning and scheduling of the simple-model of both RAC and RAL systems. Research around mixed-models has been investigated more in RAL, while MSAS focuses on multi-models, this is because it is a system designed specifically for this type of environment. It is essential to emphasize that research on planning and scheduling in a single-model context has been more frequent in the years between 1990 and 2000. From that moment on, research has shifted towards mixed and multi-model models, reflecting changes in the market dynamics where consumers look for personalized products, demands are variable, and the life cycle of products is shorter. This

practice is applied today in many industries to respond to a wide variety of customer demands. Fig 13 shows the data from the publications with reference to the assembly models and configurations of the RAS.

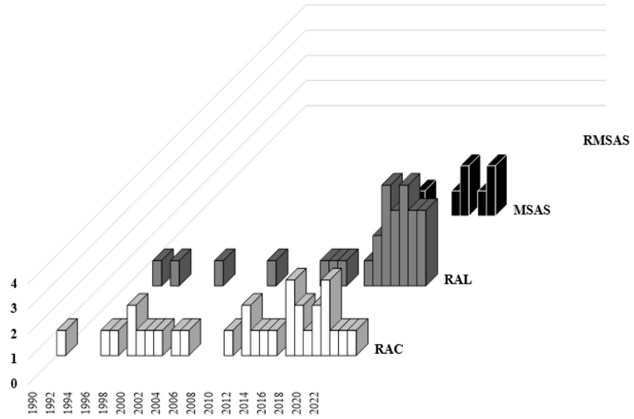

Fig. 8. Number of papers chronologically published by year classified according to RAS, and MSAS

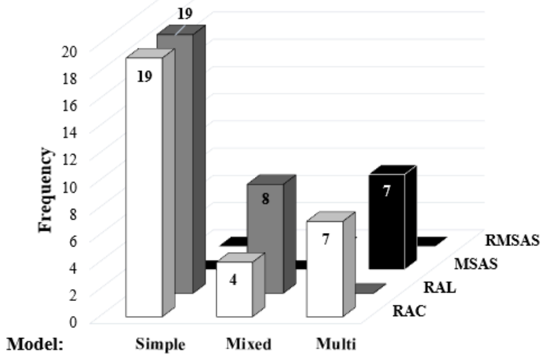

Fig. 9. Number of publications according to the classification of planning and scheduling problems in RAS, and MSAS

- In reference to the classification of planning and scheduling problems applied to RAS as described in Section 3, it is evident that RAC investigations predominantly emphasize design decisions, showing a heightened interest in addressing ASP and APP problems. Since 2011, publications addressing the ATS problem in RACs have become more prominent, attracting increased attention from researchers in recent years, primarily due to evolving demand conditions. On the other hand, the publications on RAL mainly address the problem of planning through the formulation of ALB focused on solving the planning of assembly lines of simple-model. When addressing the planning of mixed-model assembly lines in the RAL, research indicates that it is essential to tackle the ALB problem to address issues related to the allocation of robots and tasks on the assembly line. Subsequently, the ATS problem must be addressed to generate task assembly sequences for different product models. Only two publications were found that address the APP for an RAL and are focused on design decision making in an assembly workstation that is part of an assembly line. In the MSAS, only publications that addressed planning and scheduling problems through the conformation of the ATS were found. This is due to the fact that the layout of the MSAS is oriented to the multi-model assembly. The jobs and routing sequencing subproblems are the most studied in this type of configurations, and very few publications address the problem of AGV scheduling. Fig. 14 shows the data where the publications on planning and scheduling problems in RAS are classified and Fig. 15 shows the classification of the subproblems for each of the planning and scheduling problems in RAS, and MSAS.

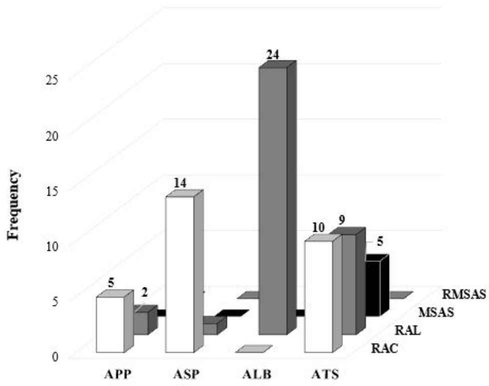

Fig. 14. Classification assembly planning and scheduling problems vs RAS and MSAS

