Management Systems in Production Engineering

2024, Volume 32, Issue 2, pp. 162-173

Date of submission of the article to the Editor: 11/2023
Date of acceptance of the article by the Editor: 04/2024

DOI 10.2478/mspe-2024-0016

OPTIMISATION OF SIMPLE ASSEMBLY LINE BALANCING PROBLEM TYPE E: A SYSTEMATIC LITERATURE REVIEW

Sana El Machouti
LASTIMI Laboratory Mohammedia School of Engineers (EMI)
Mohammed V University in Rabat
Center of Excellence in Logistics (CELOG), School of Textile and Clothing (ESITH)

Mustapha Hlyal
Center of Excellence in Logistics (CELOG), School of Textile and Clothing (ESITH)

Amel Babay
Textile Engineering Laboratory, ISET Ksar Hellal, Monastir University

Jamila El Alami
LASTIMI Laboratory, High School of Technology (EST), Mohammed V University in Rabat

Abstract:
Numerous research studies have focused on the Simple Assembly Line Balancing Problem (SALBP). Although many of these studies have attempted to minimise the number of workstations for a given cycle time (SALBP-1) or minimise the cycle time for a given number of workstations (SALBP-2), few have focused on the E-type of the SALBP, which is a general NP-hard problem. Therefore, a Systematic Literature Review (SLR) in this area is extremely important. The SALB-E problem involves scheduling a given set of tasks in an orderly sequence of workstations, where task precedence relations are satisfied, and finding the optimal pair of number of workstations and cycle time that maximises line efficiency. The aim of this systematic review was to examine existing research on SALB-E issues and predict future directions for studies on this topic. A systematic methodology was used to review papers published between 1995 and 2023, selected from the Scopus, ScienceDirect, and Google Scholar databases. The review showed that future studies should focus on multi- and mixed-model U-shaped and two-sided lines in the SALBP-E. In addition, it may be important to evaluate multiple objective functions rather than just a single one. To solve this problem, it is preferable to develop and refine genetic algorithms and investigate new approaches within the metaheuristic framework.

Key words: *SALBP-E, SALBP, assembly line balancing, line efficiency, manufacturing optimisation*

INTRODUCTION

Today, the global market is extremely competitive, and manufacturers must engage in all methods of product development to improve the efficiency and effectiveness of their manufacturing processes [1, 2, 3]. As a result, assembly lines have become increasingly important in the production of high-volume, standardised products, particularly in the automotive industry [4, 5, 6].

An assembly line is a series of workstations where components are sequentially assembled to form a semi-finished assembly, which then moves from one workstation to the next until a complete assembly is achieved. To achieve the required performance, the assembly line must be balanced by assigning tasks to workstations in a way that meets the assembly goal, demand, and constraints [7, 8]. Due to technical and organisational constraints, priority constraints between tasks must be respected. These components can be summarised and represented using an oriented acyclic graph called a "precedence graph" [9], such as the graph shown in Fig. 1. The number of each task is indicated inside each circle, and the numbers in the upper right corner of each node indicate the time of the task [10]. We assume that the tasks are numbered in topological order. As a result, assembly line balancing (ALB) is a useful technique for improving productivity and efficiency.

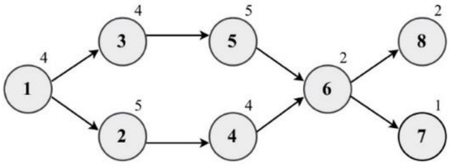

Fig. 1 An example of a precedence diagram

The ALB problem (ALBP) is a well-known problem in operations research. Its goal is to determine the optimal allocation of tasks between workstations. Salveson [11] provided the first mathematical model for this problem. Baker [12] defines ALBPs as NP-hard combinatorial problems. Generalised ALBPs (GALBPs) and Simple ALBPs (SALBPs) are two categories of ALB problems based on their assumptions, constraints, and objectives [13]. The SALBP is a basic type of problem that has been well studied. For more information, interested readers can refer to the surveys of many articles [14, 15, 16, 17, 18].

Two types of techniques can be used to solve SALBPs: *approximation methods* (including heuristic and metaheuristic methods) and *exact methods* [19]. Four types of SALBPs are often treated, depending on the objective function [9, 17]. SALBP-1 aims to allocate tasks to workstations in such a way that the number of workstations M is minimised for a given cycle time C_T, while SALBP-2 aims to minimise cycle time C_T taking into account the number of workstations M. The most general type of problem is SALBP-E, which maximises line efficiency E while minimising C_T and M due to their dependency. Category F (SALBP-F) is a feasibility problem that examines whether there is a feasible line balance for a given M and C_T configuration [20]. Fig. 2 shows the classification of line balance problems.

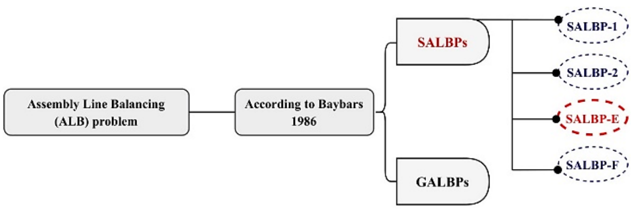

Fig. 2 ALB problem classification

Assembly line performance can be measured by more than one factor, such as line efficiency and idle time.

Line efficiency (E) is the proportion of the assembly line that is in use. 100% represents perfect balance. The following equation can be used to determine line efficiency [21]:

$$E = \frac{\text{Total working time}}{M \times C_T}\% \tag{1}$$

Idle time (I_t) is defined as the time when a worker or workstation is not engaged in any work. Perfect equilibrium is achieved when no idle time is present. The following is how idle time is calculated [21]:

$$I_t = (M \times C_T) - \text{total working time} \tag{2}$$

The SALBP-E is intended for versions where improved line efficiency is required. It is a more general problem that combines SALBP types 1 and 2. Furthermore, the majority of existing studies have focused on SALBP-1 [22, 23, 24] and SALBP-2 [25, 26, 27]. As a result, the Type E problem exists when the production line is flexible enough to change the number of workstations as well as the cycle time. However, such settings cannot be changed arbitrarily [14].

As SALBP-E is more complex than SALBP-1 and SALBP-2 (due to its non-linear form), few researchers have included SALBP-E in their studies. As a result, there is a need for a systematic literature review of simple E-type assembly line balancing to guide and assist future researchers interested in this topic. The rest of the paper is structured as follows: Section 2 provides a detailed description of SALBP-E and its mathematical formulation, as well as an analysis of work related to SALB problem type E. The methodology used to select the scientific literature is described in Section 3, and Section 4 presents the results and discusses trends in SALBP-E research. Finally, Section 5 discusses future directions for SALBP-E research and concludes.

LITERATURE REVIEW

This section introduces the notations used to describe SALBP-E and presents the mathematical model for this problem. The 23 selected articles are then analysed using the method described in the methodology section.

Problem descriptions, notations, and SALBP-E mathematical formulation

Notations

N	The number of tasks $k = (1, \dots, N)$
M	The number of workstations $j = (1, \dots, M)$
T_k	The working period for the task k
C_T	The cycle time
T_{sum}	The total duration of time spent on all tasks
T_{max}	The maximum duration of the task
M_{max}	The upper bound of workstations
M_{min}	The lower bound of workstations
C_{Tmin}	The lower bound of the cycle time C_T
C_{Tmax}	The upper bound of the cycle time C_T
I_t	The idle time
P_k	The preceding tasks of task k

Decision variable

$$x_{kj} \in \{0,1\} = \begin{cases} 1, \text{if task } k \text{ is affected} \\ \quad \text{by workstation } j, \\ 0, \qquad \text{otherwise.} \end{cases}$$

To produce a product on a basic assembly line, the work must be divided into a series of primary operations called tasks $k = 1, \dots, N$. When executing a task k, a task time T_k is used, which is the weight of the task k in the priority graph; the priority constraints of the task must be obeyed as in (4). The task durations are assumed to be deterministic with integer values. Another constraint is that each task must be assigned to exactly one workstation, as in (5), even though the workstations may contain many tasks.

