

Contents lists available at ScienceDirect

Operations Research Perspectives



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

Operations Research Perspectives

New Matrix Methodology for Algorithmic Transparency in Assembly Line Balancing Using a Genetic Algorithm

Juan Ignacio Anel^{a,*}, Pau Català^{b,1}, Moisès Serra^{c,2}, Bruno Domenech^{d,3}

^a Department of Engineering, Universitat de Vic – Universitat Central de Catalunya (UVic-UCC), Vic, Spain, 08500

^b Department of Mechanical Engineering, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 08028

^c Department of Engineering, Universitat de Vic – Universitat Central de Catalunya (UVic-UCC), Vic, Spain, 08500

^d Department of Management, Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, 08028

ARTICLE INFO

Keywords: Manufacturing Mixed-Model Assembly Line Balancing Algorithmic Transparency new matrix methodology Balanced optimization tool for SME

ABSTRACT

This article focuses on the Mixed-Model Assembly Line Balancing single-target problem of type 2 with singlesided linear assembly line configurations, which is common in the industrial environment of small and medium-sized enterprises (SMEs). The main objective is to achieve Algorithmic Transparency (AT) when using Genetic Algorithms for the resolution of balancing operation times. This is done by means of a new matrix methodology that requires working with product functionalities instead of product references.

The achieved AT makes it easier for process engineers to interpret the obtained solutions using Genetic Algorithms and the factors that influence decisions made by algorithms, thereby helping in the later decisionmaking process. Additionally, through the proposed new matrix methodology, the computational cost is reduced with respect to the stand-alone use of Genetic Algorithms.

The AT produced using the new matrix methodology is validated through its application in an industry-based paradigmatic example.

1. Introduction

Mixed-Model Assembly Line Balancing (MMALB) aims for the simultaneous assembly of a set of functionally-related products on a single assembly line. They are common in the industrial environment of small and medium-sized enterprises (SMEs). The batch atomization caused by the just-in-time strategies and the increasing extension of the product portfolio means that when considering new assembly lines, they are defined according to the One-Piece-Flow principles of the Lean philosophy with the aim of trying to alleviate the extra costs involved in managing small batches of production.

To cope with this difficult industrial situation, process engineers must achieve a balance of the operation times assigned to each workstation, the required manufacturing cadence and the product portfolio required by customers.

AT is the principle that factors with influence on the decisions made by algorithms should be visible or transparent to the people who use, regulate and are affected by the systems that employ these algorithms. In the case of balancing assembly lines, AT would ease the interpretation of the mathematical model generated for process engineers and, with the obtained results, make decisions regarding the planning of a new assembly line (e.g. be clear about the effect of assigning an additional assembly task in the future to one or another workstation and consider the alternatives according to the specific mathematical model for its processes).

In this paper, an improvement in the use of the Genetic Algorithm (GA) is provided by means of introducing a new matrix methodology when prepossessing the raw data of the problem that will confer -once the GA has been applied- superior AT to the solutions with respect to the specialized literature. This contribution represents an improvement in the implementation of the GA for SMEs dealing with optimisation when balancing assembly lines, since most of the solutions to date, although feasible, are difficult to interpret for process engineers, thus hindering subsequent work in the decision-making process. This methodology is

https://doi.org/10.1016/j.orp.2022.100223

Received 25 August 2021; Received in revised form 14 December 2021; Accepted 11 January 2022 Available online 13 January 2022

2214-7160/© 2022 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-ad/4.0/).

^{*} Corresponding author: Tell: +34938815519

E-mail addresses: juan.anel@uvic.cat (J.I. Anel), pau.catala@upc.edu (P. Català), moises.serra@uvic.cat (M. Serra), bruno.domenech@upc.edu (B. Domenech).

¹ Tell: +34 934016577

² Tell: +34938815519

³ Tell: +34934016200



Fig. 1. Classification of ALB problem

applied in an industry-based paradigmatic example.

In addition, when using the proposed new matrix methodology, the computational cost is reduced when compared to the stand-alone use of the GA for systems with an extensive product portfolio defined by product references rather than product functionalities as proposed in the paper. In general, in a company's product portfolio there are more product references than product functionalities. The reduction in computational cost allows implementing the methodology even in spreadsheets, a software widely used by SMEs, thus improving the accessibility for process engineers.

The initial hypothesis is that, in an MMALB problem, all product references can be balanced simultaneously by minimizing the standard deviation of the aggregate operation times assigned to each of the workstations for each group of product functionalities. The minimization of this defined standard deviation is used as the objective function for the GA.

The structure of the article is as follows. Section 2 reviews the specialized literature, Section 3 explains the new matrix methodology approach, along with the mathematical formulation. In Section 4, the methodology is applied in an industry-based paradigmatic example. Finally, in Section 5, the main conclusions are presented.

2. Literature review

The classification of the Assembly Line Balancing problems (ALB) according to Kamal and Martinez Lastr [9] is presented in Fig. 1:

According to Fig. 1, ALB problems based on the objective function are classified as: type-F (feasibility), obtaining a feasible solution for a given number of workstations and a given cycle time; type-1, minimization of the number of workstations for a given cycle time; type-2, minimization of the cycle time for a given number of workstations; type-E (effectiveness), minimization of both the number of workstations and the cycle time; finally, types-3, 4 and 5 correspond to maximization of the workload smoothness, maximization of the work relatedness and an aggregate of these two objectives that corresponds to types-3 and 4, respectively. Based on the problem structure, considering Scholl [1999] and Becker et al. [2006], ALB problems are classified as: SMALB, for single model ALB problems, where only one product is assembled; MuMALB, for multi-model ALB problems, where multiple products are assembled in batches on the line; and MMALB, for mixed-model ALB problems, where various models of a generic product are produced in the assembly line in a mixed situation.

Considering the literature review dealing with MMALB problems by Sivasankaran et al. [21], the following techniques have been used: Mathematical model, simulated annealing, tabu search, ant colony optimization (ACO) algorithms and GA. The literature of additional techniques is also reviewed in this paper, specifically branch and bound and bee colony algorithms.

Considering AT, the only identified study that focused on the results analysis is by Wei et al. [23], who studied binary linear programming with Excel VBA, proposing a better understanding of management practice thanks to the results being presented through the visual interface of a spreadsheet. However, it only improves the interpretation of the results, without becoming an exhaustive modelling that allows the factors that influence the solution (and how they do so) to be understood. Moreover, according to Fig. 1, the case studied is classified as a SALB problem, which is not applicable to MMALB problems as it is the case of the paper.

The specific literature analysed that addresses MMALB problems, essentially revolves around mathematical optimization or adding functionalities to algorithms, such as simultaneous objectives, uncertainty management or variability.

Pinarbasi et al. [16] developed an efficient resolution with mathematical models using constraint programming model for type-2 ALBP with several assignment restrictions. Michels et al. [14] also considered a resolution for multi-manned assembly lines, using a Mixed-Integer Linear Programming model to solve large and real-life instances optimally.

Nazari et al. [15] consider sequence dependant setup times between operations, finding that simulated annealing is more effective than a mixed integer programming model.

Abdeljaouad et al. [1] generalized tabu search algorithms solution approach to the multi-model case (MALBP) starting from the cycle time minimization used in single-model lines (SALBP) and showing a high performance.

Kucukkoc et al. [12] developed a resolution with ACO algorithms for mixed-model parallel two-sided assembly line problems. The objective values are not as good as desired although the line is more flexible to demand changes; any new model sequences could be launched with no need for balance change. This feature is also obtained with the new matrix methodology proposed in this paper, also with good objective values.