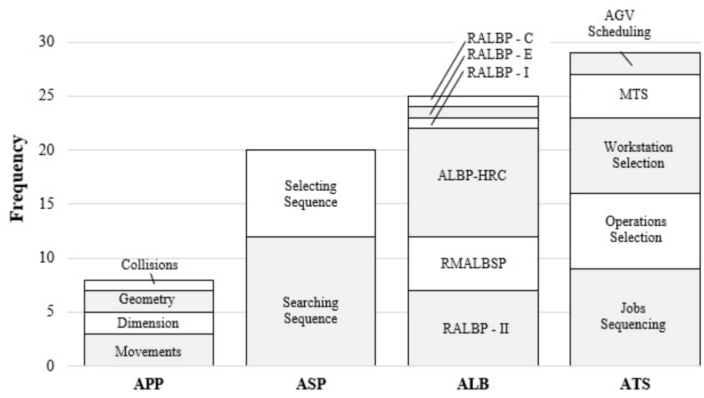

Fig. 10. Subproblems assembly planning and scheduling problems classification

- The solution approaches used to solve the RAS planning and scheduling problems are presented in Fig. 16, which shows the frequency of use of the different solution approaches where the AI approaches have a total of 26, followed by the heuristic approach with 23, the analytical approach with 18, and finally the simulation approach with 14. The GA is the most used within the AI approach, it is used since it generates good solutions to large planning and scheduling problems in reasonable times. In several articles, it is common to observe that planning and scheduling problems are addressed through the MILP formulation to solve small problems and find the optimal solution and thus be able to compare the performance of other solution approaches with the breadth of problems where the computational time spent by MILP is long. CAD simulation is a technique used to find a solution or to validate and adjust the solutions produced by other heuristic or metaheuristic techniques; its use is more common in APP and ASP, given the characteristics of the problems where it is required to solve the robot movements or assembly sequences, CAD simulations allow the observation of potential collisions between robots, a challenge that is difficult to address with other techniques. When it is necessary to analyze the occurrence of events such as failures and material shortages, among others, the simulation of unforeseen events allows for the evaluation of the performance of the program under these conditions, for these cases agent-based simulation and discrete event simulation model are used. In less quantity, different solution techniques described in Section 4 are presented.

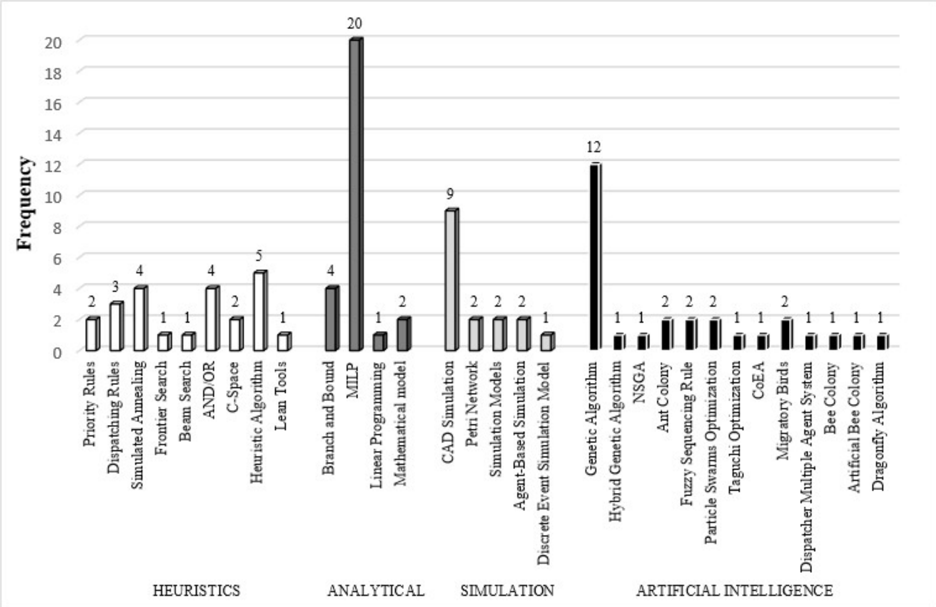

Fig. 11. Solution approaches

- Regarding the objective function, it can be observed that the performance measures focused on assembly time are the most used. When the objective function is to minimize, cycle time and makespan minimization are used most to solve RAS planning and scheduling problems, mainly in ALB and ATS problems. For APP and ASP problems, the assembly time is also very important, and in some investigations, other objectives such as the assembly travel distance are solved to reduce the time needed to carry out the assembly. Minimize the utilization of resources, including robots, workers or workstations, when there is a predefined cycle time. Another important objective is the cost. Some researchers have modeled the planning and scheduling problems to minimize the cost of the assembly based on different aspects such as reducing the number of robots needed in the system, reducing work idle time, or increasing the production rate of the system to reduce the unit cost of assembling a product. The other objective functions focused on minimizing are shown in Fig 17.

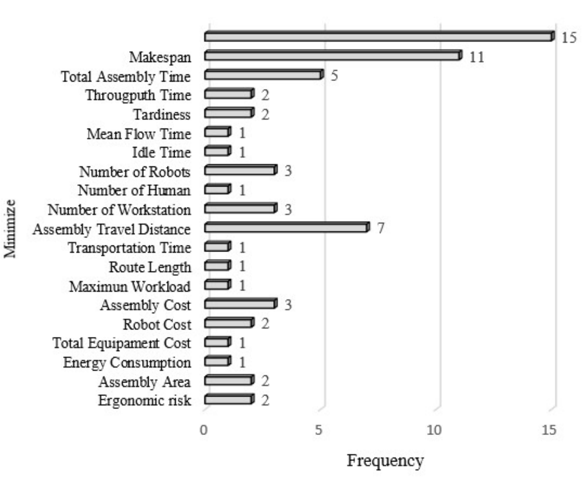

Fig. 17. Minimization objective function

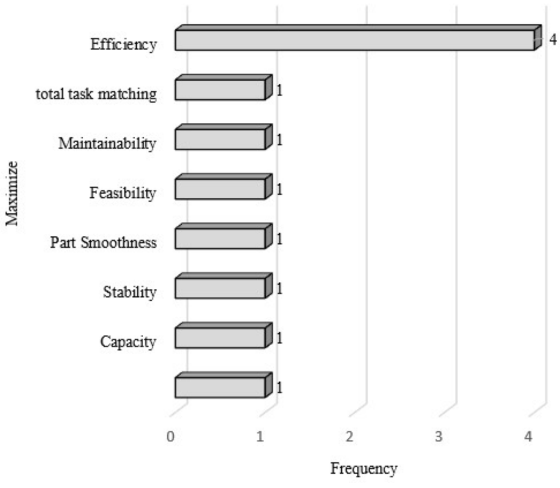

Fig. 12. Maximization objective function

In reference to the objective functions focused on maximizing, they are found in less quantity, being the objective function of maximizing the utilization of the system the most relevant. This is because the robots used in the assembly processes represent a high investment cost, and getting programs with the idle time of this equipment is not profitable. In Fig. 18, the other objective functions focused on maximizing are shown. The planning and scheduling of the RAS are addressed mostly through the formulation of a single objective function; the multi-objective formulation is only presented in 22% of the references analyzed. It should be noted that said multi-objective formulation is presented more frequently in recent years, which marks a trend in new research work.