Due to the constant movement of the line, each product unit remains at each workstation for a defined time, called the cycle time C_T. For a given common cycle time, line balancing is only possible if none of the station times exceeds C_T, as in (6). The idle times are also integers due to the integrality of the cycle time. The objective function in (3) specifies the maximisation of the line efficiency E and the simultaneous minimisation of C_T and M. Lastly, the integrity constraint in (7) ensures that the decision variables have the proper binary value.

$$\boldsymbol{Min\, L = C_T \times M} \tag{3}$$

Subject to,

$$\boldsymbol{\sum_{j=1}^{M} j x_{lj} \leq \sum_{j=1}^{M} j x_{kj}\,,\ \forall l \in P_k, \forall k} \tag{4}$$

$$\boldsymbol{\sum_{j=1}^{M} x_{kj} = 1\,,\ \forall k} \tag{5}$$

$$\boldsymbol{\sum_{k=1}^{N} T_k x_{kj} \leq C_T,\ \forall j} \tag{6}$$

$$\boldsymbol{x_{kj} \in \{0,1\},\ \forall k, j} \tag{7}$$

Reducing the number of workstations or line cycle time is equivalent to maximising assembly line efficiency, i.e., minimising idle time as much as possible [17], and it also allows us to define our SALB-E problem as follows [28, 29, 30]:

$$M_{min} \leq M \leq M_{max} \tag{8}$$

and/or

$$C_{Tmin} \leq C_T \leq C_{Tmax} \tag{9}$$

Where (8) represents the number of workstations necessary and (9) represents the necessary cycle time.

According to the research conducted by Wei and Chao [31], C_{Tmax} should have a value that is greater than or equal to the maximum duration of all tasks and, at the same time, less than or equal to the total time required to complete them. If C_{Tmax} is greater than or equal to the sum of all task times, only one workstation is needed. The following are C_{Tmax} the constraints [32]:

According to (8), and since $T_{max} \leq C_{Tmin}$, therefore,

$$T_{max} \leq C_T \leq T_{sum} \tag{10}$$

If $T_{sum} \leq C_{Tmax}$, then $M = 1$, the line efficiency $E = \frac{T_{sum}}{1 \times T_{sum}} = 1$ and the balance loss is equal to zero.

After fixing the cycle time interval [28, 31], the optimum number of workstations is between M_{min} and M_{max}, where

$$M_{min} = \left\lceil \frac{T_{sum}}{C_{Tmax}} \right\rceil \tag{11}$$

and

$$M_{max} = \left\lceil \frac{T_{sum}}{T_{max}} \right\rceil \tag{12}$$

In addition, as mentioned in [29], it may be sufficient to set $M_{min} = \left\lceil \frac{T_{sum}}{C_{Tmax}} \right\rceil$ and $M_{max} = N$ if the interval $[C_{Tmin}, C_{Tmax}]$ is given. Conversely, if the interval $[M_{min}, M_{max}]$ is given, one can theoretically determine the lower and upper bounds of the cycle time as functions of M_{max} and M_{min}, respectively, where:

$$C_{Tmin} = f(M_{max}) = max\left\{T_{max}, \left\lceil \frac{T_{sum}}{M_{max}} \right\rceil\right\} \tag{13}$$

$$C_{Tmax} = g(M_{min}) = \begin{cases} max\left\{T_{max}, \left\lfloor \frac{2(T_{sum}-1)}{M_{min}} \right\rfloor\right\}, \text{if } M_{min} is \text{ even;} \\ max\left\{T_{max}, \left\lfloor \frac{2T_{sum}}{M_{min}+1} \right\rfloor\right\}, \text{if } M_{min} \text{is odd}, M_{min} \geq 3. \end{cases} \tag{14}$$

Hackman et al. [33] show that for a SALBP-2 instance with M workstations, C_{Tmin} and C_{Tmax} are two reasonable bounds on C_T. Note that these bounds are valid if the task execution times are integers and a SALBP-E can be defined as $C_{Tmax} - C_{Tmin} + 1$ SALBP-2 instances.

To obtain cycle time bounds for the SALBP-E, we can evaluate two functions of M. In these cases, we can find C_{Tmin} by evaluating $f(M_{max})$ and C_{Tmax} by evaluating $g(M_{min})$.

For a given interval $[M_{min}, M_{max}]$, we know that the simple C_{Tmin} results from the necessary feasibility condition $M \times C_T \geq T_{sum}$, i.e., $C_T \geq \frac{T_{sum}}{M_{max}}$ (according to Equation 1 [33]). Moreover, the indivisibility of the tasks implies that $C_T \geq T_{max}$.

Therefore, there are two cases, when $C_T = T_{max}\ or\ M = 1$ and when $C_T > T_{max}$ and $M \geq 2$, and in both cases $C_{Tmin} = max\left\{T_{max}, \left\lceil \frac{T_{sum}}{M_{max}} \right\rceil\right\} = f(M_{max})$, where f is a discontinuous and non-increasing function on the interval $[M_{min}, M_{max}]$, and $\lceil x \rceil$ is the smallest integer not less than x.

This concerns the lower bound. For the upper bound, if $C_T = T_{max}$ or $M = 1$, C_{Tmax} is evident.

Considering that $C_T > T_{max}$ and $M \geq 2$, we know that the statement [If $T_{sum} \leq C_{Tmax}$, then $M = 1$] is true. According to the law of contrapositive, [If $T_{sum} \leq C_{Tmax}$, then $M = 1$]$\Leftrightarrow$[If $M \geq 2$, then $C_{Tmax} < T_{sum}$].

According to the equation 2 [33],

$$T_{sum} > \begin{cases} \frac{M}{2} \times C_T, if\ M\ even \\ \frac{M-1}{2} \times C_T, if\ M\ odd, M \geq 3 \end{cases} \Leftrightarrow C_T < \begin{cases} \frac{2T_{sum}}{M}, if\ M\ even \\ \frac{2T_{sum}}{M-1}, if\ M\ odd, M \geq 3 \end{cases} \quad (*)$$

Let $\epsilon = T_{min} < C_T$, and $h(C_T)$ is the optimum value of SALBP-1 for a given cycle time.

So that $h(C_T - \epsilon) \geq M + 1$.

If $h(C_T - \epsilon) = M + 1$ or $h(C_T - \epsilon) \geq M + 2$, we find that $C_T \leq \frac{2T_{sum}}{M+1}$ in both cases by substituting $\begin{cases} h(C_T) \text{ for } M \\ C_T - \epsilon \text{ for } C_T \end{cases}$ in $(*)$ and $\epsilon \to 0$ ($\forall$ the parity of $h(C_T - \epsilon)$).

Since $T_k \in \mathbb{N}^*$, so if $T_i < T_j, \forall i, j \in \{1, \ldots, N\}$, then $T_i + 1 \leq T_j, \forall i, j \in \{1, \ldots, N\}$.

So $(*) \Rightarrow C_T \leq \begin{cases} \frac{2T_{sum}}{M} - 1, if\ M\ even \\ \frac{2(T_{sum}-1)}{M-1} - 1, if\ M\ odd, M \geq 3 \end{cases} (+)$.

Assuming that $\begin{cases} C_T > T_{max} \\ h(C_T - 1) \geq M + 1 \end{cases}$, so after substituting $C_T - 1$ for C_T in $(+)$, we obtain $C_T \leq \begin{cases} \frac{2(T_{sum}-1)}{M}, if\ M\ even \\ \frac{2T_{sum}}{M+1}, if\ M\ odd, M \geq 3 \end{cases}$.

Finally, we can conclude (14), where g is a discontinuous and non-increasing function on the interval $[M_{min}, M_{max}]$, and $\lfloor y \rfloor$ is the largest integer not greater than y.

REVIEW OF RELATED WORK

Although SALBP-1 and SALBP-2 are frequently studied, SALBP-E has received less attention due to its complexity, as reported in the literature (e.g., [34]). Therefore, this review will focus on addressing the existing research on SALBP-E in response to real-world demands. Scholl [35] defined SALBP-E as a specific type of SALBP where the objective is to minimise the product of M and C_T, subject to an upper bound on C_T. This translates into minimising the cost per unit of product in the context of SALBP.