GA is the most extended solution approach in comparison with those previously mentioned. Simaria et al. [20] developed a GA model for optimizing mixed-model assembly lines for a defined number of positions with the goal of minimizing the cycle time and balancing the workloads within the workstations. Su et al. [22] developed a solution for balancing MMALB type-2 lines which was valid for small systems, although they did not take into account the relationships between the operations and the incompatibilities between them. Barathwaj et al. [3] studied a multi-objective GA with ergonomic variables for MMALB problems, increasing the rate of production and reducing worker fatigue. Ramezanian et al. [19], using an evolutionary imperialist algorithm, proposed a mixed-model and multi-objective system where the cycle times are optimized along with the skills of the operators. Zhao et al. [25] developed a multi-objective procedure combining cycle times and mental loads for mixed-model assembly lines. Rabbani et al. [18] studied the balancing of robotic cells with mixed-model multi-objective optimization taking the cycle time, setup time and equipment cost into account. Liu et al. [13] developed a new optimization model for mixed-model assembly line balancing under uncertain demand, solved by an improved GA. Zamzam and Elakkad [24] studied the problem of balancing multi-manned assembly lines under time and area constraint using GA. None of these studies focused on the improvement of the AT for later use in real industrial cases.

Considering the branch-and-bound solution approach, Li et al. [2014] and Yang et al. [2014] considered new heuristics for balancing the manual mixed model assembly lines, using overtime work in an environment of varying demand. Hazır et al. [8] incorporated new formulas for dynamically computing a lower bound on the optimal value of the objective function and for determining the earliest workstations for tasks.

Considering the bee colony algorithms solution approach, Akpinar et al. [2] developed a novel multiple colony hybrid bees algorithm in order to improve the search capabilities of the basic bees algorithm used for solving MMALB problems with sequence-dependent setup times between tasks. Çil et al. [5] combined mathematical models and bee algorithms for the MMALB problem with physical human–robot collaboration but addressed exclusively to the improvement of the algorithmic performance.

Additionally, it is observed that most of the authors base their studies on software which is not commonly used by SMEs. In order to improve the introduction of these methodologies in SMEs, the GA technique was chosen for this study due to its simple implementation through spreadsheets; there are even low-cost commercial developments involved with this approach. This choice of methodology is also due to the NP-hard nature of ALB problems, given the efficiency of the GA according to Pinarbaşi et al. [2020].

Furthermore, Eghtesadifard et al. [2020] presented a systematic review of research themes and hot topics in ALB between 1990–2017,

highlighting that not enough research studies have been carried out into performance in real-life scenarios, which calls into question the widespread use of ALB within the industry. They also indicate that researchers can explore the literature from the perspective of whether ALB methods would be efficient in real industrial cases or not. Consequently, they require the analysis of some case studies conducted in the past, which should be accurately assessed in terms of production line efficiency, idle time or assembly-line improvement.

Conscious of the lack of research studies centred on real industrial scenarios, as pointed out by Eghtesadifard et al. [6], the contribution of this paper is oriented in breaching the gap between all the research developed and its application in real SME assembly lines of. To help achieve this objective, a new matrix methodology is proposed, as a general approach to improve AT so that the contributions made in the field of ALB by specialized authors will arrive to real the industrial cases, especially in the case of SMEs.

3. Problem description, formulation and implementation of the new matrix methodology

The new matrix methodology used to improve AT is introduced with the aim of generalizing the usage of GA in the resolution of MMALB type-2 problems, especially in SMEs. First, there is a description of the problem studied which explains the typical approach found in the literature for optimizing an MMALB type-2 problem (section 3.1). Second, the mathematical formulation and the implementation strategy of the new matrix methodology is developed (section 3.2).

3.1. Problem description and formulation without the new matrix methodology

In general, the MMALB single-target type-2 problem for single-sided linear assembly line configurations aims to minimize the total cycle time. The general assumptions, parameters and variables of the problem are as follows:

- Equivalent assembly tasks for each different product reference are almost identical, so it can be assumed that only a single precedence diagram is needed for all the product references that make up the product portfolio.
- Operating times related to assembly tasks are the same for all the different product references.
- The operating time for an assembly task is the same for all the operators because it is assumed that they have the same skill level.
- Each operator works only on a single workstation, where they carry out their assigned assembly tasks.
- Workstations and tasks are undividable.
- Product references are assembled consecutively.
- Assembly tasks roaming is avoided because the goal is minimal setup time.

Considering the assumptions mentioned above, the parameters and indices of the model are:

W: number of Workstations (obtained with the takt time explained in Section 3.3.3); w = 1, 2, . . ., W

M: number of product references included in the product portfolio to be assembled; $m=1,\,2,\,\ldots,\,M$

F: number of possible functionalities to be given to the product portfolio; $\mathbf{f}=1,\,2,...,\,F$

N: number of assembly tasks; $n=1,\,2,\,\ldots,$ N $t_n\!:$ operating time of an individual assembly task; $n=1,\,2,\,\ldots,$ N

The variables of the problem are defined as follow: d_{wn} : possible assignment of assembly task n to Workstation w. d_{wn} is binary variable (0,1). $d_{wn} = 1$ if task n is assigned to workstation w. p_{rs} : possible precedence relationship between two assembly tasks. p_{rs} is a binary variable (0,1). $p_{rs} = 1$ if assembly task r precedes assembly task s.



Fig. 2. New matrix methodology diagram steps

 C_m : Cycle time of each product reference defined with Eq. (1)

$$C_m \ge \sum_{n=1}^{N} \sum_{w=1}^{W} t_{mn} \cdot d_{wn} \ w = 1, ..., W \ m = 1, ..., M$$
(1)

Then the objective function that minimizes the sum of the cycle times of all the product references included in the product portfolio to be assembled is defined with Eq. (2)

$$[MIN]\sum_{m=1}^{M}C_{m}$$
⁽²⁾

Subject to the following constraints

$$\sum_{w=1}^{W} d_{wn} = 1n = 1, \dots, N$$
(3)

$$\sum_{w=1}^{W} w \cdot d_{sw} \ge p_{rs} \cdot \sum_{w=1}^{W} w \cdot d_{rw} \ r = 1, \dots, N, s = 1, \dots, N \mid r \neq s$$
(4)

Eq. 3 indicates that each assembly task must be assigned to one workstation. Eq. 4 handles the precedence relations between assembly tasks.

3.2. Problem description and formulation with the proposed new matrix methodology

In contrast, the new matrix methodology is proposed for application with the GA to improve the AT obtained with the methodology described in the previous section 3.1. In order to achieve the desired AT in the application of GA, which will allow a better understanding of the obtained results and the factors that influenced in the applied algorithm's decisions, it will be necessary to develop a previous preprocessing step for the initial data when modelling the problem.

The modelling proposed for the new matrix methodology, takes advantage of the fact that functional design is common good practice in the industry. Functional design involves dividing product parts in a list of product functionalities to be assembled together to obtain the product references. A company's product portfolio will usually have more product references than product functionalities; the new matrix methodology proposes taking advantage of this, while also reducing the computational cost when solving a typical MMALB type-2 problem.

This new matrix methodology consists of five steps through which the process engineers will be able to solve and analyse the example in detail. Each of the five steps is detailed below.

Fig 2 summarizes the procedure for modelling the assembly line depending on the product portfolio and balancing operation times in assigning tasks for MMALB type-2 problems, according to the proposed new matrix methodology.

3.2.1. Collection of product data

The first step consists in creating matrix A, which contains product references and product functionalities. Matrix A is given to the process engineers by the product development and marketing departments that have designed the company's product portfolio. From this initial stage, it is the process engineer's responsibility to design an optimized assembly line capable of assembling the complete company's product portfolio in terms of quality, demand flexibility and economic viability.

A: Matrix of product references and product functionalities. It has dimensions $M \bullet F$, where M is the number of product references and F is the number of product functionalities. Values within A can be 0 or 1. If $a_{mf} = 1$, it means that the product reference m includes product functionality f.

	a_{11}	a_{12}	a_{13}	 a_{1F}	
	a_{21}	a_{22}	a_{23}	 a_{2F}	
A =	a_{31}	a_{32}	a_{33}	 a_{3F}	
	a_{M1}	a_{M2}	a_{M3}	 a_{MF}	

The first column of matrix A represents the common product functionality, which includes those product components that are the common base of the entire portfolio of product references and found in all of them. Thus, *usually*, $a_{m1} = 1$ for every m.