6. Future research

Publications on RAS planning and scheduling problems continue to attract the attention of researchers who have modeled new scenarios with more variables, constraints, and configurations and proposed new approaches and techniques to solve these problems for new market challenges. However, there are still gaps waiting to be addressed by researchers to improve RAS efficiency and productivity. Below are some guidelines for future research that are the result of having done this research.

6.1 Human-robot collaboration

Research on RAS has largely focused on optimizing planning and scheduling for highly automated processes, with less attention on human-robot collaboration in systems with lower robotization. As noted in recent works (Chutima, 2022) and (Kheirabadi et al., 2022), there are opportunities to expand RAS research to address collaborative robots working alongside humans on assembly lines. These studies propose investigating new constraints when formulating optimization models for these collaborative assembly systems, such as incorporating ergonomic risk, preventing human-robot collisions, enabling parallel work, managing monotony and fatigue, optimizing task workflow between workers and cobots, bringing in temporary workers to address bottlenecks, and adapting production rates to fluctuations in demand when additional automation is impractical. By focusing research efforts on balancing assembly lines with both human and cobot workers, incorporating social and physiological factors, and allowing for flexible production rates, researchers can advance assembly systems that leverage the strengths of both human and automated agents.

6.2 Multi-model assembly

Research shows that most RAS planning focuses on design decisions, formulated as APP, ASP, and ALB. However, existing research predominantly focuses on simple-model assembly processes and lacks application to mixed or multi-model systems. With changing market dynamics and growing consumer preference for customized products, there is a growing need to address RAS planning and scheduling challenges within a multi-model framework. This approach must consider variables such as processing and setup times, tool changes, availability of materials, and storage capacity, among others.

6.3 Dynamic scheduling

RAS scheduling has garnered increasing research attention, though further work is needed to address dynamic conditions such as variable processing times, robot failures, urgent order changes, and material shortages. Additional complexity arises from sequence-dependent robot setup times/costs when changing tools and accessories, limited buffer space for parts and tools, and transport between workstations. Explicitly modeling parts, tools and their interactions with assembly robots could improve model accuracy and acceptability. The enhancements would move towards optimized planning and scheduling for next-generation RAS.

6.4 Matrix-structure assembly

Research on MSAS planning and scheduling is still limited due to the early stage of development of such systems. Initial publications focus primarily on sequencing tasks and determining assembly paths to optimize assembly cell utilization and minimize total path lengths. However, these studies often lack real-world application and validation. Regarding RMSAS, researchers have not yet systematically addressed planning and scheduling issues in this area. Given the promising future of RMSAS, researchers must formulate planning and scheduling problems that encompass additional practical complexities. These may include sequence-dependent setup times, limited buffer capacity, equipment failures, blockages, maintenance activities, material shortages, variable processing times depending on the number and capabilities of robots in the cell, and the existence of hybrid cells with robotic functions and human workers. This latter complexity introduces variability in assembly time and costs. In addition, certain assembly tasks may be carried out exclusively by robots, collaboratively or manually. In summary, addressing these challenges is crucial to improving the effectiveness of planning and scheduling in the context of MSAS and RMSAS, paving the way for their successful implementation in real-world scenarios.

6.5 Planning and scheduling with adaptation to demand

In the RAS planning and scheduling problems studied in this research, no publications were found that address planning problems for scenarios where the demand for products fluctuates or changes in delivery times are generated and it is necessary to adjust the rate production of the assembly workstations. In the RAL, it is assumed that the number of robots is fixed and that they belong to a single workstation, which means that processing time and task assignment are fixed. No studies were found where a RAL is formulated with mobile robots that can move from one workstation to another to vary the capacity and adjust it to the demand or where the production rate is varied by adding workers for collaborative work at the workstations. This same situation occurs with RACs where cell schedules are generated with a fixed number of robots and cases are not assumed in which the cell can vary its production rate by linking workers for collaborative work or including mobile robots that can move and carry out their assembly tasks in different assembly cells.

6.6 Waste elimination and green production

The current focus on improving efficiency and productivity along with reducing costs in assembly systems has gained great importance in the industry; therefore, planning focused on eliminating waste and reducing operations that do not add value has attracted the attention of researchers, and only one study was carried out found that it addresses this type of approach for RAS. In addition to conventional concepts in production, such as just-in-time, lean assembly, the human factor, or multi-skilled and disabled workers if human workers are allowed to work as part of the RAS, there are other problems in the world today that are getting a lot of attention recently in industry, such as green production, energy conservation, renewable energy, carbon footprint, etc., and need to be integrated into the formulation of future RAS.