The same objective was pursued by Scholl and Klein [36] and Plans and Corominas [37], e.g., Plans and Corominas [37] formulated a mixed integer linear programming (MILP) model that optimally solves SALBP-E, either exactly with a standard algorithm or heuristically with two implementations of the "fix-and-relax" meta-algorithm. The main aim of this research is to develop methods for obtaining good solutions to a reasonably large family of problems in a short time, but the model is only tested on seven instances, so the computational experiment is inadequate. Scholl and Klein [36] recommend using the U-Line Optimiser (ULINO) branch and bound method directly and in conjunction with search methods to solve UALBP-E. Traditionally, assembly lines are designed in a straight line, but the workstations in a U-line (see Fig. 3) offer several advantages over the standard configuration. The authors present a lower-bound search to determine the minimum line capacity, which involves the following steps: First, calculate the value of C_{Tmin} for each possible value of M and then arrange all (M, C_T) pairs in a list sorted by increasing values of L. Next, select and eliminate the first (M, C_T) pair from the list and solve the UALBP-1 instance with C_T. Stop at the optimal solution for UALBP-E if at least M stations are required. However, if $C_T + 1$ is still within the range of feasible cycle times, create the $(M, C_T + 1)$ combination and add it to the sorted list. If the sorted list is empty, then there are no feasible solutions to UALBP-E. If not, proceed to step 2.

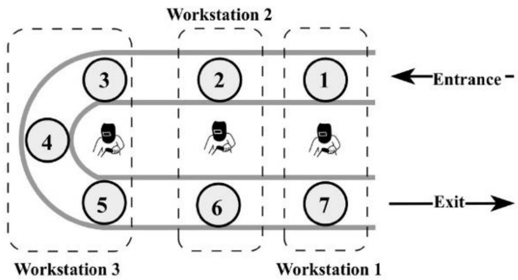

Fig. 3 Example of a U-shaped assembly line

The search algorithm produced 223 confirmed optimal solutions when applied to 256 instances of the dataset. For other cases, the relative deviations from the optimal solution are evaluated using $\frac{L^* - T_{sum}}{T_{sum}}$, where L^* is the best cost function value of the procedure. Further research is needed to determine more appropriate procedures for the UALB-E problem.

A common practice for solving SALBP-E is to repeatedly solve multiple instances of SALBP-1 or SALBP-2. Wei and Chao [31] proposed a model that combines both SALBP-1 and SALBP-2 models to maximise line efficiency and reduce total idle time. The proposed model uses the solution of several SALBP-2 formulations. To speed up the solution process, the model includes two additional variables, L_i and E_i, and recreates the SALBP-2 model. However, some errors were found in this study, which were later corrected in a study by García-Villoria and Pastor [38], e.g:

- Variables for the SALBP-1 model x_{kj} should exist for $j = 1, \dots, M_{max}$ (instead of $j = M_{min}, \dots, M_{max}$) [39];
- Constraints linking x_{kj} and variables y_j (the presence of a task j) are missing in the SALBP-1-i model;
- After giving a counterexample, they presented $Min\ L = C_T \sum_{j=1}^{M_{max}} y_j$ as the correct form of equation 17, cf. [31];
- They used a counterexample to show that $M_{max} = \left\lceil \frac{T_{sum}}{T_{max}} \right\rceil$, which is an important aspect of the method presented by Wei and Chao for solving SALBP-E, is incorrect.

Furthermore, the computational tests in [31] are limited to a small number of instances that can be solved optimally in a short time. They build on the work of Wei and Chao to find the best way to solve SALBP-E. Corominas et al. [40] provide a better MILP model and an iterative method that focuses on solving SALBP-2 at each iteration, which is better than what Wei and Chao [31] did. The exact solution of SALBP-E was of particular interest in their investigation. As SALBP-E is NP-hard, the researchers proposed various heuristics to limit the maximum processing time and return the optimal solution found for more important situations. Extensive tests were carried out, and the results suggest that the SALBP-E solution has improved. On the other hand, they found that a direct method, such as the proposed MILP model, outperforms any iterative method.

In 2006, Scholl and Becker [17] introduced a technique for addressing SALBP-E. Their method involved evaluating all possible instances of SALBP-F (M, C_T) with $M \in [M_{min}, M_{max}]$ and $C_T \in [C_{Tmin}, C_{Tmax}]$ to analyse which are viable. However, as there can be numerous (M, C_T) choices, this approach may be ineffective as the SALBP-F scenarios analysed are NP-complete. The researchers acknowledged that at the time there was no direct method for solving SALBP-E. It may be challenging to direct the search to fruitful zones within the solution space, as neither M nor C_T are fixed. In 2012 and 2013, Gurevsky et al. [41] and Zacharia and Nearchou [42] respectively addressed SALBP-E, focusing on stochastic task times. Zacharia and Nearchou's study can be seen as the first attempt to solve SALBP-E directly, without solving several SALBP-1 or SALBP-2 cases. They call this problem "fuzzy SALBP-E" because they assume that C_T and T_k are triangular fuzzy numbers. Their goal is to determine a feasible pair of task assignments $(M, \tilde{C}_T)$ that maximises the fuzzy line efficiency function $\tilde{E} = \frac{\tilde{T}_{sum}}{M \times \tilde{C}_T}$, where $\tilde{C}_T$ is the fuzzy cycle time and $\tilde{T}_{sum}$ is the total fuzzy time spent on all tasks. To solve this problem, they use a genetic algorithm-based metaheuristic. Tasan and Tunali's review [13] presents a comprehensive analysis of Genetic Algorithm-based solution techniques for almost all SALBP variants and shows that there is no existing GA-based metaheuristic for SALBP-E in the literature. Zacharia and Nearchou's approach may be the first attempt to solve SALBP-E directly with GAs. Numerical tests of their proposed method on publicly available benchmark problems show promising results. Gurevsky et al. [41] conducted a study on the stability of SALBP-E under different processing times. They investigated the stability of feasible and optimal solutions under potential variations in task processing time using stability analysis concepts such as minimum variation in processing time. The authors applied the principle of Pareto optimality [43] to strike a balance between minimising the cost function and maximising the associated stability radius. The study proposed two heuristics for detecting NDBs (non-dominated balances) in terms of Pareto optimality. On established benchmarks, these methods showed complementary behaviour and on average found a significant number of NDBs.

The MALBP-E was developed by Su et al. [32], who studied multi-objective ALB and considered mixed models with deterministic task times. Balancing a series of mixed models can be quite challenging, as there is limited research on this topic. They created a coloured timed Petri net model to represent the task-priority relation and formulated an optimisation problem as a mathematical programming model. They then proposed a two-stage heuristic procedure to solve the problem. First, they introduced a P-invariant algorithm (PA) based on the Petri net (PN) model to reduce M. Second, they presented a heuristic that combined the PA with the binary search algorithm (BSA) to further reduce C_T. The BSA-PA then minimises the C_T of the line in the second stage, depending on the output of the first stage. This method maximises E, and numerical tests have shown that it outperforms GA in terms of solution accuracy and processing efficiency for large-size problems.

In the context of heuristic methods, Kucukkoc and Zhang [44] develop an efficient swarm optimisation method for parallel two-sided assembly line systems (see Fig. 4), which are a new research topic in academia, although these lines have been used in industry for many years to produce large items such as cars, trucks, and buses. The main objective of this study [44] is to describe a PTALBP-E for the first time in the literature and to propose a novel ant colony optimisation-based solution to the problem.

The study presents a mathematical model for the formal description of the problem to simultaneously minimise two competing objectives, the cycle time and the number of workstations. The main objective of this paper is to define PTALBP-E and propose a new feasible solution strategy, which is an ACO algorithm whose parameters are optimised using a response surface methodology (RSM). The proposed technique and three additional well-known heuristics are used to solve a series of test problems originally developed for PTALBP-E, in order to minimise both the cycle time and the total number of workstations on parallel two-sided assembly lines.