3.2.2. Product functionalities matrix study

In the second step of the methodology, the aim is to relate and group together the different product functionalities within matrix A. This is done through two sub-steps that help to interpret which product functionalities can be grouped together and how to do so: these sub-steps are the coincident vectors method and the complementary vectors method.

3.2.2.1. Coincident vectors method. This method consists in merging identical columns of matrix A into single columns. For process engineers, this implies merging those initial product functionalities that have previously been defined as different, but which coincide in the same product references. Thus, different initial product functionalities are merged into a single product functionality.

Mathematically, if i is equal to the number of occurrences of identical columns of matrix A for each matching occurrence, then r_i is equal to the number of individual columns that are identical, where $r_i \ge 2$. Hence, through this substep, matrix A, originally of dimensions $M \cdot F$, is reduced to dimensions $M \cdot (F - k)$, where k is calculated using Eq. (5):

$$k = \sum_{i=1}^{l} (r_i - 1)$$
(5)

This grouping of product functionalities facilitates an equitable distribution of assembly tasks in later stages. This is because some of the product functionalities do not have enough operating times to achieve an equitable distribution of times between the Workstations. By grouping product functionalities, a greater number of operating times per product functionality can be achieved, which enables a more equitable distribution. This behaviour is illustrated in Section 4.

3.2.2.2. Complementary vectors method. In matrix A, the identification of complementary product functionalities in the column vectors (F - k) is required. Once added together, this results in another column vector

already existing in A.

This aggregated vector can be the unit column vector (previously referred to as "common product functionality") or another column vector of A. An example of complementary vectors is shown:

۱ =	$\begin{bmatrix} a_{11} \ 1 \ 0 \ \dots \ a_{1(F-k)} \\ a_{21} \ 1 \ 0 \ \dots \ a_{2(F-k)} \\ a_{31} \ 1 \ 0 \ \dots \ a_{3(F-k)} \end{bmatrix}$
	$a_{M1} 1 0 \dots a_{M(F-k)}$

A

It can be observed how product functionalities 2 and 3 are complementary columns which, once aggregated, result in the common product functionality 1, which is a frequent case in real assembly lines.

Another case of complementary vectors is presented, where product functionalities 5 and 6, if aggregated, result in product functionality 4:

	a_{M1}	a_{M2}	a_{M3}	1	1	0	 $a_{M(F-k)}$
A =	a_{31}	a_{32}	a_{33}	1	0	1	 $a_{3(F-k)}$
	a_{21}	a_{22}	a_{23}	0	0	0	 $a_{2(F-k)}$
	a_{11}	a_{12}	a_{13}	1	1	0	 $a_{1(F-k)}$

Sometimes it is possible to find alternative combinations in the identification of complementary vectors. In order to obtain a meaningful AT for process engineers, it is advisable to identify alternatives with complementary vectors that represent product functionalities related by both their utility and the necessary manufacturing tools. With the manual choice of the best complementary vectors alternative, the natural complementarities of the system are included in the model, thereby implicitly defining additional restrictions to solve the problem. These help to improve the feasibility of the final solutions obtained with the optimization algorithm, without requiring their explicit programming, thus simplifying definition of the model made by the process engineer. In section 4, an example is shown of a possible infeasible complementary vector.

However, this manual choice does not require a high workload for SME process engineers, as the number of possible complementary product functionalities is usually limited.

By identifying these complementary vectors, product functionalities with complementary occurrence are found. When added, these will be compatible in the balancing with their aggregated product functionality in A.

Identifying the complementary vectors leads to understanding the relationship between the assembly tasks associated with these complementary product functionalities and how they can affect the results when solving a MMALB type-2 problem. Breaking down the assembly tasks distribution in this way, process engineers obtain more understanding regarding the solution and consequently improve AT, as desired in this paper.

When balancing operation times of assembly tasks, it is very unlikely that these complementary product functionalities will have identical operation times in each of the Workstations. Consequently, only the minimum operation time assigned to the assembly tasks affected by these complementary product functionalities can be balanced by the optimization algorithm together with the time of the aggregated product functionalities.

This aggregate time facilitates the equitable distribution of assembly tasks in later stages of the new matrix methodology. The remaining time (debt) in the complementary product functionalities with times greater than the minimum, helps to identify systemic imbalances in the solution and thus to guide the process engineers' decisions; for instance, modifying the existing manufacturing equipment, as a result of the soughtafter improvement in the AT. 3.2.3. Study of the assembly tasks matrix B and definition of number of Workstations

This step includes the preparation of the assembly tasks and product functionalities matrix B.

3.2.3.1. Definition of assembly tasks matrix B. B is the matrix of assembly tasks and product functionalities, as presented. It has dimensions $N \cdot (F - k)$, where N is the number of assembly tasks necessary to assemble all the product references and (F - k) is the number of product functionalities left after applying the coincident vectors and complementary vectors methods.

B =	$\begin{bmatrix} b_{11} \\ b_{21} \\ b_{31} \end{bmatrix}$	$b_{12} \\ b_{22} \\ b_{32}$	$b_{13} \\ b_{23} \\ b_{33}$	 $\left egin{array}{c} b_{1(F-k)} \ b_{2(F-k)} \ b_{3(F-k)} \end{array} ight $
	\dots b_{N1}	 b _{N2}	 b _{N3}	 $\begin{bmatrix} b_{N(F-k)} \end{bmatrix}$

Values within B can be 0 o 1. If $b_{nf} = 1$, it means that the assembly task n is required to implement the product functionality f to a product reference. B is defined by the process engineer based on the engineer's tools, skills and previous knowledge concerning the company's manufacturing processes. Assembly tasks must be defined as indivisible due to the use of specific assembly tools or materials. The use of the minimum operation times offers the possibility of obtaining better balanced assembly lines; the shorter the time of the disaggregated assembly tasks, the greater the precision when balancing.

To obtain the operation time for each assembly task, the process engineer must break these down into basic tasks and calculate the operation times using tabulated or experimental time measurement systems.

Hence, a diagonal matrix T is generated, with the diagonal containing individual operation times (t_n) for each assembly task. These operation times (t_n) are represented in a diagonal matrix T in order to conduct the next steps of the new matrix methodology.

3.2.3.2. Calculation of the number of Workstations according to the takt time. Prior to running the GA for an MMALB type-2 problem, it is necessary to determine the number of Workstations W according to the takt time t_t using Eq. (6) and (7). This is a very common practice in industry and obtains the minimum number of Workstations for the assembly line balancing, ensuring on-time delivery.

$$t_t = \frac{t_w}{c} \tag{6}$$

$$W \ge \frac{t_{cmax} \cdot c_s}{t_t} \tag{7}$$

Here, t_w is the available shift work time for each Workstation, c is the necessary daily capacity according to the client and $t_c \max$ is the longest possible cycle time for any of the product references (MMALB problem). In addition, a safety coefficient c_s can be applied from the historical efficiency values. $t_c \max$ is calculated by summing up the times of all the product functionalities, except those that have been previously classified as complementary product functionalities. For these product functionalities, only the highest operation time is added (t_n).

3.2.3.3. Definition of initial chromosome. Due to the iterative nature of the GA, it is necessary to generate an initial chromosome vector from which the GA starts the resolution.

The chromosome $(chrom_{1:N})$ is a vector that contains the allocation of assembly tasks in the required Workstations (W), according to the takt time. For each Workstation a N-dimensional row vector d_w is defined with Eq. (8):

$$d_{w} = \begin{bmatrix} d_{w1} & d_{w2} & d_{w3} & \dots & d_{wN} \end{bmatrix}$$
(8)

Values of d_{wn} can be either 0 or 1. If $d_{wn} = 1$, it means that the assembly task n is assigned to the Workstation w. If $d_{wn} = 0$, it means that the assembly task n is not assigned to the Workstation w and must therefore be assigned to another Workstation.

