Int. J. Production Economics 145 (2013) 761–772

Contents lists available at ScienceDirect

Int. J. Production Economics

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

A robustness information and visualization model for time and space assembly line balancing under uncertain demand

Manuel Chica [a,*], Óscar Cordón [a,b], Sergio Damas [a], Joaquín Bautista [c]

[a] European Centre for Soft Computing, 33600 Mieres, Spain
[b] DECSAI and CITIC-UGR, University of Granada, 18071 Granada, Spain
[c] ETSEIB, Universitat Politècnica de Catalunya, 08028 Barcelona, Spain

ARTICLE INFO

Article history:
Received 8 October 2012
Accepted 29 May 2013
Available online 13 June 2013

Keywords:
Robust optimization
Visualization
Time and space assembly line balancing
Mixed products
Uncertain demand
Problem instance generator

ABSTRACT

The time and space assembly line balancing problem (TSALBP) is a realistic multiobjective version of assembly line balancing industrial problems involving the joint optimization of conflicting criteria such as the cycle time, the number of stations, and the area of these stations. However, the existing problem formulation does not consider the industrial scenario where the demand of a set of mixed products is variable and uncertain. In this work we propose to introduce novel robustness functions to measure how robust the line configuration is when the production plans demand changes. These functions are based on the stations overload under future demand conditions and are used as additional a posteriori information for the non-dominated solutions found by any multiobjective optimization method. The values of the robustness functions are put together with a novel graphical representation to form a generic model that aims to offer a better picture of the robustness of the set of Pareto-optimal solutions.

Real data from the assembly line and production planning of the Nissan plant of Barcelona is considered for the experimentation. This information is also employed to develop a new TSALBP instance generator (NTIGen) that can generate problem instances having industrial real-like features. The use of the robustness information model is illustrated in an experimentation formed by a set of instances generated by NTIGen. Results show how the use of this robustness information model is necessary for the decision maker as it allows her/him to discriminate between different assembly line configurations when future demand conditions vary.

1. Introduction

An assembly line is made up of a number of workstations, arranged either in series or in parallel. Since the manufacturing of a production item is divided into a set of tasks which require an operation time for their execution, a usual and difficult problem, called assembly line balancing (ALB), is to determine how these tasks can be assigned to the stations fulfilling certain restrictions such as precedence relations. The final aim of ALB is to get an optimal assignment of subsets of tasks to the stations of the plant (Boysen et al., 2007, 2008). An excellent review on ALB and the existing solving methods for the different problems is given in Battaïa and Dolgui (2013). Within ALB, a well-known family of problems is the simple assembly line balancing problem (SALBP) (Baybars, 1986; Scholl, 1999; Scholl and Becker, 2006). The SALBP only considers the assignment of each task to a single station in such a way that all the precedence constraints are satisfied and no station workload time is greater than the line cycle time.

As a result of the observation of the ALB operation in an automotive Nissan plant from Barcelona (Spain), Bautista and Pereira (2007) recently proposed a SALBP extension aiming to design a more realistic ALB model. They considered an additional space constraint to get a simplified but closer version to real-world situations, defining the time and space assembly line balancing problem (TSALBP). The TSALBP presents eight variants depending on three optimization criteria: m (the number of stations), c (the cycle time), and A (the area of the stations). In this paper we tackle the TSALBP-m/A variant[1] which tries to jointly minimize the number of stations and their area for a given product cycle time, a complex and realistic multicriteria problem in the automotive industry.

The multicriteria nature of the TSALBP-m/A (also known as TSALBP-1/3) favored the application of multiobjective metaheuristics (MOMHs) such as multiobjective ant colony optimization

* Corresponding author. Tel.: +34 985 456 545.
E-mail addresses: manuel.chica@softcomputing.es, mchserrano@gmail.com (M. Chica), oscar.cordon@softcomputing.es (Ó. Cordón), sergio.damas@softcomputing.es (S. Damas), joaquin.bautista@upc.edu (J. Bautista).

[1] Originally, this TSALBP variant is referred as TSALBP-1/3 (Bautista and Pereira, 2007). This new notation is introduced in this work for a better understanding.

0925-5273/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ijpe.2013.05.030

(MOACO) (Chica et al., 2010), evolutionary multiobjective optimization (EMO) (Chica et al., 2011), and memetic algorithms (MAs) (Chica et al., 2012). These MOMHs are able to return a set of non-dominated solutions for a known demand of homogeneous products. However, assembly lines are generally balanced for producing mixed products and their demand is not usually fixed and certain. When the assembly line is devoted to produce mixed products in a given sequence, the operation times of the required tasks are obtained from the average value of the different products and their demand. This is a problematic rough estimate of the actual operation times. If the demand changes, the operation times also change and a re-balancing could be necessary for the configuration. This re-balancing causes production losses as workers, assigned to a workstation, will have to comply with new tasks within the station. These workers must be trained in the development of the new tasks. This is normally a two weeks learning process in the Nissan plant at Barcelona.

These difficulties and associated efficiency losses are common in the automotive industry. This fact has encouraged us to propose a model for evaluating and analyzing the convenience of the solutions found by a multiobjective optimization (MOO) method when these future demand conditions have changed. Normally, a set of production plans are used to define the demand in future scenarios. Our proposed model is based on this set of real production plans by introducing the concept of robustness of a solution linked to the flexibility of an assembly line configuration to the demand changes.