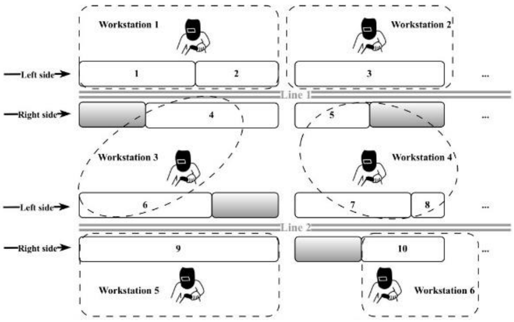

Fig. 4 Example of a parallel two-sided assembly line

The results of the computational analysis show that reducing C_T and M helps to increase the efficiency of the system. Comparing the results of the proposed algorithm with those of the other three known heuristics, the results are promising for the situations analysed by PTALBP-E.

As highlighted above, the development of efficient mathematical programming formulations is an essential research direction in SALBPs. However, type E problems lack a linear description. The MILP formulation specifically for SALBP-E was presented by Esmaeilbeigi et al. [29]. In addition, two augmentation strategies in the form of valid inequalities and supplementary variables are included to improve the proposed formulation. The secondary objectives of the problem are to reduce M, C_T and SI. In addition, the authors consider three linear techniques where SI is the smoothness index and is defined using I_t as a variable, and they assume a range of idle time values. The novel formulation can successfully solve problems with up to 50 tasks in a total time of less than 10 seconds. As a result of the mathematical model developed by Esmaeilbeigi et al. [29] for the E-type SALB problem, El Ahmadi et al. [45] developed a new model that can be adapted to the automotive industry to maximise efficiency by minimising idle time on the line. In addition, additional constraints are optimised as part of the model, taking into account priority and feasibility rules. This paper presents a computational experiment using the newly constructed model to balance an assembly line in an automotive manufacturing plant with five workstations. The results show that the proposed model increases the line efficiency by 15%, demonstrating that the proposed strategy is adaptive.

Since SALBP-E is an NP-hard category of combinatorial optimisation problems, most papers define metaheuristic algorithms rather than developing heuristic or exact methods. The difficulty of finding an optimal solution grows exponentially with the size of the case studied. Considerable research has been conducted to enhance the development of robust metaheuristic and hybrid metaheuristic algorithms, such as genetic algorithms (GAs), which are capable of tackling larger and more difficult problems that cannot be solved by traditional approaches. Researchers are increasingly interested in GAs because they provide an alternative to standard optimisation approaches by using direct randomised search to find optimal solutions in complicated landscapes. For example, in [46], a novel evolutionary method is proposed to solve multi-objective SALBPs. The proposed approach is called "Multiple-Assignment Genetic Algorithm" and it introduces a novel way of approaching work assignments. The objectives are to reduce the number of workstations, maximise the efficiency of the assembly line, and reduce the workload variation between workstations. The proposed MA-GA achieved the appropriate number of workstations in all test problems while improving line efficiency and workload variance between workstations in numerous test problems selected from the literature.

The vast majority of existing ALB-E research assumes that any assembly task can be distributed to any workstation. This assumption has led to an increased use of assembly line resources. Jusop and Ab Rashid [47] consider ALB-E with resource constraints (ALBE-RC), such as the machine, tool, and worker required to complete an assembly process for a given product. In this work, three objective functions are investigated: the number of workstations, the cycle time, and the amount of resources. To optimise the problem, an elitist non-dominated sorting genetic algorithm (NSGA-II) is developed in this study. Six benchmark problems were used to evaluate the optimisation method and the results were compared with those of the Multi-Objective Genetic Algorithm (MOGA) and the Hybrid Genetic Algorithm (HGA). According to the computational tests, NSGA-II can cover the search space and has a higher solution accuracy than other algorithms.

The studies by Belassiria et al. use a GA to solve SALBP-E. There are two objectives: to maximise E and to balance the workstations. According to Belassiria et al. [48], a hybrid GA (h-GA) is presented to improve the performance of the classical GA for large-scale SALBP-E with workstation constraints and zoning restrictions. Since the GA may lack the ability to effectively explore the solution space, the researchers aim to provide it with such ability by gradually hybridising known distribution rule heuristics with the GA. This will improve the quality of pure GA solutions, especially for large-scale problems. In another study by Belassiria et al. [49], they adapted two approaches from traditional GA to improve the search performance of GA. The first is a hybridisation of GA with a priority rule-based process that aims to seed the initial population with excellent solutions, and the second is GA with a localised evolution technique. Computational tests on 10 test sets taken from the literature showed that h-GA and neighbourhood GA (n-GA) outperformed classical GA on all test problems, confirming the ability of these strategies to improve the solution quality of classical GA.

Recently, Belkharroubi and Yahyaoui [30] proposed a genetic algorithm-based approach to deal with SALBP-E with the primary objective of maximising assembly line efficiency. This method involves determining the interval of the potential number of workstations $[M_{min}, M_{max}]$ and then searching for the best cycle time for M_{min}, M_{max}, and all values in between. In this situation, a SALBP-2 sequence must be solved and the best pair (M, C_T) that maximises E is selected as the best SALBP-E solution. The performance of the proposed GA is evaluated on three different problems and the results are compared with those achieved by the Hybrid Reactive Greedy Randomised Adaptive Search procedure [50]. Based on the results, the authors of this study show that the proposed GA is efficient in solving SALBP-2 as a subproblem of SALBP-E and can achieve the same results as the efficient Hybrid Reactive GRASP.

Following a comprehensive review of the available literature on SALBP-E, it is now time to summarise the results and findings of these studies in a more accessible format. We have gathered key information from a variety of articles to discuss the different techniques used to address this challenging problem. The table below (Table 1) is a useful reference tool, summarising the main characteristics, objectives, and methodology of the studies reviewed.

Table 1

A review of some selected papers related to SALBP-E published between 1999 and 2023

References	Line types	Type of objectives		Optimisation objectives							Types of production lines			Line configurations			Solution methods	Type of methods
		SO	MO	1	2	3	4	5	6	7	SL	MuL	MiL	S	U	T		
[37]	SALBP-E	x								x	x			x			Fix-and- relax	Metaheuristic
[36]	UALBP-E	x								x	x				x		Branch and Bound	Exact
[31]	SALBP-E	x								x	x			x			Excel	Software tool
[42]	Fuzzy SALBP-E	x						x			x			x			GA	Metaheuristic
[51]	SALBP-E	x							x		x			x			GA	Metaheuristic
[32]	MALBP-E		x	x	x								x	x			PN-based heuristic	Heuristic
[29]	SALBP-E		x	x	x	x	x				x			x			CPLEX	Software tool
[44]	PTALBP-E	x								x	x					x	ACO based approach	Metaheuristic
[40]	ALBE-RC	x								x	x			x			IP[1] and EIP[2] derived heuristics	Heuristic
[48]	SALBP-E		x			x	x				x			x			Hybrid GA	Metaheuristic
[49]	SALBP-E		x			x	x				x			x			h-GA[3] and n-GA[4]	Metaheuristic
[30]	SALBP-E	x								x	x			x			GA based approach	Metaheuristic
[45]	SALBP-E	x						x			x			x			Excel	Software tool

[1] Iterative Procedure. [2] Enhanced Iterative Procedure. [3] Hybrid genetic algorithm. [4] Neighbourhood genetic algorithm.

UALBP-E: the U-line ALBP of Type-E, MALBP-E: Mixed-model ALBP of Type-E, ALBE-RC: ALB-E with resource constraints, PTALBP-E: Parallel Two-sided ALBP of Type-E. 1: Minimise cycle time, 2: Minimise number of workstations, 3: Maximise line efficiency, 4: Maximise workload smoothness, 5: Minimise idle time, 6: Maximise production output, 7: Minimise line capacity.

SO: Single Objective, MO: Multi-Objective, SL: Single-model Line, MuL: Multi-model Line, MiL: Mixed-model Line, S: Straight lines, U: U-shaped lines, T: Two-sided lines, GA: Genetic Algorithms, PN: Petri Net, ACO: Ant Colony Optimisation.

METHODOLOGY

To ensure the high quality of the review process and its replicability, we followed the Preferred Reporting Items for Systematic Reviews (PRISMA) criteria [52]. The components of PRISMA 2020 are useful for mixed methods SLRs. However, the reporting requirements for the presentation and synthesis of qualitative data should also be used. PRISMA 2020 can be applied to original, modified, or continuously updated SLRs that use a standardised, peer-reviewed methodology that includes a flowchart and checklist of guidelines. According to the PRISMA standard, the research method included "search strategy ", "literature selection", and "qualitative analysis". The three stages of the process are discussed in detail below.