The sum of d_w vectors multiplied by their corresponding Workstation number w (w being a natural number) defines the chromosome to be iterated by the GA and presented in Eq. (9).

$$chrom_{1:N} = \sum_{1}^{W} w \cdot d_{w}$$
⁽⁹⁾

The GA carries out iterations until the solution that optimizes the system is found, thus updating the components of the chromosome.

3.2.3.4. Consequences of previous steps on the operation times. The allocation of the operation times of assembly tasks for each individual Workstation is obtained through Eq. (10). For each Workstation, *tw* is a row vector of dimension N:

$$tw_w = d_w \cdot T \tag{10}$$

Then, the operation times of assembly tasks of all the individual Workstations are grouped into a single matrix. TW has as many rows as Workstations (W) and as many columns as assembly tasks (N)

TW =	tw_1 tw_2 tw_3	=	t_{11} t_{21} t_{31}	$t_{12} \\ t_{22} \\ t_{32}$	t_{13} t_{23} t_{33}	 t_{1N} t_{2N} t_{3N}
	 tw _w		t_{W1}	 t _{W2}	 t _{W3}	 t _{WN}

Then, Eq. (11) is applied to obtain matrix TWF,

 $TWF = TW \cdot B$

$$= \begin{bmatrix} t_{11} & t_{12} & t_{13} & \dots & t_{1N} \\ t_{21} & t_{22} & t_{23} & \dots & t_{2N} \\ t_{31} & t_{32} & t_{33} & \dots & t_{3N} \\ \dots & \dots & \dots & \dots & \dots \\ t_{W1} & t_{W2} & t_{W3} & \dots & t_{WN} \end{bmatrix} \cdot \begin{bmatrix} b_{11} & b_{12} & b_{13} & \dots & b_{1(F-k)} \\ b_{21} & b_{22} & b_{23} & \dots & b_{2(F-k)} \\ b_{31} & b_{32} & b_{33} & \dots & b_{3(F-k)} \\ \dots & \dots & \dots & \dots & \dots \\ b_{N1} & b_{N2} & b_{N3} & \dots & b_{N(F-k)} \end{bmatrix}$$
(11)

TWF is a matrix of dimensions $W \cdot (F - k)$. Each element of the matrix, twf_{wf} , represents the allocation time for assembling the product functionality f to the Workstation w.

To achieve an optimal modelling of the system, TWF must be corrected by considering the previously identified complementary product functionalities. For each Workstation w, the minimum sum of the allocation times twf_{wf} of the complementary product functionalities will be added to the sum of allocation times twf_{wf} of the added functionality.

The twf_{wf} value of the aggregate product functionality is increased, which improves the chances of obtaining a solution with better assembly line balancing in comparison to merely applying the GA without identifying complementary vectors. The debts values, defined as the difference between the minimum twf_{wf} and the twf_{wf} of the complementary product functionalities which have a twf_{wf} greater than the minimum, are important for the process engineers, as they indicate and quantify the systemic imbalances in the assembly line.

The debt values which differ from 0 cannot currently be solved by the GA, this means that process engineers require technical modifications of the manufacturing tools currently available for the Workstations. This is valuable information that provide the searched AT to the model obtained and indicate the way towards optimizing the system.

An example is provided, for further clarity, where product functionality 2 and product functionality 3 are complementary to product functionality 1, which is the common product functionality. The times of the matrix TWF are thus modified as follows, giving rise to a new matrix TWF' as shown.

$$TWF' = \begin{bmatrix} twf_{11} + \min(twf_{12}, twf_{13})twf_{12} - \min(twf_{12}, twf_{13})twf_{13} - \min(twf_{12}, twf_{13})\dots twf_{1\cdot(F-k)} \\ twf_{21} + \min(twf_{22}, twf_{23})twf_{22} - \min(twf_{22}, twf_{23})twf_{23} - \min(twf_{22}, twf_{23})\dots twf_{2\cdot(F-k)} \\ twf_{31} + \min(twf_{32}, twf_{33})twf_{32} - \min(twf_{32}, twf_{33})twf_{33} - \min(twf_{32}, twf_{33})\dots twf_{3\cdot(F-k)} \\ \dots \dots \dots \dots \\ twf_{W1} + \min(twf_{W2}, twf_{W3})twf_{W2} - \min(twf_{W2}, twf_{W3})twf_{W3} - \min(twf_{W2}, twf_{W3})\dots twf_{W\cdot(F-k)} \\ \end{bmatrix}$$

The twf_{wf} which are modified due to the use of the complementary vectors method are named twf'_{wf} . For instance, $twf'_{11} = twf_{11} + min(twf_{12}, twf_{13})$.

Operation times of assembly tasks associated to product functionalities which have not been aggregated due to the coincident or the complementary vectors method remain the same, twf_{wf} .

TWF['] is a new modelling of the system which will reflect the assignment of assembly tasks for each Workstation more clearly and will allow for systemic imbalances to be easily identified, thus providing the desired improvement of the AT to the application of the GA, as will be shown in the industry-based paradigmatic example of Section 4.

3.2.4. Model resolution

Once the new matrix methodology is carried out, results are obtained starting from Eq. (9) and an initial matrix TWF' is obtained. The objective is to optimize this matrix with the value of the chromosome by means of a GA.

3.2.4.1. Definition of the new objective function and constrains. The general assumptions, parameters and variables of the problem formulation for the new matrix methodology are the same as those described in section 3.1, with the exception of objective function of Eq. (2).

In general, the MMALB single-target type-2 problem for single-sided linear assembly line configurations aims to minimize the total cycle time; the new matrix methodology uses the minimization of the standard deviation of the aggregated operation times, assigned to each of the workstations, for each group of product functionalities. This is due to the simple integration with the new methodology proposed and its proven efficiency by Kim et al. [10].

The objective function of the MILP model indicated in Eq. (2), as it is not based on product functionalities, it is not suitable for solving the problem with the new matrix methodology.

The application of the proposed new matrix methodology has generated the objective variables that are to be studied, which are the scalars twf_{wf} that represent the sum of all the operating times of the assembly tasks assigned to a single Workstation *w* for one of the product functionalities *f*. Then, the formulation of the objective function based on the standard deviation with Eq. (12) is used

$$[MIN]\Delta T = Min\left(\frac{\sum_{f=1}^{F-k} \sqrt{\frac{\sum_{w=1}^{W} \left(twf_{wf} - \overline{twf_{wf}}\right)^{2}}{W}}}{F-k}\right)$$
(12)

The objective function aims to minimize the sum of the standard deviations (ΔT) between the operation times assigned to each of the Workstations for each product functionality. The minimization of ΔT is obtained by means of the GA, acting as the fitness function. Fitness functions are used in GAs to guide simulations towards the optimal solution. A fitness function is a particular type of objective function that is used to summarize, as a single figure of merit, how close a given solution is to achieving the objective. In addition, the following restrictions are required to ensure the feasibility of the solution for the assembly line:

3.2.4.2. Application of the GA. The stages for the application of the GA with the new matrix methodology are listed below:

- 1 Randomly assign assembly tasks to the required Workstations in the spreadsheet, creating the initial chromosome.
- 2 Calculate ΔT of the initial chromosome with the spreadsheet. The best and closest result to the objective corresponds to the *Min* ΔT .
- 3 By using commands and programming tools defined in the GA solver of the spreadsheet, apply genetic operators of reproduction, crossing and mutation to the current generation to produce the next one.
- 4 Select the most competent chromosomes of the new generation based on ΔT .
- 5 Go back to stage 3 until the stop condition is satisfied. This condition can be, for example, an elapsed computational time without an improvement of ΔT .
- 6 When the stop condition is reached, the best-fit chromosome is returned. This is the best achieved solution for the assembly line balancing.