Robustness can be applied to many components in an optimization process: noise in constraints, objective function, or uncertainties in data variables (Roy, 1998, 2010; Beyer and Sendhoff, 2007). Real-world applications, as ALB, normally involve uncertainties because of operating conditions or manufacturing process (Miettinen et al., 2008). In our case, the interest lies on measuring the robustness of a specific operating condition, i.e. the operation times originated by the mixed products demand. The goal is to identify how much robust the non-dominated solutions for the TSALBP-m/A are in a set of production plans. Three robustness functions are defined based on the overloaded stations and the overloading production plans which occur when the demand changes and the line configuration are set. The use of overloads in assembly lines is not new in industrial production but, to our knowledge, it is a novelty in robust balancing of assembly lines.

The latter robustness measures are used as additional a posteriori information associated to each non-dominated solution returned by the MOO method. We followed this design for our model because these robustness functions belong to a secondary importance level with respect to the TSALBP-m/A objectives. Therefore, the model might be seen as a hierarchical decision support system where the optimization objectives are the most important criteria and the robustness is an additional assessment.

Meanwhile, practitioners are requiring better and easier ways to understand the truly useful information to make their decisions. Visualizing the results of a multi-criteria decision making (MCDM) process is gaining importance and becoming a crucial part of a global framework: search, preference trade-offs, and interactive visualization (Bonissone et al., 2009). In our case and to facilitate the understanding of the robustness information, a novel graphical representation is firstly introduced in this work. This general-purpose representation both shows the objective values of the non-dominated solutions found and embeds the information provided by the robustness functions. Therefore, the robust assembly line configuration options for the problem are depicted at a glance.

A real-like Nissan TSALBP instance generator software (NTIGen) is also described in order to validate the robustness model in a diverse set of TSALBP instances and production plans. The design and implementation of NTIGen is done by using the real data and industrial features of the Nissan industry plant of Barcelona. The software is freely available on-line to be used for future research works. A set of eight instances are used in our experimentation where the robustness function results and the novel graphical representation are computed and shown for the non-dominated solutions returned by a specific MOO method, the *advanced TSALBP-NSGA-II* (Chica et al., 2011).

The rest of the paper is structured as follows. In Section 2, the TSALBP-m/A formulation and the uncertain demand scenario modeled by production plans are explained. The numerical robustness functions for assembly line balancing are given in Section 3. The robustness information model and its novel graphical representation are introduced in Section 4. The description of the NTIGen software is shown in Section 5. The experimentation results are discussed in Section 6. Finally, we present some concluding remarks in Section 7.

2. Demand variation in the TSALBP-m/A for mixed product products

We first introduce the TSALBP-m/A (Section 2.1) and then the real scenario of having a mixed products with changing demand (Section 2.2).

2.1. Time and space assembly line balancing problem

The manufacturing of a production item is divided into a set J of n tasks. Each task j requires an operation time for its execution $t_j > 0$ that is determined as a function of the manuf acturing technologies and the employed resources. Each station k ($k = 1, 2, \ldots, m$) is assigned to a subset of tasks S_k ($S_k \subseteq J$), called workload. Each task j can only be assigned to a single station k.

Each task j has a set of direct "preceding tasks" P_j which must be accomplished before starting it. These constraints are normally represented by means of an acyclic precedence graph, whose vertices stand for the tasks and where a directed arc (i, j) indicates that task i must be finished before starting task j on the production line. Thus, task j cannot be assigned to a station that is ordered before the one where task i was assigned. Each station k also presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' processing time assigned to the station k. SALBP focuses on grouping tasks in workstations by an efficient and coherent way.

In this simplistic model there is a need of introducing space constraints in assembly lines' design based on two main reasons: (a) the length of the workstation is limited in the majority of the situations and (b) the required tools and components to be assembled should be distributed along the sides of the line. Hence, an area constraint may be considered by associating a required area a_j to each task j and an available area A_k to each station k that, for the sake of simplicity, we shall assume it to be identical for every station and equal to $A = \max_{k=1,2,\ldots,m} A_k$. Thus, each station k requires a station area $a(S_k)$ that is equal to the sum of areas required by the tasks assigned to station k.

This leads us to a new family of problems called TSALBP (Bautista and Pereira, 2007). It may be stated as: given a set of n tasks with their temporal t_j and spatial a_j attributes ($1 \le j \le n$) and a precedence graph, each task must be assigned to a single station such that: (i) every precedence constraint is satisfied, (ii) no station workload time ($t(S_k)$) is greater than the cycle time (c), and (iii) no area required by any station ($a(S_k)$) is greater than the available area per station (A).

TSALBP presents eight variants depending on three optimization criteria: m (the number of stations), c (the cycle time) and A (the area of the stations). Within these variants there are four

multiobjective problems and we will tackle one of them, the TSALBP-m/A. It consists of minimizing the number of stations m and the station area A, given a fixed value of the cycle time c, mathematically formulated as follows:

$$f^0(x) = m = \sum_{k=1}^{UB_m} \max_{j=1,2,\ldots,n} x_{jk}, \quad (1)$$

$$f^1(x) = A = \max_{k=1,2,\ldots,UB_m} \sum_{j=1}^{n} a_j x_{jk}, \quad (2)$$

where UB_m is the upper bound for the number of stations m, a_j is the area information for task j, x_{jk} is a decision variable taking value 1 if task j is assigned to station k, and n is the number of tasks.