Search strategy

As part of our literature search procedure, a preliminary search of Scopus using the keyword "SALBP-E" was performed to identify keywords of interest. A full search was performed using the title, abstract, and keywords of the article, resulting in up to 10 relevant publications. Of these, 8 were journal articles, 1 was a conference paper, and 1 was an erratum. A review of these publications produced a list of interesting keywords, as shown in Table 2, which helped to identify further articles on this topic. Due to the lack of articles found in Scopus, the process had to be carried out in two other databases, ScienceDirect and Google Scholar.

Table 2
A list of keywords used in the search

Database	keywords
Scopus	“Simple assembly line balancing problem”, “SALBP”, “Assembly lines”, “Assembly line balancing”, “Assembly line balancing problems”, “Type E simple assembly line balancing problem”, “SALBP-E”, “Efficiency”, “Line efficiency”, “Manufacturing optimisation”.

LITERATURE SELECTION

According to the updated PRISMA diagram in Fig. 5, we retrieved 48 articles from different databases.

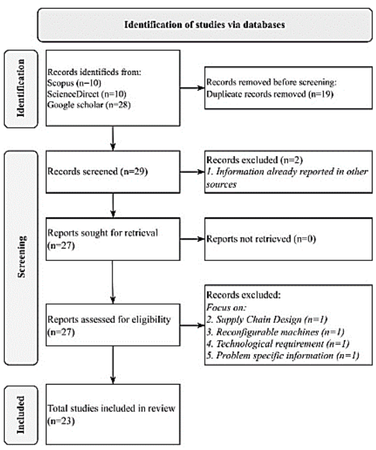

Fig. 5 Flowchart of the systematic review methodology according to PRISMA

Out of the initial 48 articles, 19 duplicates were removed. Two more articles were excluded after reviewing the abstracts of the remaining ones. The final selection of relevant studies was made after a thorough review of the remaining 27 articles, applying exclusion criteria. As a result, the "review of related work" subsection is based on 23 publications, which include only English-language journal articles, conference papers, and book chapters.

Qualitative analysis

The number of articles included in the analysis shows that there is a gap in the literature on SALBP-E. It should be noted that the database search was conducted in August 2023. Therefore, the results presented in this analysis only include articles published up to this date. Of the 23 articles included in the literature review subsection, 16 are journal articles, 5 are conference papers or proceedings, 1 is an erratum, and 1 is a book chapter.

RESULTS AND DISCUSSION

SALBP-E optimisation objectives

An objective function evaluates possible solutions and selects the best one. Depending on the objective function used to optimise assembly lines, there are two basic types of assembly lines: those optimised on the basis of a single objective function and those optimised simultaneously with multiple objectives. Most Type E assembly line balancing problems are characterised by single objective optimisation problems, and there is a limited amount of research that discusses other types of objectives that can be optimised, as shown in Fig. 6. Fig. 7 shows the frequency of SALBP-E objectives as collected in 13 referenced research articles.

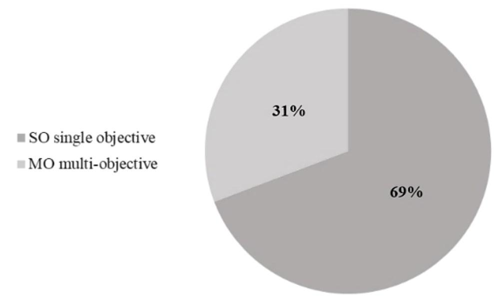

Fig. 6 Percentage of articles with single and multiple objectives

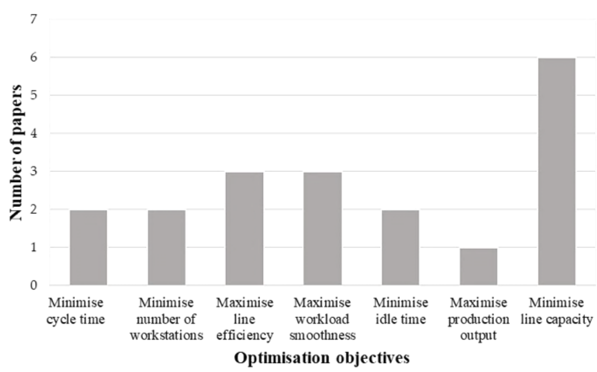

Fig. 7 Objectives of the SALBP-E frequently found in the studies cited

According to Table 1 and Fig. 7, SALBP-E is commonly used for the following main objectives:
- Minimise line capacity L;
- Maximise line efficiency E;
- To maximise workload smoothness, the aim is to reduce the differences between the workloads of different workstations, i.e., minimise $SI = \sqrt{\sum_{j=1}^{M}[C_T - t(M_j)]^2}$, where $t(M_j)$ is the working time of workstation j;
- Minimise idle time I_t;
- Minimise the number of workstations M;
- Minimise cycle time C_T.

The interest of researchers in these objectives can be explained as follows:
- It is sufficient to minimise the capacity of the line in order to maximise the efficiency of the line as the main objective of a SALBP-E problem.

$$E = \frac{T_{sum}}{C_T \times M} = \frac{T_{sum}}{L} \tag{15}$$

- To maximise line efficiency E, it is essential to reduce idle time at workstations. Higher efficiency results in a smaller difference between the sum of task times T_{sum} and line capacity L.

$$I_t = (C_T \times M) - T_{sum} = L - T_{sum} = L(1 - E) \tag{16}$$

$$I_t = 0 \Leftrightarrow L(1 - E) = 0 \Leftrightarrow L = 0 \; or \; E = 1 \tag{17}$$

- We can achieve workload smoothness by reducing I_t at all workstations by reducing SI. With less variance in processing time and workload at the workstations, each workstation can have a workload close to the cycle time of the line.
- Minimising line capacity, maximising line efficiency, maximising workload smoothness, and minimising idle time all depend implicitly on two key variables: the number of workstations and the cycle time. Therefore, to achieve any one of these objectives, it is sufficient to minimise the number of workstations, minimise the cycle time, or minimise both at the same time.

Types and configurations of production lines

Based on product characteristics, SALBP-E can be categorised as a single-model, mixed-model, or multi-model line. The single-model line produces a large quantity of only one product type and represents both task time and priority constraints as a single-priority graph. However, personalised products require separate assembly lines for different product types, which is a costly investment [53]. To solve this problem, a mixed-model assembly line has been developed that can produce a variety of products on a single assembly line.

When assembling different products on a mixed-model assembly line, the priority relationships are established by combining the priority diagrams for each product model. Table 3 shows that there is a strong tendency for researchers to study the SALBP-E using single model lines.

This is due to the fact that in mixed model lines, the design requirements for each model may change, resulting in different assembly times for each model at each workstation, leading to uneven workloads and reduced productivity.

Table 3

Number of papers used different types of production lines

Types of production lines	Number of papers
Single-model Line	⬆ **12**
Mixed-model Line	⬇ **1**
Multi-model Line	⬇ **0**

According to the literature, there are two main challenges associated with mixed-model assembly lines. The first is the allocation of work to workstations, while the second, known as model sequencing (MS), is the determination of the order in which different product models are built at the workstations. For more details, see [54].

Based on our survey of the 13 articles, it appears that the third type of assembly line, known as a multi-model line, has not received much attention. This type produces several products in a single line. Instead of combining different models, each model is produced in batches. Once a batch of one model has been produced, the station layout is changed to meet the requirements of the next batch of products [55]. However, producing multiple batches of models with different batch sizes can be challenging. When batch sizes are large, each batch of product is balanced independently.

Depending on the type of model to be assembled and its characteristics, SALBP-E can adopt different flow line procedures in terms of line shape, the most commonly used being straight, two-sided, and U-shaped lines. Fig. 8 confirms that straight lines produce the most research results (84%).