3.2.5. Model analysis

3.2.5.1. GA-improved AT. Using the GA with the established restrictions, once the chromosome that minimizes the defined fitness variable is found, the values of matrix *TWF* can be obtained.

Improved AT is attained by means of analysis of these values, twf_{wf} , by the process engineers, as well as obtaining the standard deviations in the positions of each of these product functionalities. Thus, in the analysis of the model, two cases of interest in TWF' for the process engineers can be distinguished:

a Non-complementary product functionalities

The imbalances observed in these product functionalities correspond to an insufficient amount of assembly tasks or to excessively disparate operation times. Despite operation times being grouped using the coincident vectors method and the complementary vectors method, in many cases it is not feasible to find a good balance between Workstations and the assigned assembly tasks of product functionalities.

These systemic imbalances can be corrected by the process engineer by modifying the manufacturing tools, reducing some of the operation times or further atomizing the breakdown of assembly tasks, although the latter will not be possible if such a breakdown has already been implemented before.

Any action carried out by the process engineer following the guidelines shown by the model will contribute to an optimization, leading to a better balance in the system for all mixed-model products assembled in the line.

a Complementary product functionalities

In the proposed new matrix methodology, these product functionalities devote part of their assembly tasks' operation times to increasing the amount of assembly tasks of the added product functionalities, specifically with their minimum twf_{wf} value in each Workstation. This contribution helps to improve the distribution among Workstations of such added product functionalities.

Product functionalities, f							
Common 1	Chassis 2/3/4	Front Doors 5	Rear Doors 6	Air cond. 7	Clima 8		
	Family References	Yes	Yes	Option	Option		
6 ⁽³)	Van References	Yes	No	No	No		
	Pick-up References	Yes	No	No	No		

Fig. 4. Flexible modular vehicle from García [7]

Table 1

Matrix A of product references and product functionalities as described in step 3.2.1

	Product function	onalities, f						
Product Reference m	Common 1	Family Chassis 2	Van Chassis 3	Pick up Chassis 4	Front Doors 5	Rear Doors 6	Air cond. 7	Clima 8
1	1	1	0	0	1	1	1	0
2	1	0	1	0	1	0	1	0
3	1	0	0	1	1	0	0	0
4	1	1	0	0	1	1	0	1

Table 2

New matrix A after applying the coincident vectors method step

	Product functionalities, f					
Product Reference m	Common + Front Doors 1,5	Family Chassis + Rear Doors 2,6	Van Chassis 3	Pick up Chassis 4	Air cond. 7	Clima 8
1	1	1	0	0	1	0
2	1	0	1	0	1	0
3	1	0	0	1	0	0
4	1	1	0	0	0	1

However, for each product functionality and Workstation, once the minimum of the complementary product functionalities has been discounted, there is a remainder that can be 0 (for the product functionality that marks the minimum time) or higher.

If $twf'_{wf} \neq 0$, this is a debt. These debts twf'_{wf} show the undesirable variability of time between Workstations and clearly point out the sys-

temic imbalances caused by these product functionalities, since they are an accurate guide for optimizing the balancing through the modification of the manufacturing tools.

Table 3

Matrix B with the	e assembly tasks and	their assigned	product functionalities.
-------------------	----------------------	----------------	--------------------------

		Product functionalities, f					
n	Assembly tasks	Common + Front Doors 1,5	Family Chassis + Rear Doors 2,6	Van Chassis 3	Pick up Chassis 4	Air cond. 7	Clima 8
1	Position chassis	1	0	0	0	0	0
2	Assembly wheels	1	0	0	0	0	0
3	Wire chassis	1	0	0	0	0	0
4	Place mats	1	0	0	0	0	0
5	Mount fam. chassis	0	1	0	0	0	0
6	Screw fam. chassis	0	1	0	0	0	0
7	Mount Van chassis	0	0	1	0	0	0
8	Screw Van chassis	0	0	1	0	0	0
9	Assembly Pickup chassis	0	0	0	1	0	0
10	Screw Pickup chassis	0	0	0	1	0	0
11	Assemble Front Door	1	0	0	0	0	0
12	Front windows	1	0	0	0	0	0
13	Assemble Rear Doors	0	1	0	0	0	0
14	Rear windows	0	1	0	0	0	0
15	Fit air conditioning.	0	0	0	0	1	0
16	Check Air Conditioning	0	0	0	0	1	0
17	Fit clima	0	0	0	0	0	1
18	Check clima	0	0	0	0	0	1
19	Basic Check	1	0	0	0	0	0
20	Packing	1	0	0	0	0	0



Fig. 5. Precedence diagram of assembly tasks

4. Example

In this section an industry-based paradigmatic example is presented to demonstrate the implementation of the new matrix methodology.

4.1. Description of the problem

The proposed new matrix methodology is not limited exclusively to improving the AT and the performance of a GA in a single example. It also proposes a complete organizational and design framework in which the GA is a suitable algorithm among AI algorithms (e.g., Bee colony, ACO, simulated annealing, etc.). Consequently, it is not possible to use a standard problem set to analyse its performance, as the standard problems are not prepared for the application of the proposed new matrix methodology. Therefore, the results of the improved AT are demonstrated using an industry-based paradigmatic example, which meets the necessary requirements and a GA is used as an optimization algorithm to solve the problem.

The choice of GA to solve the problem is due to its more common use among other AI algorithms, which allows its application through spreadsheets and guarantees accessibility to the SME process engineer.

The industry-based example consists in designing a new assembly line for a product portfolio consisting in a modular vehicle like the one shown in Fig. 4. This industry-based example represents an MMALB type-2 problem paradigmatic and simplified for these SMEs to clearly show the improvement of the AT.

The matrix of product references and product functionalities A is presented in Table 1. In order to facilitate the identification of complementary product functionalities, columns are grouped separately in Tables 1, 2 and 3. Given its relevance, the common product functionality has also been presented separately.

As can be observed in Table 1, in this example there are more product functionalities (F=8) than product references (M=4) to facilitate the understanding of the implementation of the new matrix methodology. However, real industrial problems commonly have more product references than product functionalities. Later, in Fig. 6, it is proven using a

larger example with more product functionalities and product references, closer to real industrial scenarios.

Using the proposed new matrix methodology (step 3.2.2), two coincident vectors can be detected in Table 1: f = 5 is coincident with f = 1, and f = 6 is coincident with f = 2. Therefore, these product functionalities are grouped together according to Eq. (2) such that k = 2 (i = 2, $r_1 = 2$, $r_2 = 2$). The new matrix A is presented in Table 2:

The following complementary vectors can now be detected: product functionalities 2 + 3 + 4 = common product functionality 1. As was previously stated, the case where the addition of the complementary vectors is equal to the unit column corresponding to the common functionality is frequently found in real assembly lines.

Other complementary product functionalities could be detected, such as product functionalities 4 + 7 + 8 = common product functionality 1. However, this complementarity, despite being mathematically feasible, as explained in Section 3.2.2, is not advisable in manufacturing because the equipment for assembling a chassis differs greatly (in size, space...) from the manufacturing equipment for assembling an air comfort option between air conditioning or clima. Thus, the process engineer will discard it in favour of the one selected in the example, which is that product functionalities 2 + 3 + 4 = commonproduct functionality 1. This is because product functionalities 2, 3 and 4 require very similar tools. Thus, the process engineer will get tool cost savings if the product functionalities are defined as complementary, since their assembly tasks will be in nearby locations after model's the resolution, without the need to include additional constraints in the system. This is the step in which the process engineer's knowledge of the particular manufacturing resources becomes especially important, guiding the feasibility of the final solutions as a result.

For this reason, in the case of SMEs, it is recommended that the process engineers perform this step manually, since the number of product functionalities is generally small enough.

Table 3 presents the assembly tasks matrix B defined in step 3.2.3. The assembly tasks (N=20) are defined as being indivisible.