We chose this variant because it is realistic in the automotive industry since the annual production of an industrial plant (and therefore, the cycle time c) is usually set by some market objectives. For more information about the problem we refer the interested reader to Chica et al. (2010, 2012).

2.2. Production plans for modeling changing demand

The latter TSALBP-m/A formulation assumes both a constant demand and fixed operation times t_j. However, real assembly lines are normally employed to assemble more than one single product, and when the demand of each product changes, the operation times of the tasks change in consequence. The demand of a set of mixed products is defined by means of production plans. In this work, the engine assembly line of the Nissan Spanish Industrial Operations (NSIO) plant is the chosen uncertain environment to define the different production plans.

Nine different engines are assembled in the main line of the NSIO plant, m_1,…, m_9, having different destinations and assembly characteristics. The first three engine products are built for 4×4 vehicles; products m_4 and m_5 are for VANs; and the remaining four products are used by medium tonnage trucks. When demand is balanced (identical for the nine products) and the cycle time is 3 min, the assembly line is divided into 21 workstations having an average length A_k of 4 m each.

In Bautista and Pereira (2007), authors grouped the primary operations of this assembly line in the so-called Nissan TSALBP instance having 140 tasks. For each type of engine, operation times change. In Table 1 the operation times of five tasks are listed for illustration. The average operation time when having a balanced demand for the nine products is also shown in the *t-average* column.

Of course, it is difficult to always have the same uniform demand for all the engines within a global demand. Although the line is supposed to have a fixed daily production of, for instance, 270 products, the line should be capable of producing the required products for the specific product demand of a given production plan. In other words, the production plan of the 270 engines is not constant. Then, the goal is to have an assembly line configuration that is robust enough for different production plans.

Table 1
Operation times and average time for five tasks belonging to the NSIO engine assembly line.

Task	m_1	m_2	m_3	m_4	m_5	m_6	m_7	m_8	m_9	t-average
1	64.8	61.2	60	54	58.8	55.2	63	66	57	60
3	18.4	18	20	19.6	19	21.6	21	20.4	22	20
5	19	19.6	18.4	20	21	20.4	18	21.6	22	20
8	9.8	9	10.5	10.8	9.5	11	9.2	10	10.2	10
9	20	19.6	19	18	20.4	18.4	21.6	21	22	20

Table 2
Production units of the engine models for each production plan.

Family	Product	Production plans						
		# 1	#2	# 3	# 6	# 9	# 12	# 18
4 x 4	m_1	30	30	10	50	70	24	60
	m_2	30	30	10	50	70	23	60
	m_3	30	30	10	50	70	23	60
VAN	m_4	30	45	60	30	15	45	30
	m_5	30	45	60	30	15	45	30
Trucks	m_6	30	23	30	15	8	28	8
	m_7	30	23	30	15	8	28	8
	m_8	30	22	30	15	7	27	7
	m_9	30	22	30	15	7	27	7

There are currently 23 production plans for the nine engines and one working day at the NSIO. Each program corresponds to a set of operation times biased by the demand of each of the nine products. We summarize here the characteristics of each of the 23 production plans. We have grouped them into seven categories according to the type of engine demand. One representative production plan is selected for each category to be used in the computational experimentation developed in Section 6. As said, the total number of engines assembled in a working day is 270 in two shifts:

Cat-1 (plan #1): identical demand for each of the nine products (balanced demand) (30 products per product).
Cat-2 (plan #2): identical demand for each of the three engine families: 4×4, VAN, and trucks (90 per product family).
Cat-3 (plan #3): one of the engine families has low demand while the demand of the other two families is high and identical.
Cat-4 (plan #6): one of the engine families has high demand while the demand of the other two families is medium and identical.
Cat-5 (plan #9): one of the engine families has high demand while the demand of the other two families is low and identical.
Cat-6 (plan #12): the demand of the engine families follows an arithmetic progression.
Cat-7 (plan #18): the demand of the engine families follows a geometric progression.

The seven representative production plans, one per category, are shown in Table 2. Definitely, the demand variation of the production plan for mixed products conditions the average operation times of the 140 assembly line tasks. In that case, a re-balancing of the assembly line could be necessary. For example, task 1 has operation times of 64.8 s, 61.2 s, 60 s, 54 s, 58.8 s, 55.2 s, 63 s, 66 s, and 57 s for products m_1–m_9, respectively. On the other hand, production plan #12 has a demand of 24, 23, 23, 45, 45, 28, 28, 27, and 27 products for each of the engine products. Consequently, the average time for task 1 in the latter plan is 59.44 s$(= (64.8 \times 24 + 61.2 \times 23 + \cdots + 66 \times 27 + 57 \times 27)/270)$ in comparison with the 60 s needed by production plan #1.

The selected representative production plans are used in this work to present additional information to the decision maker (DM) about how robust a new assembly line configuration is under demand changes, i.e. how good it is with respect to those changes.