7. Conclusion

Today's market conditions, where products are increasingly customized with short life cycles and fluctuating demands, have led industries around the world to install robots in their assembly systems to increase the productivity and flexibility of their system. However, this is not achieved only with the implementation of robots in assembly systems; it also requires solving RAS planning and scheduling problems, which has turned out to be a very challenging area for researchers due to its high complexity. RAS has received less attention from researchers than FMS, but in recent years, RAS has seen an increase due to the increasing application of robots in assembly processes. This article proposes a classification of RAS and the planning and scheduling problems that have been developed for their administration as well as a description of the approaches and solution techniques that researchers have proposed to solve problems of different complexity. The classifications made in Sections 2, 3, and 4 allowed us to establish the statistics on the publications made and to find the gaps in the formulation of RAS planning and scheduling problems for the development of future works.

Acknowledgement

This work was supported by the Faculty of Ciencias Empresariales of the Universidad Tecnológica de Pereira. We sincerely thank the Translation Agency of the Universidad de Antioquia for their valuable work providing help with the language, and assistance in writing the article.

References

Abd, K., Abhary, K., & Marian, R. (2011a). Scheduling and performance evaluation of robotic flexible assembly cells under different dispatching rules. https://doi.org/10.5729/ame.vol2.issue1.31

Abd, K., Abhary, K., & Marian, R. (2011b). A scheduling framework for robotic flexible assembly cells. *Applied Science and Engineering Progress*, *4*(1), 31-38. https://doi.org/https://ph02.tci-thaijo.org/index.php/ijast/article/view/67377

Abd, K., Abhary, K., & Marian, R. (2012). Intelligent modeling of scheduling robotic flexible assembly cells using fuzzy logic. https://doi.org/https://www.researchgate.net/publication/262398812_

Abd, K., Abhary, K., & Marian, R. (2016). Multi-objective optimisation of dynamic scheduling in robotic flexible assembly cells via fuzzy-based Taguchi approach. *Computers & Industrial Engineering*, *99*, 250-259. https://doi.org/https://doi.org/10.1016/j.cie.2016.07.028

Abd, K. K. (2015). *Intelligent scheduling of robotic flexible assembly cells*. Springer. https://doi.org/https://doi-org.ezproxy.utp.edu.co/10.1007/978-3-319-26296-3

Abd, K. K. (2016). Development of an Intelligent Methodology for Scheduling RFAC. In *Intelligent Scheduling of Robotic Flexible Assembly Cells* (pp. 31-47). Springer. https://doi.org/http://dx.doi.org/10.1007/978-3-319-26296-3_3

Alfadhlani, A. M. r., Toha, I. S., & Samadhi, T. A. (2019). Automatic Precedence Constraint Generation for Assembly Sequence Planning using a Three-Dimensional Solid Model. https://doi.org/https://doi.org/10.14716/ijtech.v10i2.3064

Allahverdi, A., Pesch, E., Pinedo, M., & Werner, F. (2018). Scheduling in manufacturing systems: new trends and perspectives. https://doi.org/https://doi.org/10.1080/00207543.2018.1504252

Andrzejewski, K., Cooper, M., Griffiths, C., & Giannetti, C. (2018). Optimisation process for robotic assembly of electronic components. *The International Journal of Advanced Manufacturing Technology*, *99*, 2523-2535. https://doi.org/https://doi.org/10.1007/s00170-018-2645-y

Basran, J. S., Petriu, E. M., & Petriu, D. C. (1997). Flexible agent-based robotic assembly cell. Proceedings of International Conference on Robotics and Automation,

Blazewicz, J. (2013). Scheduling computer and manufacturing processes. https://doi.org/https://doi.org/10.1057/palgrave.jors.2600793

Boschetti, G., Faccio, M., Milanese, M., & Minto, R. (2021). C-ALB (Collaborative Assembly Line Balancing): a new approach in cobot solutions. *The International Journal of Advanced Manufacturing Technology*, *116*, 3027-3042. https://doi.org/https://doi.org/10.1007/s00170-021-07565-7

Boysen, N., Fliedner, M., & Scholl, A. (2009). Production planning of mixed-model assembly lines: overview and extensions. *Production Planning and Control*, *20*(5), 455-471. https://doi.org/https://doi.org/10.1016/S0360-8352(01)00065-1

Brucker, P. (2006). *Scheduling algorithms* (Vol. 50).

Bukchin, J., & Tzur, M. (2000). Design of flexible assembly line to minimize equipment cost. *Iie transactions*, *32*(7), 585-598. https://doi.org/https://doi.org/10.1023/A:1007646714909

Cai, M., Liang, R., Luo, X., & Liu, C. (2022). Task allocation strategies considering task matching and ergonomics in the human-robot collaborative hybrid assembly cell. *International Journal of Production Research*, 1-20. https://doi.org/https://doi.org/10.1080/00207543.2022.2147234

Capacho Betancourt, L., & Pastor Moreno, R. (2004). Generación de secuencias de montaje y equilibrado de líneas. https://doi.org/http://hdl.handle.net/2117/509

Caprihan, R., Kumar, A., & Stecke, K. E. (2013). Evaluation of the impact of information delays on flexible manufacturing systems performance in dynamic scheduling environments. *The International Journal of Advanced Manufacturing Technology*, *67*(1-4), 311-338. https://doi.org/https://doi.org/10.1007/s00170-013-4755-x

Carlson, J. S., Spensieri, D., Söderberg, R., Bohlin, R., & Lindkvist, L. (2013). Non-nominal path planning for robust robotic assembly. *Journal of Manufacturing Systems*, *32*(3), 429-435. https://doi.org/https://doi.org/10.1016/j.jmsy.2013.04.013

Casalino, A., Zanchettin, A. M., Piroddi, L., & Rocco, P. (2019). Optimal scheduling of human–robot collaborative assembly operations with time petri nets. *IEEE Transactions on Automation Science and Engineering*, *18*(1), 70-84. https://doi.org/https://doi.org/10.1109/TASE.2019.2932150