The precedence diagram for assembly tasks is represented in Fig. 5. Its function is to restrict the possible solutions to guarantee the

	Check Basic Basic clima Check Check	10 10 5
	Fit clima	15
	Check Air Conditioning	10
	Fit air conditioning.	10
	Rear windows	л
	Assemble Rear	Doors 15
	Front windows	л
	Assemble Front	Door 10
	Screw Pickup	chassis 5
	Assemble Pickup	chassis 10
	Screw Van	chassis 7
	Mount Van	chassis 15
	Screw fam.	chassis 10
sks	Mount fam.	chassis 20
mbly ta:	Place mats	ß
r the asse	Wire chassis	25
imes (t _n) foi	Assemble wheels	15
peration ti	Position chassis	20
Table 4 Individual o	Assembly tasks,n	t, [min]

Operations Research Perspectives 9 (2022) 100223

feasibility of the solutions found by the GA, according to the process engineer's technical criteria.

Table 4 illustrates the individual operation times (t_n) for each assembly task in Table 3. The values t_n are then used to define the diagonal of matrix T.

To calculate the number of Workstations for the example, the customer requests c =1500 units/year to the manufacturing SMEs. To design the assembly line, the process engineer has 250 annual workdays in single shifts of 8 h ($t_w = 250 \cdot 8 \cdot 60$, in minutes). In addition, a security factor of $c_s = 1,2$ is assumed. Using the information from Tables 2, 3 and 4 and Fig. 6, $t_c_{max} = t_1 + t_2 + t_3 + t_4 + t_5 + t_6 + t_{11} + t_{12} + t_{13} + t_{14} + t_{15} + t_{16} + t_{17} + t_{18} + t_{19} + t_{20} = 170$ min as explained in step 3.2.3. Using Eq. (6) and (7), the required number of Workstations is W = 3. Therefore, there are 3 vectors d_w (d_1 , d_2 , d_3) that have initially been randomly defined using Eq. (8), as shown in Table 5.

As a result, the evolutionary chromosome **chrom**_{1.20} is defined with values between 1 and 3, identifying the assigned Workstation for each assembly task. The initial chromosome is obtained through Eq. (9). It corresponds to the initial distribution of assembly tasks per Workstation and will be iteratively balanced throughout the GA in order to fulfil the precedence restrictions defined in Fig. 5. Then a non-optimized value of **TWF** is obtained with Eq. (11).

4.2. Resolution

Once the GA is applied to the modelling and solved the optimization function of Eq.(12), the solution obtained is presented with the assembly tasks assigned to the Workstations.

Thanks to the new matrix methodology the following modelling \mathbf{TWF}' is obtained and presented.

	45	10	5	0	0	0	
$\mathbf{TWF}' =$	30	10	0	0	10	15	
	35	15	2	0	10	10	

The GA ran on a personal computer with 4 CPU cores and a base clock of 2.1 GHz, requiring an average time of 84 s. with an average iteration number of 12.400, using Solver evolutionary tools in Microsoft Excel with the population-based method provided by Excel; this involves a population size of 100, a convergence rate of 0,0001, a mutation rate of 0.075 and a crossover rate with these standard excel values.

Table 6 is presented to clearly demonstrate the improved AT achieved when applying the proposed new matrix methodology along with a GA.

According to the allocation of assembly tasks after running the GA, the distribution of the aggregated operation times for each product functionality for each Workstation is calculated, using the modelling generated with the new matrix methodology. These aggregated operation times have been balanced according to Eq. (12) to minimise the standard deviation between Workstations for each of the defined product functionalities. Starting from TWF['], Table 6 can be developed by acquiring the TWF['] and illustrating the minimums of the complementary product functionalities. Furthermore, the standard deviations of both the complementary and non-complementary functionalities are also calculated using Eq. (12).

Standard deviations for common Functionality 1 with minimum time between functionalities 2, 3 and 4 (using Eq. (13)).

Standard deviations for 2, 3 and 4 time debts, using Eq. (13). The GA also minimizes them. Time debts are the differences between 2, 3, 4 and the minimum value between them.

Standard deviations for 7 and 8 time debts, using Eq. (13). The GA also minimizes them.



Fig. 6. Comparative computational graph with the standard genetic algorithm

The shadowed column in grey (Min^{*}) means the minimum time between product functionalities 2, 3 and 4. For instance, for workstation 1, the values twf_{w2} , twf_{w3} and twf_{w4} are 20, 15 and 10 min respectively, therefore the minimum value obtained by the GA is 10.

Thus, this minimum value has been added to the common product functionality being, as a result, the obtained value with GA in twf'w1, 45 min ($t_1 + t_2 + 10$).

As can be seen, the new matrix methodology is not used to carry out the calculations of the GA but rather to build the modelling. It is this modelling in which the GA optimizes the chromosome until it finds the numerical values of the solution.

By applying Eq (12), a final average value is obtained between the standard deviations of the product functionalities of $[Min]\Delta T = 2,56$ min, which has been minimised by the GA.

In Table 6, columns of complementary product functionalities twf'_{wf} illustrate the time debts of these product functionalities, representing the origin of the imbalances in the assembly line, which can be easily analyzed. It also allows us to foresee the imbalances generated by future modifications, thereby facilitating their planning and avoiding costly changes in the location of the tools.

Contrary to the previous methodologies analyzed in the literature review, a schematic modelling is obtained that shows how the balancing has been carried out and also shows the opportunities for improvement. Additionally, it is applicable even with spreadsheets.

In this way, a methodology is proposed that can be generalized due to its simplicity of analysis and its low implementation costs.

4.3. Discussion of the AT achieved and other advantages of applying the new matrix methodology

4.3.1. Enhanced Algorithm Transparency for analyzing future modifications in the ALB problem

Future modifications can be generated at the of assembly tasks level, such as modifications of the manufacturing tools, operation times or implementing new assembly tasks. The analysis of the effects of these future modifications is an immediate calculation, so process engineers can evaluate and understand these effects when balancing the designed assembly line simultaneously for the whole product portfolio.

Considering the example of Section 4.1, a modification of product

functionality 7 involving an additional assembly task with operation time of $t_{21} = 5$ s (new row in Table 3 and new column in Table 4) with no predecence restrictions can be implemented. The possible alternatives can then be quantified in Table 7:

Table 7 allows an immediate decision: task t_{21} is included in $\mathbf{w} = 1$ since it has the lowest standard deviation value and is therefore the most balanced alternative for all the product references. These alternatives must be compared with column $\mathbf{twf'}_{w7}$ of Table 6. This optimization is valid for any combination of product functionalities for all product references.

4.3.2. Enhanced AT for intuitive diagnosis of systemic imbalances

The modelling offers a visual interface that details the inherent imbalances due to product definition and assembly tasks, thus immediately showing how to correct them. Process engineers can detect the inherent imbalances in Table 6 in the colums for time debts by product functionality at each of the Workstations. The improvements carried out by the process engineer to minimize each of these debts can then lead to the perfect balance of the assembly line for the product portfolio as a whole.

For example, considering Table 6, column twf _{w4}, it can be observed that the standard deviation is 0. Therefore, this product funcionality does not need to be modified from a manufacturing perspective because it does not bring an imbalance to the assembly line for all product references: the allocation of assembly task 9 in workstation 1 (operation time, $t_9 = 10$ min) and assembly tasks 10 in workstation 3 ($t_{10} = 5$ min) is suitable for the whole product portfolio.