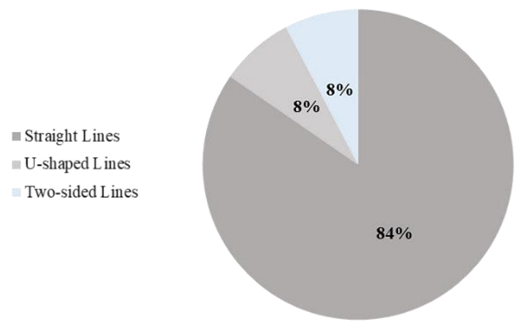

Fig. 8 Distributing articles according to line configuration

U-shaped and double-sided lines produce the fewest research results (8% each).

A straight-line assembly line is a series of workstations used to build semi-finished products. Each workstation performs a specific task, and components move from one workstation to the next until the assembly process is complete. The decision to use straight lines for SALBP-E rather than U-shaped or double-sided lines depends on a number of factors, including the simplicity, flexibility, and efficiency of the system. However, in some cases, U-shaped or two-sided layouts may be preferred to improve line efficiency [15].

The workstations in a U-shaped line are not arranged in a straight line but in a U-shape, so that workers in so-called crossover workstations can perform activities at two workstations in the same cycle. One of the many benefits of a U-shaped line layout is that idle workstations can be used to perform additional tasks, helping to improve workstation utilisation. A U-shaped line not only helps to reduce the number of workstations but also helps to improve line efficiency by utilising workstations that may have idle time during the assembly process [56].

Unlike straight and U-shaped assembly lines, two-sided lines have two sides. Some tasks are assigned to one side, while others can be passed to both sides. The latter type of task creates a difficult constraint that requires formal task planning within the workstation. Two-sided assembly lines have the advantage that many tasks can be performed simultaneously, provided that the priority constraints are not broken [7]. However, these lines are limited by several additional constraints in addition to the task priority constraints. Furthermore, when two-sided assembly lines are grouped in a serial layout, the workload at a workstation is more than just the sum of the task times at that workstation. Each workstation has a scheduling problem that can cause variations in workload [15].

Optimisation methods

The selection of an appropriate optimisation method is critical to supporting the types and configurations of production lines. Based on our research, researchers have applied a variety of methods to the SALB-E problem, such as exact, heuristic, and metaheuristic methods.

Fig. 9 shows the number of articles published in the last twenty-four years using different soft computing methods to optimise SALBP-E.

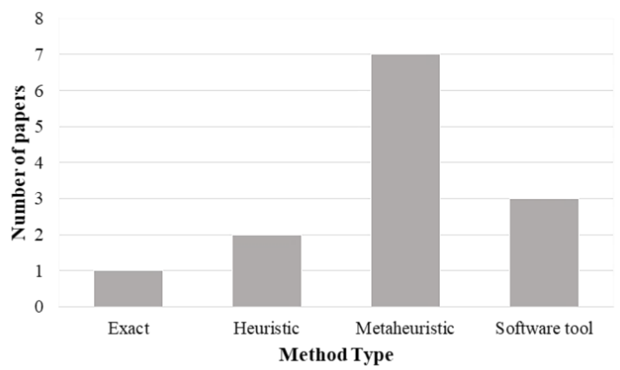

Fig. 9 Solution techniques

According to the graph, the most popular optimisation approaches are metaheuristic methods, and it can be seen from Fig. 10 that GA is the most frequently used metaheuristic algorithm to solve SALBP-E.

Genetic algorithms (GAs) are popular population-based metaheuristics inspired by biological evolution that use many candidate solutions to find optimal or near-optimal solutions to large-scale optimisation problems [57]. This metaheuristic method starts by generating an initial population of chromosomes representing the initial solutions.

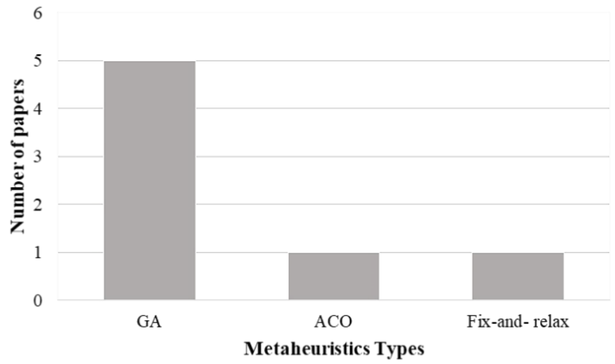

Fig. 10 The most commonly used metaheuristic algorithm for solving SALBP-E

A fitness function is used to evaluate the fitness of each solution to determine the termination criterion. The best individuals are then selected for crossover and mutation processes to create super-fit offspring chromosomes. Individuals produced by the genetic operators are also evaluated, and the process is repeated for a number of generations [58]. Fig. 11 shows a flowchart of the process.

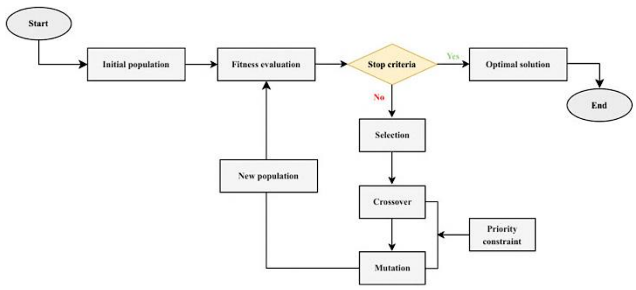

Fig. 11 Genetic algorithm flowchart

GA has been proven in the literature to be a successful method for solving challenging problems. This is due to its ability to move from one set of solutions to another and to adapt to specific problem characteristics [58]. While GAs cannot guarantee to find an optimal solution, they will usually identify good solutions with acceptable computational complexity. GA is extremely parallel and adaptive, with two major advantages. First, it searches a population rather than a single point, which reduces the risk of getting stuck in a local optimum. Second, it is capable of solving any type of objective function or constraint [59].

CONCLUSION

This systematic literature review aims to examine SALBP-E. SALBP is an important research topic, but SALBP-E has received less attention due to its complexity. According to our research, the majority of SALBP-E are single-objective optimisation problems, and multi-objective problems are rare. Furthermore, researchers often analyse SALBP-E using single-model lines, ignoring the value of mixed-model lines and multi-model lines. Further research in these neglected areas is needed to better understand the problem and to propose feasible solutions that fit real-world situations.

In addition, there are several layout characteristics to consider when designing assembly lines, such as straight lines, U-shaped lines, and two-sided lines. However, research has shown that the majority of studies have focused on straight-line configurations, with U-shaped and two-sided lines receiving much less attention. Research into these less studied versions is required to develop a thorough and balanced understanding of SALBP-E.

In recent years, soft computing techniques have been widely applied to improve SALBP-E. Among many optimisation approaches, metaheuristic methods have emerged as the most popular, with GA being the most widely used to tackle SALBP-E. This suggests that researchers should further explore the potential of metaheuristics to find innovative solutions to this challenging problem.

There are many interesting avenues for future SALBP-E researchers:

– The SALBP-E may need to analyse multiple objective functions rather than a single objective function to represent real operational requirements.
– In order to effectively meet the diverse needs of customers in today's rapidly evolving market with short product life cycles, multi- and mixed-model assembly lines are considered to be more efficient. It is therefore essential to investigate the impact of mixed and multi-model assembly lines in the context of SALBP-E. This will lead to a more comprehensive understanding of how this issue operates in real-world environments covering a wide range of product models.
– Further studies are required to investigate U-shaped and two-sided lines in the SALBP-E, as they are underrepresented in the current literature. Investigating their application in different industrial situations may provide useful insights.
– According to the major study, metaheuristics, in particular genetic algorithms, are the most successful methods for solving the SALB-E problem. To solve this problem, it is crucial to focus on improving and refining genetic algorithms, as well as investigating new approaches within the metaheuristic framework.

ACKNOWLEDGMENTS

"*The authors express their gratitude to the Moroccan Minister of Higher Education, Scientific Research and Innovation, for funding the MENFPESRS/DESRS bilateral R&D project.*"

REFERENCES

[1] I. Jebbor, Z. Benmamoun, and H. Hachimi, "Optimizing Manufacturing Cycles to Improve Production: Application in the Traditional Shipyard Industry," *Processes*, vol. 11, no. 11, p. 3136, Nov. 2023, doi: 10.3390/pr11113136.