Chan, F. T., & Chan, H. K. (2004). A comprehensive survey and future trend of simulation study on FMS scheduling. *Journal of Intelligent Manufacturing*, *15*(1), 87-102. https://doi.org/https://doi.org/10.1023/B:JIMS.0000010077.27141.be

Chang, H., & Li, T.-Y. (1995). Assembly maintainability study with motion planning. Proceedings of 1995 IEEE International Conference on Robotics and Automation,

Chen, J., & Lee, C. Y. (1999). General multiprocessor task scheduling. *Naval Research Logistics (NRL)*, *46*(1), 57-74. https://doi.org/https://doi.org/10.1002/(SICI)1520-6750(199902)46:1<57::AID-NAV4>3.0.CO;2-H

Chincholkar, A., & Chetty, O. K. (1996). Stochastic coloured Petri nets for modelling and evaluation, and heuristic rule base for scheduling of FMS. *The International Journal of Advanced Manufacturing Technology*, *12*(5), 339-348. https://doi.org/https://doi.org/10.1007/BF01179809

Chutima, P. (2022). A comprehensive review of robotic assembly line balancing problem. *Journal of Intelligent Manufacturing*, *33*(1), 1-34. https://doi.org/https://doi.org/10.1007/s10845-020-01641-7

Çil, Z. A., Li, Z., Mete, S., & Özceylan, E. (2020). Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human–robot collaboration. *Applied Soft Computing*, *93*, 106394. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106394

Çil, Z. A., Mete, S., & Ağpak, K. (2017). Analysis of the type II robotic mixed-model assembly line balancing problem. *Engineering Optimization*, *49*(6), 990-1009. https://doi.org/https://doi.org/10.1080/0305215X.2016.1230208

Cortés, J., Jaillet, L., & Siméon, T. (2008). Disassembly path planning for complex articulated objects. *IEEE Transactions on Robotics*, *24*(2), 475-481. https://doi.org/10.1109/TRO.2008.915464

Cottrez, F., & Van Brussel, H. (1989). SESFAC: A scheduling expert system for flexible assembly cells. Proceedings of the IEEE Intern. Conf. on Robotics and Automation,

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. https://doi.org/https://doi.org/10.1007/s10732-014-9239-0

De Fazio, T., & Whitney, D. (1987). Simplified generation of all mechanical assembly sequences. *IEEE Journal on Robotics and Automation*, *3*(6), 640-658. https://doi.org/10.1109/JRA.1987.1087132

Del Valle, C., & Camacho, E. (1996). Automatic assembly task assignment for a multirobot environment. *Control engineering practice*, *4*(7), 915-921. https://doi.org/https://doi.org/10.1016/0967-0661(96)00089-5

Delchambre, A. (2012). *Computer-aided assembly planning*. Springer Science & Business Media.

Ferreira, C., Figueira, G., & Amorim, P. (2021). Scheduling human-robot teams in collaborative working cells. *International Journal of Production Economics*, *235*, 108094. https://doi.org/https://doi.org/10.1016/j.ijpe.2021.108094

Framinan, J. M., Leisten, R., & García, R. R. (2014). Manufacturing scheduling systems. *An integrated view on Models, Methods and Tools*, 51-63. https://doi.org/https://doi.org/10.1007/978-1-4471-6272-8

French, S. (1982). Sequencing and scheduling. *An Introduction to the Mathematics of the Job-shop*. https://doi.org/https://doi.org/10.1002/net.3230130218

Ghandi, S., & Masehian, E. (2015). Review and taxonomies of assembly and disassembly path planning problems and approaches. *Computer-Aided Design*, *67*, 58-86. https://doi.org/https://doi.org/10.1016/j.cad.2015.05.001

Glibert, P.-R., Coupez, D., Peng, Y., & Delchambre, A. (1990). Scheduling of a multi-robot assembly cell. *Computer Integrated Manufacturing Systems*, *3*(4), 236-245. https://doi.org/https://doi.org/10.1016/0951-5240(90)90064-L

Göppert, A., Schukat, E., Burggräf, P., & Schmitt, R. H. (2021). Agile hybrid assembly systems: bridging the gap between line and matrix configurations. In *Advances in Automotive Production Technology–Theory and Application* (pp. 3-11). Springer. https://doi.org/https://doi.org/10.1007/978-3-662-62962-8_1

Greschke, P., Schönemann, M., Thiede, S., & Herrmann, C. (2014). Matrix structures for high volumes and flexibility in production systems. *Procedia CIRP*, *17*, 160-165. https://doi.org/https://doi.org/10.1016/j.procir.2014.02.040