3. Robust solutions for assembly line balancing when demand is uncertain

In Sections 3.1 and 3.2 we provide a brief review of the outstanding proposals in generic robust optimization as well as

for the specific application to ALB. Then, in Section 3.3, we introduce two kinds of robustness functions to be applied to the assembly line configurations in order to know how the configuration behaves when demand changes.

3.1. Robust optimization in EMO and production

The search for optimal robust designs often appears as a MCDM problem optimizing conditional expectation and variance. For example, one of the proposals in this line is the multiobjective six sigma (DFMOSS) by Shimoyama et al. (2005). In Lim et al. (2006) a priori information is used to specify the desired robustness of the final design through a multiobjective evolutionary algorithm with good nominal performance and maximal robustness.

The work of Deb and Gupta (2006) is the first and one of the most important contributions in introducing robustness in EMO. The authors define a robust solution as one which is less sensitive to the perturbation of the decision variables in its neighborhood. In MOO problems, this insensitivity must be shown for the non-dominated solutions with respect to all the objectives and must be checked for all the Pareto-optimal solutions. Using this concept, Deb and Gupta (2006) suggest two types of multiobjective robust solutions: type I and type II. These two types can be seen as the two major approaches when dealing with robustness (Ferreira et al., 2008): (a) expectation measure, where the original objective function is replaced by a metric of expectation and performance of the vicinity and (b) variance measure, where an additional criterion is appended to the objective function to account for the deviation of the latter around the vicinity of the design point.

There are also works in production and design problems where some of the parameters of the problem are uncertain or depend on future actions (Scheffermann et al., 2009; Tan et al., 2007; Ong et al., 2006). An example of a robust optimization model for a multi-site production planning problem is developed in Leung et al. (2007). In this work, the authors assume a future economic scenario with an associated probability. An optimal production plan less sensitive to the change in the noisy and uncertain data is given by a stochastic non-linear programming model.

3.2. Robust optimization for assembly line balancing

The most related work to our problem is the robust optimization approach for ALB proposed by Xu and Xiao (2010, 2011). They deal with the mixed ALB problem where the exact quantity of products to be manufactured is unknown. The objective is to minimize the workload variance over all the stations in the line. For that goal two ways of solving the problem are provided: by using a min–max indicator which minimizes the maximum workload variance among all the input data scenarios and by considering a α-worst approach. The authors claim to be the first ones to propose a α-worst scenario-based robust criteria and to apply it for ALB. This criteria can generate flexible robust solutions as there is a permitted tolerance threshold for each solution. A basic genetic algorithm is presented where the objective function to be minimized is either min–max or α-worst scenario criterion. As we will explain later in Section 4, one of the main differences of our contribution with the latter one is not to include the robustness criterion within the search process.

Another way of considering uncertainty in ALB is by assuming that task times are uncertain and not deterministic. In Gurevsky et al. (2012) authors deal with the SALBP-E when having variable task processing times and propose a way to find a compromise between minimizing the objective function and a stability ratio for the solutions. A related stability study is done in Gurevsky et al. (2013) but for the case of the GALBP (a problem where each workstation is equipped with blocks).

In recent works, Dolgui and Kovalev (2012) propose an ALB model with uncertain operation execution times. Operation execution times are uncertain in the sense that their sets belong to a given set of scenarios. The difference with the TSALBP formulation is that task time uncertainty is modeled by upper and lower bounds associated to a specific station. Following this research line, Hazır and Dolgui (2013) have recently presented two robust SALBP-2 models which present interval uncertainty for operation times. A decomposition based algorithm is developed and combined with enhancement strategies to solve both problem models.

Finally, we should remark an existing genetic algorithm for a bi-criteria ALB problem which considers flexible operation times (Hamta et al., 2011). The used meta-heuristic is single-objective and makes use of a weighted combination for both objectives. The authors use the traditional SALBP formulation and create ranges of tasks' processing times by adding four units as the upper bound value.

3.3. A proposal to evaluate the robustness of an assembly line configuration

Solving the TSALBP when the mixed products demand is uncertain belongs to the robust optimization case where the operating conditions change after the optimal solution is found (Ferreira et al., 2008). In our case, the operating conditions are the operation times originated by the different mixed products demands represented by the production plans of Section 2.2. The overall goal is to find a set of non-dominated solutions for the TSALBP-m/A and calculate their robustness for all the possible production plans. In the next paragraphs we will present our proposal for evaluating this robustness.

Let E be the set of possible production plans based on the demand variation and ε^0 a reference production plan, our evaluation proposal is based on determining the workload of the set K of stations of an assembly line configuration in the plans of E.

First, being S_k^0 the tasks assignment to the station k in ε^0 (normally, the balanced plan), the workload of this station k is obtained for all the production plans $\varepsilon \in E$: $t(S_k^0, \varepsilon)$.