On the other hand, considering column twf $_{w8}$, there is a large standard deviation (6.24 min). This product functionality needs to be modified from a manufacturing perpective because it is probably impossible to find a more suitable balance with the current assembly tasks. Thanks to AT, the process engineer can detect this situation and study the solutions, such as improving tools to minimize operation times or dividing the assembly tasks 17 and 18 if possible, reassigning the new assembly tasks from the functional modelling of Table 6 with the criteria for reducing the standard deviations of the product functionalities. Any of these solutions will minimize imbalances in the assembly line.

	c Basic	ck Check		d _{w20}	0	0	1	
	c Basi	Che		d_{w19}	0	0	1	
	Check	clima		d_{w18}	0	0	1	
	Fit	clima		d_{w17}	0	0	1	
	Check Air	Conditioning		d_{w16}	0	0	1	
	Fit air	conditioning.		d_{w15}	0	0	1	
	Rear	windows		d_{w14}	0	0	1	
	Assemble	Rear	Doors	d_{w13}	0	0	1	
	Front	windows		d_{w12}	0	1	0	
	Assemble	Front	Door	d_{w11}	0	1	0	
	Screw	Pickup	chassis	d_{w10}	0	1	0	
	Assemble	Pickup	chassis	6wb	0	1	0	
	Screw	Van	chassis	\mathbf{d}_{w8}	0	1	0	
	Mount	Van	chassis	\mathbf{d}_{w7}	0	1	0	
	Screw	fam.	chassis	d _{w6}	1	0	0	
	Mount	fam.	chassis	\mathbf{d}_{w5}	1	0	0	
	Place	mats		\mathbf{d}_{w4}	1	0	0	ĺ
lauy.	Wire	chassis		\mathbf{d}_{w3}	1	0	0	
nini nenner	Assemble	wheels		$\mathbf{d}_{\mathbf{w}2}$	1	0	0	
randomly c	Position	chassis		\mathbf{d}_{w1}	1	0	0	
ctors dw	Assembly	tasks,n		d"	d ₁	d2	d ₃	

Table

Operations Research Perspectives 9 (2022) 100223

4.3.3. Enhanced Algorithmic Transparency for the immediate balancing of new product references

Once an optimal solution is achieved for the assembly line, the same balance is shared by any added product reference that meets the conditions of coincident and complementary vectors of the model.

For instance, a new product reference is presented in Table 8 and added to the paradigmatic industry-inspired example, despite not being in the initial portfolio. It must to be assembled in the designed assembly line ($\mathbf{m} = 5$).

The new product reference does not modify the results presented in Table 6 through the proposed new matrix methodology. Hence, the methodology is also valid for this new product reference. Adding new product references does not force a rebalancing of the system. This lowers the industrialization costs and minimizes the delivery times of the new references to the customer. Note that this is a common situation in real cases for SMEs, where new product references can be required due to marketing issues (or product references may be cancelled) and it is not possible to redesign the planned assembly line.

Table 9 shows the results of applying the modelling obtained in Table 6 to product references 1, 2, 3, 4 and to the new product reference 5 that has been added to the assembly line. It can be seen that, thanks to balancing by product functionality through the proposed new matrix methodology, the new product reference is automatically balanced since it meets the conditions of the coincident and the complementary vectors method.

However, if the GA is used alone, the inclusion of a new product reference means reapplying the GA, possibly obtaining a second solution unrelated to the first. For this reason, this second solution obtained may not be feasible to implement, due to the involved cost of modifing the workstations positions for the new assembly line configuration.

4.3.4. Automated and fast resolution

The resolution of systems of different sizes using the new matrix methodology compared to simply using the GA by itself is carried out with the aim of evaluating the computational cost. Results are presented in Fig. 6, solving 12 examples derived from the initial example but with higher M and F dimensions. These are obtained with the same personal computer used beforehand. Table 10 shows the dimensions of evaluated examples:

Although the number of GA iterations required to find the solution is similar in both cases, the time per iteration of the proposed new matrix methodology is generally much lower and it is proportional to the number of product functionalities F of the model. On the other hand, the cycle time of the GA method by itself is proportional to the number of product references M in the system.

It must be taken into account that the number of product references M is a consequence of the combination of the product functionalities present in the model, therefore the theoretical maximum number of product references M is larger than the number of product functionalities F. This usually implies a very different magnitude of time per iteration between both methodologies.

As shown in Fig. 6, the execution times were shorter for the new matrix methodology, with wide differences, which makes it advisable to apply a logarithmic scale to display the results correctly. Both methods show a similar tendency to fall into local minimums. In all cases, a very close result to the optimal outcome was reached, which in the example on the graph was a fitness value $[Min]\Delta T = 0,45$ min.

Table 11 compares three methods for confronting the assembly line balancing problem in the SMEs: 1) traditional human balance, 2) reference modelling plus the GA stand alone, 3) proposed new matrix methodology alongside the GA.

The table summarizes the contributions, demonstrated by the example of Section 4 and the discussion of Section 4.3, obtained with the new matrix methodology proposed in this paper.

Functional modelling after the model's resolution

	Product functionalities, f									
	Common + Front Doors 1,5			Family Chassis + Rear Doors 2,6		Van Chassis 3		Pick up Chassis 4	Air cond. 7	Clima 8
w	twf' _{w1}	Min*	twf' _{w2}	twf _{w2}	twf' _{w3}	twf _{w3}	twf' _{w4}	twf _{w4}	twf' _{w7}	twf _{w8}
1	45	10	10	20	5	15	0	10	0	0
2	30	0	10	10	0	0	0	0	10	15
3	35	5	15	20	2	7	0	5	10	10
STD Dev.	6,24		2,36		2,05		0,00		4,71	6,24

Table 7

Alternative effects of adding a new assembly task to product functionality 7

	Air cond. 7	Air cond. 7 (t_{21} in $w = 1$)	Air cond. 7 (t_{21} in $w = 2$)	Air cond. 7 (t ₂₁ in $w = 3$)
W	twf' _{w7}	twf _{w7}	twf' _{w7}	twf _{w7}
1	0	5	0	0
2	10	10	15	10
3	10	10	10	15
STD Dev.	4,71	2,35	6,23	6,23

Table 8

Occurrence of product functionalities for the new product reference added to the assembly line

	Product functionalities, f								
m	Common 1	Family Chassis 2	Van Chassis 3	Pick up Chassis 4	Front Doors 5	Rear Doors 6	Air cond. 7	Clima 8	
5	1	0	1	0	1	1	0	1	

Table 9

Workstation times (min) per product reference after adding a new on	e
---	---

w =	1	2	3
Reference 1	55	50	60
Reference 2	50	40	47
Reference 3	45	30	35
Reference 4	55	55	60
Reference 5	50	45	47

Table 10

M and F dimensions of evaluated examples

Example	1	2	3	5	6	7	9	11	12
Μ	20	200	1200	20	200	1200	20	200	1200
F	9	9	9	18	18	18	36	36	36

5. Conclusions

To increase the use of genetic algorithm for assembly line balancing among SMEs, this article proposes a new matrix methodology for improving the AT when balancing assembly lines that can be classified as MMALB type-2 problems, in contrast to the AT obtained using a standalone GA. The current AT achieved with the use of stand-alone GA could be a crucial barrier for SMEs process engineers and little of the scientific

Table 11

Comparison of the balancing methods found in the SMEs

literature reviewed has focussed on improving AT.

As a consequence of applying the new matrix methodology together with the usage of a GA, the AT of the results is enhanced for process engineers as follows: opportunities for the improvement of the assembly line designed are clearly displayed; it offers an intuitive diagnosis of systemic imbalances as a consequence of placing assembly tasks in certain workstations; there is no need to adjust the designed assembly lines when adding new product references, since they are immediately balanced if the new product references meet the conditions of coincident and complementary vectors.

This enhanced AT is demonstrated in an industry-based example, in order to show clearly how the GA could be widely implemented as a simple tool for balancing assembly lines for SMEs.

Another advantage is that the proposed methodology using the GA is even applicable through spreadsheets, which are widely used among SMEs. Moreover, a reduction in computational cost is also demonstrated, which becomes more important as the number of product references increases, thus making the methodology not only suitable for many SMEs but also even for larger companies.

As a main future research, it is proposed to carry out an accurate analysis in terms of the obtained assembly line balancing between workstations, the achieved Algorithm Transparency of the results and measurable improvements implemented in a real case. With the results obtained, another future study would involve applying the entire proposed approach to other types of ALB problems, such as parallel, twosided and U-line set-ups.