[2] N. M. Aripin, G. Nawanir, F. Mahmud, M. A. Fauzi, S. Hussain, and K. L. Lee, "Systematic Literature Review: Theory Perspective in Lean Manufacturing Performance," *Management Systems in Production Engineering*, vol. 31, no. 2, pp. 230-241, Jun. 2023, doi: 10.2478/mspe-2023-0025.

[3] P. Chutima and A. Khotsaenlee, "Multi-objective parallel adjacent U-shaped assembly line balancing collaborated by robots and normal and disabled workers," *Computers & Operations Research*, vol. 143, p. 105775, Jul. 2022, doi: 10.1016/j.cor.2022.105775.

[4] A. Nourmohammadi, M. Fathi, and A.H.C. Ng, "Balancing and scheduling assembly lines with human-robot collaboration tasks," *Computers & Operations Research*, vol. 140, p. 105674, Apr. 2022, doi: 10.1016/j.cor.2021.105674.

[5] D.M. Popa, "Engineering Risk of Series Production in Automotive Industry," *Management Systems in Production Engineering*, vol. 27, no. 1, pp. 5-11, Mar. 2019, doi: 10.1515/mspe-2019-0001.

[6] Y. Bouazza, A. Lajjam, and B. Dkhissi, "The Impact of Lean Manufacturing on Enviromental Performance in Moroccan Automotive Industry," *Management Systems in Production Engineering*, vol. 29, no. 3, pp. 184-192, Sep. 2021, doi: 10.2478/mspe-2021-0023.

[7] H. Aguilar, A. García-Villoria, and R. Pastor, "A survey of the parallel assembly lines balancing problem," *Computers & Operations Research*, vol. 124, p. 105061, Dec. 2020, doi: 10.1016/j.cor.2020.105061.

[8] İ. Baybars, "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem," *Management Science*, vol. 32, no. 8, pp. 909-932, Aug. 1986, doi: 10.1287/mnsc.32.8.909.

[9] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," *European Journal of Operational Research*, vol. 183, no. 2, pp. 674-693, Dec. 2007, doi: 10.1016/j.ejor.2006.10.010.

[10] N. Boysen, M. Fliedner, and A. Scholl, "Assembly line balancing: Which model to use when?," *International Journal of Production Economics*, vol. 111, no. 2, pp. 509-528, Feb. 2008, doi: 10.1016/j.ijpe.2007.02.026.

[11] M.E. Salveson, "The Assembly-Line Balancing Problem," *Transactions of the American Society of Mechanical Engineers*, vol. 77, no. 6, pp. 939-947, 1955, doi: 10.1115/1.4014559.

[12] K.R. Baker, *Introduction to sequencing and scheduling*. John Wiley & Sons, 1974.

[13] S.O. Tasan and S. Tunali, "A review of the current applications of genetic algorithms in assembly line balancing," *J Intell Manuf*, vol. 19, no. 1, pp. 49-69, Feb. 2008, doi: 10.1007/s10845-007-0045-5.

[14] A. Scholl and A. Scholl, *Balancing and sequencing of assembly lines*. Springer, 1999.

[15] U. Saif, Z. Guan, B. Wang, J. Mirza, and S. Huang, "A survey on assembly lines and its types," *Front. Mech. Eng.*, vol. 9, no. 2, pp. 95-105, Jun. 2014, doi: 10.1007/s11465-014-0302-1.

[16] B. Rekiek, A. Dolgui, A. Delchambre, and A. Bratcu, "State of art of optimization methods for assembly line design," *Annual Reviews in Control*, vol. 26, no. 2, pp. 163–174, Jan. 2002, doi: 10.1016/S1367-5788(02)00027-5.

[17] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *European Journal of Operational Research*, vol. 168, no. 3, pp. 666-693, Feb. 2006, doi: 10.1016/j.ejor.2004.07.022.

[18] A. Scholl and S. Voß, "Simple assembly line balancing?Heuristic approaches," *J Heuristics*, vol. 2, no. 3, pp. 217-244, 1997, doi: 10.1007/BF00127358.

[19] N. Boysen, P. Schulze, and A. Scholl, "Assembly line balancing: What happened in the last fifteen years?," *European Journal of Operational Research*, vol. 301, no. 3, pp. 797-814, Sep. 2022, doi: 10.1016/j.ejor.2021.11.043.

[20] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *European Journal of Operational Research*, vol. 168, no. 3, pp. 694-715, Feb. 2006, doi: 10.1016/j.ejor.2004.07.023.

[21] N. Kumar and D. Mahto, "Assembly Line Balancing: A Review of Developments andTrends in Approach to Industrial Application," p. 23, 2013.

[22] E.C. Sewell and S.H. Jacobson, "A Branch, Bound, and Remember Algorithm for the Simple Assembly Line Balancing Problem," *INFORMS Journal on Computing*, vol. 24, no. 3, pp. 433-442, Aug. 2012, doi: 10.1287/ijoc.1110.0462.

[23] O. Kilincci and G.M. Bayhan, "A Petri net approach for simple assembly line balancing problems," *Int J Adv Manuf Technol*, vol. 30, no. 11-12, pp. 1165-1173, Oct. 2006, doi: 10.1007/s00170-005-0154-2.

[24] C. Blum, "Beam-ACO for Simple Assembly Line Balancing," *INFORMS Journal on Computing*, vol. 20, no. 4, pp. 618-627, Nov. 2008, doi: 10.1287/ijoc.1080.0271.

[25] E. Álvarez-Miranda, J. Pereira, H. Torrez-Meruvia, and M. Vilà, "A Hybrid Genetic Algorithm for the Simple Assembly Line Balancing Problem with a Fixed Number of Workstations," *Mathematics*, vol. 9, no. 17, p. 2157, Sep. 2021, doi: 10.3390/math9172157.

[26] C. Blum, "Iterative Beam Search for Simple Assembly Line Balancing with a Fixed Number of Work Stations." arXiv, Dec. 15, 2010. doi: https://doi.org/10.48550/arXiv.1012.3273.

[27] A. Mozdgir, I. Mahdavi, I.S. Badeleh, and M. Solimanpur, "Using the Taguchi method to optimize the differential evolution algorithm parameters for minimizing the workload smoothness index in simple assembly line balancing," *Mathematical and Computer Modelling*, vol. 57, no. 1-2, pp. 137-151, Jan. 2013, doi: 10.1016/j.mcm.2011.06.056.

[28] S.V. Ravelo, "Approximation algorithms for simple assembly line balancing problems," *J Comb Optim*, vol. 43, no. 2, pp. 432-443, Mar. 2022, doi: 10.1007/s10878-021-00778-2.

[29] R. Esmaeilbeigi, B. Naderi, and P. Charkhgard, "The type E simple assembly line balancing problem: A mixed integer linear programming formulation," *Computers & Operations Research*, vol. 64, pp. 168-177, Dec. 2015, doi: 10.1016/j.cor.2015.05.017.

[30] L. Belkharroubi and K. Yahyaoui, "Maximization of the assembly line efficiency using an approach based on Genetic Algorithm," in *2022 2nd International Conference on Innovative Research in Applied Science, Engineering and Technology (IRASET)*, Meknes, Morocco: IEEE, Mar. 2022, pp. 1-6. doi: 10.1109/IRASET52964.2022.9737934.

[31] N.-C. Wei and I.-M. Chao, "A solution procedure for type E simple assembly line balancing problem," *Computers & Industrial Engineering*, vol. 61, no. 3, pp. 824-830, Oct. 2011, doi: 10.1016/j.cie.2011.05.015.

[32] P. Su, N. Wu, and Z. Yu, "A Petri net-based heuristic for mixed-model assembly line balancing problem of Type-E," *International Journal of Production Research*, vol. 52, no. 5, pp. 1542–1556, Mar. 2014, doi: 10.1080/00207543.2013.849010.

[33] S. T. Hackman, M. J. Magazine, and T. S. Wee, "Fast, Effective Algorithms for Simple Assembly Line Balancing Problems," *Operations Research*, vol. 37, no. 6, pp. 916-924, Dec. 1989, doi: 10.1287/opre.37.6.916.