Then, the relative station overloads with respect to the available cycle time c are calculated for all the existing production plans ε by applying Eq. (3)

$$\omega(S_k^0, \varepsilon) = \frac{\max\{0, t(S_k^0, \varepsilon) - c\}}{c} \quad \forall k \in K, \ \forall \varepsilon \in E. \tag{3}$$

From these overload values, the average and maximum station overload values are also calculated through Eqs. (4) and (5)

$$\overline{\omega}(S_k^0) = \overline{\omega}_k = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \omega(S_k^0, \varepsilon) \quad \forall k \in K, \tag{4}$$

$$\omega_{max}(S_k^0) = \max_{\varepsilon \in E} \{\omega(S_k^0, \varepsilon)\} \quad \forall k \in K. \tag{5}$$

Analogously, the average and maximum overloading values for each production plan are obtained by applying Eqs. (6) and (7)

$$\overline{\omega}(\varepsilon) = \frac{1}{|K|} \sum_{k=1}^{|K|} \omega(S_k^0, \varepsilon) \quad \forall \varepsilon \in E, \tag{6}$$

$$\omega_{max}(\varepsilon) = \max_{k \in K} \{\omega(S_k^0, \varepsilon)\} \quad \forall \varepsilon \in E. \tag{7}$$

The latter values allow us to define and calculate the proposed robustness functions. We can distinguish two types: (a) based on the overload size (Section 3.3.1) and (b) based on the number of overloaded stations (Section 3.3.2).

3.3.1. Robustness functions based on the overload size

The robustness of an assembly line configuration can be measured by the overload size of the stations in the configuration in all the production plans. Higher station overload sizes mean less robust configurations. We propose two functions. The first one, R_1, considers the average overload of all the stations for all the production plans

$$R_1 = f(\overline{\omega}) = \frac{1}{|K|}\sum_{k=1}^{|K|}\overline{\omega}(S_k^0) = \frac{1}{|E|}\sum_{\varepsilon=1}^{|E|}\overline{\omega}(\varepsilon). \quad (8)$$

The second function, R_2, reflects the maximum overload value in the stations of the configuration and the defined production plans. It could be seen as the worst possible scenario within the set of production plans

$$R_2 = f(\omega_{max}) = \max_{k\in K,\varepsilon\in E}\{\omega(S_k^0,\varepsilon)\} = \max\{\omega_{max}(S_k^0),\omega_{max}(\varepsilon)\}. \quad (9)$$

3.3.2. Robustness functions based on the number of overloaded stations

Another way of determining the robustness of an assembly line configuration is by counting the number of overloaded stations and/or the number of overloading production plans. Given a station $k\in K$, a production plan $\varepsilon\in E$, a configuration line $(S_1^0, S_2^0, \ldots, S_m^0)$ for a reference production plan ε^0, and a cycle time c, we can state that there is an overload in $(k, S_k^0, \varepsilon, c)$ iff

$$t(S_k^0,\varepsilon) > c \Leftrightarrow \omega(S_k^0,\varepsilon) > 0 \quad \forall k\in K,\ \forall\varepsilon\in E. \quad (10)$$

Associated to the concept of overload, the sets of overloaded stations for a plan, overloading production plans, and total overloads are respectively defined as follows:

$$D(\varepsilon) = \{\forall k\in K|\omega(S_k^0,\varepsilon) > 0\} \quad \forall\varepsilon\in E, \quad (11)$$

$$D(S_k^0) = \{\forall\varepsilon\in E|\omega(S_k^0,\varepsilon) > 0\} \quad \forall k\in K, \quad (12)$$

$$D = \{\forall\varepsilon\in E\wedge\forall k\in K|\omega(S_k^0,\varepsilon) > 0\}. \quad (13)$$

Finally, taking into account sets $D(\varepsilon)$, $D(S_k^0)$, and D, three functions are defined as follows. The third one (Eq. (16)) will be the third robustness function to be used in this study.

- Overloaded stations rate

$$\delta(\varepsilon) = \frac{|D(\varepsilon)|}{|K|}\in[0,1] \quad \forall\varepsilon\in E. \quad (14)$$

- Overloading plans rate

$$\delta(S_k^0) = \frac{|D(S_k^0)|}{|E|}\in[0,1] \quad \forall k\in K. \quad (15)$$

- Total number of overloads rate

$$R_3 = \delta = \frac{|D|}{|E||K|}\in[0,1] \quad (16)$$

3.3.3. An illustrative example

Table 3 shows an example of five stations from an assembly line in four different production plans. Average and maximum overload values, $\overline{\omega}$ and $\omega_{\max}$, are included in the table. It can be observed that all the production plans overload at least one of the stations. The second station is more robust than the rest since it is never overloaded. The first and second proposed robustness functions, R_1 and R_2, which face the average and maximum overload values in all the stations and production plans, take value 1.17 and 4.2, respectively. Overloaded and overloading rates highlight how many stations and production plans are overloaded and overload this line, respectively. The third robustness fun ction is also computed from the values in the table: $R_3 = \delta = 9/20 = 0.45$.

Table 3
Overload values using time units and robustness functions for five stations of a configuration line when having four different production plans.

Plans	Stations					Overload values		
	k_1	k_2	k_3	k_4	k_5	$\overline{\omega}(\varepsilon)$	$\omega_{\max}(\varepsilon)$	$\delta(\varepsilon)$
Plan #1	2	0	0	0	0	0.4	2	0.2
Plan #2	1	0	1.5	6.3	0	1.76	6.3	0.6
Plan #3	3	0	0	0	1.4	0.88	3	0.4
Plan #4	2	0	0	2	4.2	1.64	4.2	0.6
Overload values								
$\overline{\omega}(S_k^0)$	2	0	0.375	2.075	1.4			
$\omega_{\max}(S_k^0)$	3	0	1.5	6.3	4.2			
$\delta(S_k^0)$	0.4	0	0.25	0.5	0.5			

4. Visualization model to include robustness information for the non-dominated solutions

In all the works reviewed in Section 3, robustness is always included as a part of the search process. Unlike such previous works, the approach followed in this study is not to embed this robustness information into the search but to use it as a posteriori information when the MOO method has finished its run and has found a set of non-dominated solutions. Our approach has the advantage of not increasing the computational costs derived from the solution neighborhood calculation as well as its independence with respect to the MOO method used.