Gultekin, H., Akturk, M. S., & Karasan, O. E. (2008). Scheduling in robotic cells: process flexibility and cell layout. *International Journal of Production Research*, *46*(8), 2105-2121. https://doi.org/http://dx.doi.org/10.1080/00207540601100262
Gultekin, H., Coban, B., & Akhlaghi, V. E. (2018). Cyclic scheduling of parts and robot moves in m-machine robotic cells. *Computers & Operations Research*, *90*, 161-172. https://doi.org/https://doi.org/10.1016/j.cor.2017.09.018
Hagemann, S., & Stark, R. (2020). An optimal algorithm for the robotic assembly system design problem: An industrial case study. *CIRP Journal of Manufacturing Science and Technology*, *31*, 500-513. https://doi.org/https://doi.org/10.1016/j.cirpj.2020.08.002
Halperin, D., Latombe, J.-C., & Wilson, R. H. (2000). A general framework for assembly planning: The motion space approach. *Algorithmica*, *26*(3), 577-601. https://doi.org/10.1007=s004539910025
Hofmann, C., Brakemeier, N., Krahe, C., Stricker, N., & Lanza, G. (2018). The impact of routing and operation flexibility on the performance of matrix production compared to a production line. Congress of the German Academic Association for Production Technology,
Hsu, H.-H., & Fu, L.-C. (1995). Fully automated robotic assembly cell: scheduling and simulation. Proceedings of 1995 IEEE International Conference on Robotics and Automation,
Hui, C., Yuan, L., & Kai-fu, Z. (2009). Efficient method of assembly sequence planning based on GAAA and optimizing by assembly path feedback for complex product. *The International Journal of Advanced Manufacturing Technology*, *42*(11), 1187-1204. https://doi.org/https://doi.org/10.1007/s00170-008-1661-8
Izui, K., Murakumo, Y., Suemitsu, I., Nishiwaki, S., Noda, A., & Nagatani, T. (2013). Multiobjective layout optimization of robotic cellular manufacturing systems. *Computers & Industrial Engineering*, *64*(2), 537-544. https://doi.org/https://doi.org/10.1016/j.cie.2012.12.003
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. https://doi.org/https://doi.org/10.1016/j.apm.2018.08.016
Jiménez, P. (2013). Survey on assembly sequencing: a combinatorial and geometrical perspective. *Journal of Intelligent Manufacturing*, *24*(2), 235-250. https://doi.org/10.1007/s10845-011-0578-5
Joseph, O., & Sridharan, R. (2011). Effects of routing flexibility, sequencing flexibility and scheduling decision rules on the performance of a flexible manufacturing system. *The International Journal of Advanced Manufacturing Technology*, *56*, 291-306. https://doi.org/https://doi.org/10.1007/s00170-011-3158-0
Kheirabadi, M., Keivanpour, S., Chinniah, Y., & Frayret, J. (2022). A Review on Collaborative Robot Assembly Line Balancing Problems. *IFAC-PapersOnLine*, *55*(10), 2779-2784. https://doi.org/https://doi.org/10.1016/j.ifacol.2022.10.151
Khouja, M., Booth, D. E., Suh, M., & Mahaney Jr, J. K. (2000). Statistical procedures for task assignment and robot selection in assembly cells. *International Journal of Computer Integrated Manufacturing*, *13*(2), 95-106. https://doi.org/https://doi.org/10.1080/095119200129957
Kojima, S., & Hashimoto, H. (1999). 3-D CAD data oriented self-planning of assembly robot cell systems. 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (Cat. No. 99TH8399),
Koltai, T., Dimény, I., Gallina, V., Gaal, A., & Sepe, C. (2021). An analysis of task assignment and cycle times when robots are added to human-operated assembly lines, using mathematical programming models. *International Journal of Production Economics*, *242*, 108292. https://doi.org/https://doi.org/10.1016/j.ijpe.2021.108292
Lee, J.-K., & Lee, T.-E. (2002). Automata-based supervisory control logic design for a multi-robot assembly cell. *International Journal of Computer Integrated Manufacturing*, *15*(4), 319-334. https://doi.org/https://doi.org/10.1080/09511920110078097
Levitin, G., Rubinovitz, J., & Shnits, B. (2006). A genetic algorithm for robotic assembly line balancing. *European Journal of Operational Research*, *168*(3), 811-825. https://doi.org/https://doi.org/10.1016/j.ejor.2004.07.030
Li, Y., Zou, J., Jia, Y., Meng, L., & Zou, W. (2023). An improved genetic algorithm for multi-AGV dispatching problem with unloading setup time in a matrix manufacturing workshop. *International Journal of Industrial Engineering Computations*, *14*(4), 767-784. https://doi.org/10.5267/j.ijiec.2023.7.002
Li, Z.-K., Sang, H.-Y., Li, J.-Q., Han, Y.-Y., Gao, K.-Z., Zheng, Z.-X., & Liu, L.-l. (2023). Invasive weed optimization for multi-AGVs dispatching problem in a matrix manufacturing workshop. *Swarm and Evolutionary Computation*, *77*, 101227. https://doi.org/https://doi.org/10.1016/j.swevo.2023.101227
Li, Z., Janardhanan, M., Tang, Q., & Zhang, Z. (2023). Models and algorithms for U-shaped assembly line balancing problem with collaborative robots. *Soft Computing*, 1-21. https://doi.org/https://doi.org/10.1007/s00500-023-08130-y
Li, Z., Janardhanan, M. N., & Tang, Q. (2021). Multi-objective migrating bird optimization algorithm for cost-oriented assembly line balancing problem with collaborative robots. *Neural Computing and Applications*, 1-22. https://doi.org/https://doi.org/10.1007/s00521-020-05610-2
Li, Z., Janardhanan, M. N., Tang, Q., & Nielsen, P. (2018). Mathematical model and metaheuristics for simultaneous balancing and sequencing of a robotic mixed-model assembly line. *Engineering Optimization*, *50*(5), 877-893. https://doi.org/https://doi.org/10.1080/0305215X.2017.1351963
Li, Z., Janardhanan, M. N., Tang, Q., & Ponnambalam, S. (2019). Model and metaheuristics for robotic two-sided assembly line balancing problems with setup times. *Swarm and Evolutionary Computation*, *50*, 100567. https://doi.org/https://doi.org/10.1016/j.swevo.2019.100567