Support	Advantage Feature	Productivity MOD	Productivity MOI	Flexibility	Methodology Traditional human balance	Reference modelling + GA.	New Matrix + GA
GA	Automated and fast resolution	-	+	+	\boxtimes	\checkmark	\checkmark
AT	Enhanced transparency for analysing future modifications	-	+	+	\boxtimes	\boxtimes	
	Enhanced transparency for intuitive diagnosis of systemic imbalances.	+	+	+	\boxtimes	\boxtimes	
Matrix modelling	New default balanced product references *	+	+	+	\boxtimes	\boxtimes	

* The new references must be compatible with the previously defined matrix A.

Additionally, the initial matrix methodology can also be applied along with other algorithms such as ACO, hybrid bees or simulated annealing to analyse whether the AT and performance improvement are similar to GA.

6. AUTHORSHIP STATEMENT

Manuscript title: New Matrix Methodology for Algorithmic Transparency in Assembly Line Balancing Using a Genetic Algorithm

All persons who meet authorship criteria are listed as authors, and all authors certify that they have participated sufficiently in the work to take public responsibility for the content, including participation in the concept, design, analysis, writing, or revision of the manuscript. Furthermore, each author certifies that this material or similar material has not been and will not be submitted to or published in any other publication before its appearance in the Operations Research Perspectives.

7. Authorship contributions

7.1. Category 1

Conception and design of study: J.I. Anel, P. Català; acquisition of data: J.I. Anel, P. Català, M. Serra; analysis and/or interpretation of data: J.I. Anel, B. Domenech.

7.2. Category 2

Drafting the manuscript: J.I. Anel, P. Català, M. Serra; revising the manuscript critically for important intellectual content J.I. Anel, B. Domenech.

7.3. Category 3

Approval of the version of the manuscript to be published (the names of all authors must be listed):

J.I. Anel, P. Català, M. Serra, B. Domenech.

Declaration of Competing Interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

The authors would like to thank the reviewers for their valuable comments and time in reviewing previous versions of the article.

All persons who have made substantial contributions to the work reported in the manuscript (e.g., technical help, writing and editing assistance, general support), but who do not meet the criteria for authorship, are named in the Acknowledgements and have given us their written permission to be named. If we have not included an Acknowledgements, then that indicates that we have not received substantial contributions from non-authors.

References

 Abdeljaouad MA, Klement N. Tabu Search Algorithm for Single and Multi-model Line Balancing Problems. In: IFIP Advances in Information and Communication Technology, 630 IFIP; 2021. p. 409–15. https://doi.org/10.1007/978-3-030-85874-2_43.

- [2] Akpinar Ş, Baykasołlu A. Modeling and solving mixed-model assembly line balancing problem with setups. Part II: A multiple colony hybrid bees algorithm. Journal of Manufacturing Systems 2014;33(4):445–61. https://doi.org/10.1016/j. jmsy.2014.04.001.
- [3] Barathwaj N, Raja P, Gokulraj S. Optimization of assembly line balancing using genetic algorithm. Journal of Central South University 2015;22(10):3957–69. https://doi.org/10.1007/s11771-015-2940-9.
- [5] Çil ZA, Li Z, Mete S, Özceylan E. Mathematical model and bee algorithms for mixed-model assembly line balancing problem with physical human-robot collaboration. Applied Soft Computing Journal 2020;93:106394. https://doi.org/ 10.1016/j.asoc.2020.106394.
- [6] Eghtesadifard M, Khalifeh M, Khorram M. A systematic review of research themes and hot topics in assembly line balancing through the web of science within 1990–2017. Computers and Industrial Engineering 2020;139:106182. https://doi. org/10.1016/j.cie.2019.106182. February 2019.
- [7] García De Jesús, A. (2015). mOdul The first modular car for all. Retrieved from https://www.domestika.org/es/projects/342422-modul-the-first-modular-car-for -all.
- [8] Hazır Ö, Agi MAN, Guérin J. An efficient branch and bound algorithm for smoothing the workloads on simple assembly lines. International Journal of Production Research 2021;59(2):617–34. https://doi.org/10.1080/ 00207543.2019.1701208.
- [9] Kamal M, Martinez Lastr JL. Assembly Line Balancing and Sequencing. Assembly Line - Theory and Practice. 2011. https://doi.org/10.5772/19953.
- [10] Kim YJ, Kim YK, Cho Y. A heuristic-based genetic algorithm for workload smoothing in assembly lines. Computers and Operations Research 1998;25(2): 99–111. https://doi.org/10.1016/S0305-0548(97)00046-4.
- [12] Kucukkoc I, Zhang DZ. Mixed-model parallel two-sided assembly line balancing problem: A flexible agent-based ant colony optimization approach. Computers and Industrial Engineering 2016;97:58–72. https://doi.org/10.1016/j. cie.2016.04.001.
- [13] Liu X, Yang X, Lei M. Optimisation of mixed-model assembly line balancing problem under uncertain demand. Journal of Manufacturing Systems 2021;59: 214–27. https://doi.org/10.1016/j.jmsy.2021.02.019. September 2020.
- [14] Michels AS, Lopes TC, Magatão L. An exact method with decomposition techniques and combinatorial Benders' cuts for the type-2 multi-manned assembly line balancing problem. Operations Research Perspectives 2020;7:100163. https://doi. org/10.1016/j.orp.2020.100163. September.
- [15] Nazari A, Thiruvady D, Elmi A, Schneider JG. Simulated Annealing for Single and Mixed Model Assembly Line Balancing with Setups. In: 2020 IEEE Symposium Series on Computational Intelligence. SSCI; 2020. p. 2762–9. https://doi.org/ 10.1109/SSCI47803.2020.9308143. 2020.
- [16] Pinarbasi M, Alakas HM, Yuzukirmizi M. A constraint programming approach to type-2 assembly line balancing problem with assignment restrictions. Assembly Automation 2019;39(5):813–26. https://doi.org/10.1108/AA-12-2018-0262.
- [18] Rabbani M, Mousavi Z, Farrokhi-Asi H. Multi-objective metaheuristics for solving a type II robotic mixed-model assembly line balancing problem. Journal of Industrial and Production Engineering 2016. https://doi.org/10.1080/ 21681015.2015.1126656.
- [19] Ramezanian R, Ezzatpanah A. Modeling and solving multi-objective mixed-model assembly line balancing and worker assignment problem. Computers and Industrial Engineering 2015;87:74–80. https://doi.org/10.1016/j.cie.2015.04.017.
- [20] Simaria AS, Vilarinho PM. A genetic algorithm based approach to the mixed-model assembly line balancing problem of type II. Computers and Industrial Engineering 2004;47(4):391–407. https://doi.org/10.1016/j.cie.2004.09.001.
- [21] Sivasankaran P, Shahabudeen P. Literature review of assembly line balancing problems. International Journal of Advanced Manufacturing Technology 2014;73: 1665–94. https://doi.org/10.1007/s00170-014-5944-y.
- [22] Su P, Lu Y. Combining genetic algorithm and simulation for the mixed-model assembly line balancing problem. In: Proceedings - Third International Conference on Natural Computation, ICNC; 2007. https://doi.org/10.1109/ICNC.2007.306. 2007.
- [23] Wei NC, Chao IM. A solution procedure for type e simple assembly line balancing problem. Computers and Industrial Engineering 2011;61(3):824–30. https://doi. org/10.1016/j.cie.2011.05.015.
- [24] Zamzam N, Elakkad A. Time and space multi-manned assembly line balancing problem using genetic algorithm. Journal of Industrial Engineering and Management 2021;14(4):733. https://doi.org/10.3926/jiem.3542.
- [25] Zhao X, Hsu CY, Chang PC, Li L. A genetic algorithm for the multi-objective optimization of mixed-model assembly line based on the mental workload. Engineering Applications of Artificial Intelligence 2016;47:140–6. https://doi.org/ 10.1016/j.engappai.2015.03.005.