4.1. Using the robustness information for the TSALBP-m/A

As the inclusion of the robustness information model is done a posteriori, any of the existing MOMHs to solve the TSALBP-m/A (Chica et al., 2011, 2012) and even future methods can be used to illustrate the behavior of our proposal. The chosen MOMH is devoted to find and present a set of non-dominated solutions to the DM. Robustness functions R_1, R_2, and R_3 are calculated for all the non-dominated solutions, offering a ranking of the most robust solutions for the problem among those included in the obtained Pareto set approximation.

However, the set of non-dominated solutions is normally large and the application of the robustness functions implies a list of numerical values for R_1, R_2, and R_3 which could be unmanageable. It is already known that selecting a solution from a long list of objective vectors is complicated for human beings (Larichev, 1992; Benson and Sayin, 1997). Generally in EMO, attainment surfaces (Knowles, 2005; López-Ibáñez et al., 2010) and even more advanced graphical tools have been proposed to offer a better understanding of the Pareto front quality assessment, sometimes more useful than numerical values (Blasco et al., 2008; Lotov and Miettinen, 2008; Obayashi and Sasaki, 2003).

Besides a list of robustness function values, we propose the introduction of the robustness information in the graphical representation of the Pareto front approximation by means of robustness attainment surfaces. These robustness attainment surfaces can represent more than just one function value. In the case of the TSALBP-m/A, we have included R_2 and R_3 in the representation. We did not additionally include R_1 in order to avoid saturating the DM with excessive information. Moreover, R_1 is less discriminative than the others. Beyond the particular robustness functions considered, the goal of our visualization model is not to include

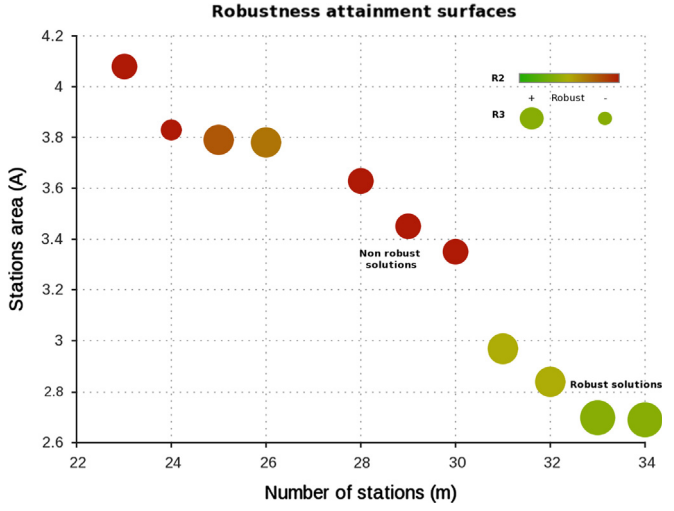


Fig. 1. Example of a robustness attainment surface where each non-dominated solution is represented by a point. The robustness information is encoded by the diameter of the point and its color.

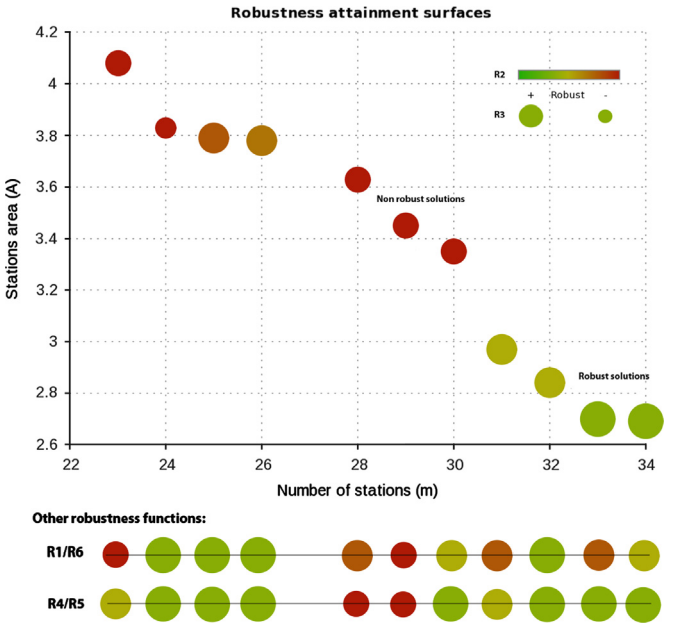


Fig. 2. Example of the scalability of a robustness attainment surface when having six different robustness functions.

every possible robustness function but only the real valuable information for the DM. The graph of Fig. 1 shows an example with a set of 14 non-dominated solutions and their robustness information for a given problem instance.