Lin, H.-C., Egbelu, P. J., & WU, C.-T. (1995). A two-robot printed circuit board assembly system. *International Journal of Computer Integrated Manufacturing*, *8*(1), 21-31. https://doi.org/https://doi.org/10.1080/09511929508944626

Lotter, B. (2013). *Manufacturing assembly handbook*. Butterworth-Heinemann. https://doi.org/https://doi.org/10.1016/C2013-0-04107-2

Lozano-Pérez, T., Mason, M. T., & Taylor, R. H. (1983). Automatic synthesis of fine-motion strategies for robots. *Robotics Research*, *1*, 65-96. https://doi.org/https://doi.org/10.1177/027836498400300101

Maoudj, A., & Bouzouia, B. (2019). Distributed multi-agent scheduling and control system for robotic flexible assembly cells. *Journal of Intelligent Manufacturing*, *30*(4), 1629-1644. https://doi.org/https://doi.org/10.1007/s10845-017-1345-z

Marian, R., Kargas, A., Luong, L., & Abhary, K. (2003). A framework to planning robotic flexible assembly cells. https://doi.org/http://hdl.handle.net/102.100.100/192866?index=1

Marian, R. M., Luong, L. H., & Abhary, K. (2003). Assembly sequence planning and optimisation using genetic algorithms: part I. Automatic generation of feasible assembly sequences. *Applied Soft Computing*, *2*(3), 223-253. https://doi.org/https://doi.org/10.1016/S1568-4946(02)00064-9

Mayer, S., Höhme, N., Gankin, D., & Endisch, C. (2019). Adaptive Production Control in a Modular Assembly System–Towards an Agent-based Approach. 2019 IEEE 17th international conference on industrial informatics (INDIN),

Miao, M.-P., Sang, H.-Y., Wang, Y.-T., Zhang, B., & Tian, M.-X. (2023). Joint scheduling of parallel machines and AGVs with sequence-dependent setup times in a matrix workshop. *Computers & Industrial Engineering*, 109621. https://doi.org/https://doi.org/10.1016/j.cie.2023.109621

Michniewicz, J., & Reinhart, G. (2016). Cyber-Physical-Robotics–Modelling of modular robot cells for automated planning and execution of assembly tasks. *Mechatronics*, *34*, 170-180. https://doi.org/https://doi.org/10.1016/j.mechatronics.2015.04.012

Michniewicz, J., Reinhart, G., & Boschert, S. (2016). CAD-based automated assembly planning for variable products in modular production systems. *Procedia CIRP*, *44*, 44-49. https://doi.org/https://doi.org/10.1016/j.procir.2016.02.016

Mohamed, S. B., Petty, D. J., Harrison, D. K., & Rigby, R. (2001). A cell management system to support robotic assembly. *The International Journal of Advanced Manufacturing Technology*, *18*(8), 598-604. https://doi.org/https://link.springer.com/content/pdf/10.1007/s001700170037.pdf

Morato, C., Kaipa, K. N., & Gupta, S. K. (2013). Improving assembly precedence constraint generation by utilizing motion planning and part interaction clusters. *Computer-Aided Design*, *45*(11), 1349-1364. https://doi.org/https://doi.org/10.1016/j.cad.2013.06.005

Mueller, D., & Schmitt, T. V. (2020). Production planning in autonomous and matrix-structured assembly systems: effects of similarity of precedence graphs on order release sequencing. *Procedia CIRP*, *93*, 1358-1363. https://doi.org/https://doi.org/10.1016/j.procir.2020.06.002

Nof, S. Y., & Drezner, Z. (1993). The multiple-robot assembly plan problem. *Journal of Intelligent and Robotic Systems*, *7*(1), 57-71. https://doi.org/https://doi.org/10.1007/BF01258212

Nourmohammadi, A., Fathi, M., & Ng, A. H. (2022). Balancing and scheduling assembly lines with human-robot collaboration tasks. *Computers & Operations Research*, *140*, 105674. https://doi.org/https://doi.org/10.1016/j.cor.2021.105674

Pan, C. (2005). *Integrating CAD files and automatic assembly sequence planning*. Iowa State University. https://doi.org/https://doi.org/10.31274/rtd-180813-15396

Pelagagge, P. M., Cardarelli, G., & Palumbo, M. (1995). Design criteria for cooperating robots assembly cells. *Journal of Manufacturing Systems*, *14*(4), 219-229. https://doi.org/https://doi.org/10.1016/0278-6125(95)98875-7

Pellegrinelli, S., Pedrocchi, N., Tosatti, L. M., Fischer, A., & Tolio, T. (2017). Multi-robot spot-welding cells for car-body assembly: Design and motion planning. *Robotics and Computer-Integrated Manufacturing*, *44*, 97-116. https://doi.org/https://doi.org/10.1016/j.rcim.2016.08.006

Pinedo, M. (2012). Advanced Single Machine Models. In *Scheduling; Theory, Algorithms, and Systems* (4 ed., pp. 70 -77). Springer.