[34] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solutionapproaches," *International Journal of Production Economics*, vol. 142, no. 2, pp. 259-277, Apr. 2013, doi: 10.1016/j.ijpe.2012.10.020.

[35] A. Scholl, "Balancing and Sequencing of Assembly Lines. Vol. 168," *Heidelberg: Physica-Verlag HD*, pp. 694-715, 1995.

[36] A. Scholl and R. Klein, "ULINO: Optimally balancing U-shaped JIT assembly lines," *International Journal of Production Research*, vol. 37, no. 4, pp. 721-736, Mar. 1999, doi: 10.1080/002075499191481.

[37] J. Plans and A. Corominas, "Modelling and solving the SALB-E problem," in *Proceedings of the 1999 IEEE International Symposium on Assembly and Task Planning (ISATP'99) (Cat. No.99TH8470)*, Porto, Portugal: IEEE, 1999, pp. 356-360. doi: 10.1109/ISATP.1999.782984.

[38] A. García-Villoria and R. Pastor, "Erratum to 'A solution procedure for type E simple assembly line balancing problem,'" *Computers & Industrial Engineering*, vol. 66, no. 1, pp. 201-202, Sep. 2013, doi: 10.1016/j.cie.2013.05.015.

[39] L. Capacho Betancourt, "ASALBP: the Alternative Subgraphs Assembly Line Balancing Problem. Formalization and Resolution Procedures," Ph.D. Thesis, Universitat Politècnica de Catalunya, 2008. Accessed: Oct. 18, 2022. [Online]. Available: http://www.tdx.cat/handle/10803/5950

[40] A. Corominas, A. Garcıa-Villoria, and R. Pastor, "Improving the resolution of the simple assembly line balancing problem type E," p. 16, 2016.

[41] E. Gurevsky, O. Battaïa, and A. Dolgui, "Balancing of simple assembly lines under variations of task processing times," *Ann Oper Res*, vol. 201, no. 1, pp. 265-286, Dec. 2012, doi: 10.1007/s10479-012-1203-5.

[42] P.Th. Zacharia and A.C. Nearchou, "A meta-heuristic algorithm for the fuzzy assembly line balancing type-E problem," *Computers & Operations Research*, vol. 40, no. 12, pp. 3033-3044, Dec. 2013, doi: 10.1016/j.cor.2013.07.012.

[43] X. Gandibleux and M. Ehrgott, "1984-2004 – 20 Years of Multiobjective Metaheuristics. But What About the Solution of Combinatorial Problems with Multiple Objectives?," in *Evolutionary Multi-Criterion Optimization*, vol. 3410, C. A. Coello Coello, A. Hernández Aguirre, and E. Zitzler, Eds., in Lecture Notes in Computer Science, vol. 3410. , Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 33-46. doi: 10.1007/978-3-540-31880-4_3.

[44] I. Kucukkoc and D.Z. Zhang, "Type-E parallel two-sided assembly line balancing problem: Mathematical model and ant colony optimisation based approach with optimised parameters," *Computers & Industrial Engineering*, vol. 84, pp. 56-69, Jun. 2015, doi: 10.1016/j.cie.2014.12.037.

[45] S.E.A. El Ahmadi, L. El Abbadi, and S. Elrhanimi, "Efficiency improvement of automotive assembly lines using simple assembly line balancing problem Type-E.," *LogForum*, vol. 19, no. 2, 2023.

[46] T. Al-Hawari, M. Ali, O. Al-Araidah, and A. Mumani, "Development of a genetic algorithm for multi-objective assembly line balancing using multiple assignment approach," *Int J Adv Manuf Technol*, vol. 77, no. 5-8, pp. 1419-1432, Mar. 2015, doi: 10.1007/s00170-014-6545-5.

[47] M. Jusop and M.F.F. Ab Rashid, "Optimisation of Assembly Line Balancing Type-E with Resource Constraints Using NSGA-II," *KEM*, vol. 701, pp. 195-199, Jul. 2016, doi: 10.4028/www.scientific.net/KEM.701.195.

[48] I. Belassiria, M. Mazouzi, S. El Fezazi, and Z. El Maskaoui, "An efficient approach for workload balancing of assembly line systems with assignment restrictions," in *2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, Rabat, Morocco: IEEE, Apr. 2017, pp. 7-12. doi: 10.1109/LOGISTIQUA.2017.7962865.

[49] I. Belassiria, M. Mazouzi, S. ELfezazi, and Z. ELMaskaoui, "A performance comparison between hybrid and neighbourhood genetic algorithm applied to assembly line balancing problem," in *2018 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)*, Tangier:

IEEE, Apr. 2018, pp. 25-31. doi: 10.1109/LOGISTIQUA.2018.8428279.

[50] L. Belkharroubi and K. Yahyaoui, "A Hybrid Grasp-genetic Algorithm for Mixed-model Assembly Line Balancing Problem Type 2," *IJC*, pp. 424-432, Sep. 2021, doi: 10.47839/ijc.20.3.2289.

[51] N. Azmi, I.Y. Azzubaidi, and S. Batubara, "Solving Assembly Line Balancing Problem Using Genetic Algorithm Technique With Partitioned Chromosome," *Dipetik Juni*, vol. 19, p. 2019, 2013.

[52] M.J. Page *et al.*, "The PRISMA 2020 statement: An updated guideline for reporting systematic reviews," *International Journal of Surgery*, vol. 88, p. 105906, Apr. 2021, doi: 10.1016/j.ijsu.2021.105906.

[53] J.I. Van Zante-de Fokkert and T.G. De Kok, "The mixed and multi model line balancing problem: a comparison," *European Journal of Operational Research*, vol. 100, no. 3, pp. 399-412, Aug. 1997, doi: 10.1016/S0377-2217(96)00162-2.

[54] E.M. Dar-El and A. Nadivi, "A mixed-model sequencing application," *International Journal of Production Research*, vol. 19, no. 1, pp. 69-84, Jan. 1981, doi: 10.1080/00207548108956630.

[55] M. Eghtesadifard, M. Khalifeh, and M. Khorram, "A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990-2017," *Computers & Industrial Engineering*, vol. 139, p. 106182, Jan. 2020, doi: 10.1016/j.cie.2019.106182.

[56] I. Kucukkoc and D.Z. Zhang, "Balancing of parallel U-shaped assembly lines," *Computers & Operations Research*, vol. 64, pp. 233-244, Dec. 2015, doi: 10.1016/j.cor.2015.05.014.

[57] S. Katoch, S.S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimed Tools Appl*, vol. 80, no. 5, pp. 8091-8126, Feb. 2021, doi: 10.1007/s11042-020-10139-6.

[58] Sabuncuglu, Erel, and Tanyer, "Assembly line balancing using genetic algorithms," 2000, doi: https://doi.org/10.1023/A:1008923410076.

[59] A.S. Simaria and P.M. Vilarinho, "A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II," *Computers & Industrial Engineering*, vol. 47, no. 4, pp. 391-407, Dec. 2004, doi: 10.1016/j.cie.2004.09.001.

Sana El Machouti (corresponding author)
ORCID ID: 0009-0000-2588-6117
LASTIMI Laboratory
Mohammedia School of Engineers (EMI),
Mohammed V University in Rabat
Center of Excellence in Logistics (CELOG)
School of Textile and Clothing (ESITH)
e-mail: sana_elmachouti@um5.ac.ma,
sana.elmachouti@gmail.com

Mustapha Hlyal
ORCID ID: 0000-0003-1814-5438
Center of Excellence in Logistics (CELOG)
School of Textile and Clothing (ESITH)
Casablanca 7731, Morocco
e-mail: hlyal@esith.ac.ma

Amel Babay
ORCID ID: 0000-0002-5942-8146
Textile Engineering Laboratory, ISET Ksar Hellal
Monastir University, Ksar Hellal 5070, Tunisia
e-mail: babay_amel@yahoo.fr

Jamila El Alami
ORCID ID: 0000-0003-4658-4550
LASTIMI Laboratory
High School of Technology (EST)
Mohammed V University in Rabat
Sale 227, Morocco
e-mail: jamila.elalami@um5.ac.ma