Each non-dominated solution is represented by a circle, whose diameter is proportional to the robustness value given by R_3, and the color illustrates the value of R_2 (green: low overload value and good robustness; red: high overload value and poor robustness). This way of using graphs associated to valuable MCDM information is done in many fields as scientific information analysis (Vargas-Quesada and de Moya-Anegón, 2007) or description of gene expression profiles in bioinformatics (Romero Zaliz et al., 2008).

We can thus see that such a graphical representation of robustness is complementary to the list of R values. In fact, when the number of non-dominated solutions increases, the graphical representation of the robustness is more interpretable for the DM. In addition, it is also more informative as spatial information can be analyzed when using this kind of representation since the DM is able to discover robust Pareto front regions where all the non-dominated solutions are robust, or non-robust regions where their solutions are not (see the example in Fig. 1).

4.2. Scalability issues of the model

Our proposed graphical representation model is generic and can be applied not just to the TSALBP but to any MOO problem in which presenting the robustness of a set of non-dominated solutions is an added value for the final decision of the DM. In that sense, there is a chance to design a representation software package in which the user can customize her/his representation.

Concretely, the DM will be allowed to perform the following actions before launching the MOMH for the specific problem in order to adjust the settings of the final graphic:

- Selecting the desired robustness functions: From a set of available robustness functions, the DM is able to select the robustness measures to be integrated in the visualization model. This is the same operation we performed in the current contribution. We just selected two of the three available robustness functions.
- Representing more than two robustness functions: If the DM desires to represent more than two robustness functions in the visualization model the tool will provide a way to show as many as she/he likes. An example of graph depicting six different robustness functions is shown in Fig. 2. In this graph, four functions (R_1, R_4, R_5 and R_6) are horizontally represented in pairs below the X-axis.
- Zooming in dense solution areas: In some problems, many solutions with similar objective values might appear. The DM can zoom in the area and inspect the solutions found by the MOMH. In addition, if there are more than one unique solution for the same objective values the user will be able to navigate through them.

5. NTIGen: a Nissan TSALBP instance generator software

5.1. Justification and basics of NTIGen

The main goal of the NTIGen software is to create real-like TSALBP instances with different features to serve as a benchmark for showing our robustness approach and for any future research work. Although there are ALB instances available on-line and even a SALBP instance generator (Otto et al., 2011), there is no any existing source where TSALBP instances can be generated and referred.

Assembly lines in the automotive industry present a set of industrial features which condition the task and graph distribution of the problem instance. The user must be allowed to incorporate these industrial real-like features to the generated instances and these instances should be similar to the original Nissan instance context (Chica et al., 2012). Concretely, the developed NTIGen software includes the following features, which are illustrated in Fig. 3:

- Checkpoints: They are assembly line points in which workers test the quality and completeness of a set of operations previously finished. If we consider these checkpoints as new tasks, the representation of a checkpoint in an assembly line graph is given by a task having a high number of preceding tasks (for instance, task 11 in Fig. 3).
- Tasks without precedences: In real industrial scenarios, such tasks are justified if there are operations unconditioned by

other operations. They are commonly found in the engine and trim lines of the car manufacturing. In Fig. 3, tasks 1, 3, 8, 7, and 10 have no precedences.

- Final tasks: Tasks in an assembly line which are associated to the most external and final operations of the product. They are represented as tasks with no successors in the precedence graph (tasks 12–14 in Fig. 3).
- Isolated tasks: They can be performed at any part of the assembly of an item. An example of these kinds of tasks are those related with additional parts of a product which can be incorporated to the global product at any station. Task 4 in Fig. 3 is an isolated task as it has no precedence relations.
- Operations aggregation: This process comes up when some operations need the same tools or are done by the same worker. In this case, several tasks of the same stage are put together in just one task.
- Operations breaking up: If possible, it is used in the industrial context to detail the implementation of an operation in different operating tasks. It is useful for balancing an assembly line when the cycle time is reduced.

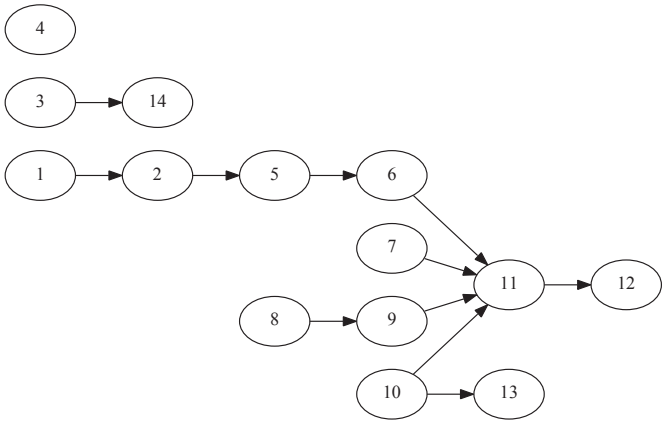


Fig. 3. A precedence graph with 13 tasks showing examples of different kinds of tasks in an industrial context: chains of tasks, initial and final tasks, isolated tasks, and checkpoints.

- Chains of tasks: They appear when there are strongly linked operations, normally in the same station or stage. A chain of tasks represents natural sequences of operations within the assembly process (see tasks 1, 2, 5, and 6 in Fig. 3).