Rabbani, M., Behbahan, S. Z. B., & Farrokhi-Asl, H. (2020). The collaboration of human-robot in mixed-model four-sided assembly line balancing problem. *Journal of Intelligent & Robotic Systems*, *100*, 71-81. https://doi.org/https://doi.org/10.1007/s10846-020-01177-1

Rakshit, S., & Akella, S. (2014). The influence of motion paths and assembly sequences on the stability of assemblies. *IEEE Transactions on Automation Science and Engineering*, *12*(2), 615-627. https://doi.org/10.1109/TASE.2014.2345569

Rosell, J. (2004). Assembly and task planning using Petri nets: a survey. *Proceedings of the institution of mechanical engineers, part B: journal of engineering manufacture*, *218*(8), 987-994. https://doi.org/https://doi.org/10.1243/0954405041486019

Rubinovitz, J. (1991). Design and balancing of robotic assembly lines. 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. https://doi.org/https://doi.org/10.1016/S0007-8506(07)62494-9

Şahin, M. C., & Tural, M. K. (2023). Robotic stochastic assembly line balancing. *Flexible Services and Manufacturing Journal*, 1-40. https://doi.org/https://doi.org/10.1007/s10696-023-09494-x

Sawik, T. (1998). Simultaneous loading, routing, and assembly plan selection in a flexible assembly system. *Mathematical and computer modelling*, *28*(9), 19-29. https://doi.org/https://doi.org/10.1016/S0895-7177(98)00142-3

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. https://doi.org/https://doi.org/10.1016/j.ejor.2004.07.022

Scholz-Reiter, B., & Freitag, M. (2007). Autonomous processes in assembly systems. *CIRP annals*, *56*(2), 712-729. https://doi.org/https://doi.org/10.1016/j.cirp.2007.10.002

Schönemann, M., Herrmann, C., Greschke, P., & Thiede, S. (2015). Simulation of matrix-structured manufacturing systems. *Journal of Manufacturing Systems*, *37*, 104-112. https://doi.org/https://doi.org/10.1016/j.jmsy.2015.09.002

Schukat, E., Rachner, J., Maidl, A., Göppert, A., Adlon, T., Burggräf, P., & Schmitt, R. H. (2022). Agent-based Order Release in Matrix-Structured Assembly Systems. Proceedings of the Conference on Production Systems and Logistics: CPSL 2022,

Stadnicka, D., & Antonelli, D. (2019). Human-robot collaborative work cell implementation through lean thinking. *International Journal of Computer Integrated Manufacturing*, *32*(6), 580-595. https://doi.org/https://doi.org/10.1080/0951192X.2019.1599437

Stecke, K. E., & Mokhtarzadeh, M. (2022). Balancing collaborative human–robot assembly lines to optimise cycle time and ergonomic risk. *International Journal of Production Research*, *60*(1), 25-47. https://doi.org/https://doi.org/10.1080/00207543.2021.1989077

Su, C.-T., & Fu, H.-P. (1998). A simulated annealing heuristic for robotics assembly using the dynamic pick-and-place model. *Production Planning & Control*, *9*(8), 795-802. https://doi.org/https://doi.org/10.1080/095372898233560

Thomas, U., & Wahl, F. M. (2001). A system for automatic planning, evaluation and execution of assembly sequences for industrial robots. Proceedings 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180),

Trierweiler, M., Foith-Förster, P., & Bauernhansl, T. (2020). Changeability of matrix assembly systems. *Procedia CIRP*, *93*, 1127-1132. https://doi.org/https://doi.org/10.1016/j.procir.2020.04.029

Tsarouchi, P., Michalos, G., Makris, S., Athanasatos, T., Dimoulas, K., & Chryssolouris, G. (2017). On a human–robot workplace design and task allocation system. *International Journal of Computer Integrated Manufacturing*, *30*(12), 1272-1279. https://doi.org/https://doi.org/10.1080/0951192X.2017.1307524

Tuncel, G., & Bayhan, G. M. (2007). Applications of Petri nets in production scheduling: a review. *The International Journal of Advanced Manufacturing Technology*, *34*(7-8), 762-773. https://doi.org/https://doi.org/10.1007/s00170-006-0640-1

Uddin, M. K., & Lastra, J. L. M. (2011). Assembly line balancing and sequencing. *Assembly Line–Theory and Practice*, 13-36. https://doi.org/10.5772/19953

Wang, D., & Zhang, J. (2023). Flow shop scheduling with human–robot collaboration: a joint chance-constrained programming approach. *International Journal of Production Research*, 1-21. https://doi.org/https://doi.org/10.1080/00207543.2023.2181025

Weckenborg, C., Kieckhäfer, K., Müller, C., Grunewald, M., & Spengler, T. S. (2020). Balancing of assembly lines with collaborative robots. *Business Research*, *13*(1), 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. https://doi.org/https://doi.org/10.1016/j.ifacol.2019.11.473

Wilson, R. H., Kavraki, L., Latombe, J.-C., & Lozano-Pérez, T. (1995). Two-handed assembly sequencing. *The International journal of robotics research*, *14*(4), 335-350. https://doi.org/https://doi.org/10.1177/027836499501400403

Wolter, J., Chakrabarty, S., & Tsao, J. (1992). Mating constraint languages for assembly sequence planning. ICRA,

Yadav, A., & Agrawal, S. (2022). Mathematical model for robotic two-sided assembly line balancing problem with zoning constraints. *International Journal of System Assurance Engineering and Management*, *13*(1), 395-408. https://doi.org/https://doi.org/10.1007/s13198-021-01284-8

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. https://doi.org/https://doi.org/10.1016/j.jmsy.2011.10.002

Zhang, B., Xu, L., & Zhang, J. (2021). Balancing and sequencing problem of mixed-model U-shaped robotic assembly line: Mathematical model and dragonfly algorithm based approach. *Applied Soft Computing*, *98*, 106739. https://doi.org/https://doi.org/10.1016/j.asoc.2020.106739

Zhang, L., Huang, X., Kim, Y. J., & Manocha, D. (2008). D-plan: Efficient collision-free path computation for part removal and disassembly. *Computer-Aided Design and Applications*, *5*(6), 774-786. https://doi.org/10.3722/cadaps.2008.774-786