5.2. Tuneable parameters of the generator

The features introduced in the previous section can be parameterized by the NTIGen user to generate a customizable instance. NTIGen is also fed by a set of stages with some initial tasks. By default, these stages and tasks correspond to the original Nissan instance with 140 tasks and 21 workstations (Chica et al., 2012) although they can be modified by the user before launching the application. The user can set all the desired features by changing the parameters of an XML file (see Fig. 4). The most important input parameters are the following:

Number of tasks (n): This is an important parameter of the instance that enormously conditions its complexity. From the initial set of tasks, new operating tasks are generated by breaking up them until reaching the user needs. If we need less tasks than the original ones, they are merged at random. The new generated tasks are required to belong to the same or close stages than their original ones.

Processing times (t_j): The processing time of each task t_j is randomly disrupted by a normal distribution within a user-defined interval. When creating or merging tasks, the processing times for the resulting tasks are reduced or duplicated, respectively. This is done to maintain the original situation of the Nissan instance.

Production plans: The production plans are always set to the NSIO original plans, described in Section 2.2. The processing times of the tasks for the different engine products are created by randomly modifying the original processing time t_j within the range $[0.9\ t_j, 1.1t_j]$.

Cycle time (c): It is also disrupted independently from the processing times of the tasks. As done with t_j, the disruption is created within a user-defined interval. In our case, the new cycle time is set to a value within $[0.75c, 1.25c]$.

Required operation area (a_j): Task areas are specified by two-dimensional units, i.e. length (a_j) and width (b_j). The first

```
<?xml version="1.0" ?>

<TSALBP-GEN>

        <!-- Settings for generating real-world TSALBP instance-->
        <name> nissan_320 </name>

        <!-- general values as seed or number of desired tasks -->
        <seed> 56399 </seed>
        <noTasks> 320 </noTasks>

        <!-- values for generating operating times -->
        <lowerTimeFactor> 0.90 </lowerTimeFactor>
        <upperTimeFactor> 1.10 </upperTimeFactor>

        <!-- percentage of initial (no predecessors) and final tasks (no successors) -->
        <initialTasks> 0.1 </initialTasks>
        <finalTasks> 0.1 </finalTasks>

        <!-- tasks which are checkpoints (many successors and predecessors) -->
        <noOfCheckPoints minPreds="5" maxPreds="10"> 5 </noOfCheckPoints>

        <!-- tasks which have not got any precedence relation -->
        <isolatedTasks> 5 </isolatedTasks>

        <!-- list of possible widths for areas -->
        <areaWidths>
                <width value="0.5"/>
                <width value="0.75"/>
                <width value="1"/>
                <width value="1.5"/>
                <width value="1.75"/>
```

Fig. 4. An XML configuration file to set the input parameters of the NTIGen software.

Table 4
Main characteristics of the generated TSALBP problem instances.

Features	NTIGen instances							
	P1	**P2**†	P3	P4	P5	P6	P7	P8
Random seed	24 151	N/A	117 017	21 277	113 683	56 399	5869	73 553
No. of tasks	100	140	190	220	280	320	376	420
Cycle time	199.97	180	207.07	222.42	221.62	169.552	186.65	137.751
OS	0.5	0.9	0.7	0.5	0.3	0.6	0.25	0.95
Precedences	156	293	314	304	407	435	548	608
Precs. window	5	N/A	5	1	2	1	3	2
TV	35.95	24	41.75	151.45	224.29	2742.28	901.34	1003.77
AV	500	513.86	266.67	300	400	200	300	133.33
Initial tasks	14	1	6	33	59	32	87	6
Final tasks	8	5	7	20	42	31	49	8
Isolated tasks	2	0	5	3	0	5	0	3
Checkpoints	3	N/A	0	6	7	1	12	0

† Original NISSAN instance.

dimension, a_j, is the truly useful variable for the TSALBP optimization. In the original instance, b_j is always set to one distance unit. To generate a new instance, the squared area of each task is always maintained by the generator but b_j is randomly changed to a set value. In our case, the set is given by {0.5, 0.75,…,2.25}. This set of possible b_j values can be modified by the user of the NTIGen software. Therefore, the length of each task a_j, used for the optimization, is different for each generated TSALBP instance. As done with the processing times, a_j is reduced or duplicated when increasing or decreasing the number of tasks to try to maintain the original Nissan situation.

Apart from the operating tasks and their corresponding processing times and areas, NTIGen generates the precedence graph of the instance. These precedence relations are created between tasks of the same stage (generating chains) or different stages within a maximum window, set by the user, in order to link tasks which are industrially close. The minimum and maximum number of preceding tasks for a checkpoint in a problem instance can be set prior the instance generation. The same definition can be done for the number of initial, final, and isolated tasks.

NTIGen creates precedence relations until it reaches the required complexity of the graph which is another important feature of an ALB instance (Bhattacharjee and Sahu, 1990). This complexity of the precedence graph is also a user parameter and it is measured by the order strength (OS) of the graph (Dar-El, 1975). The OS is calculated from the graph in transitive closure. The transitive closure of a set of direct precedences E is given by $E^T = \{(i,j)|i \in V, j \in F_i^T\}$, with V being the set of nodes and F_i^T the set of indirect successors of the task i. The OS represents the number of ordering relations of the graph in a transitive closure with respect to all possible ordering relations (Eq. 17)

$$OS = \frac{|E^T|}{\frac{n(n-1)}{2}}. \quad (17)$$

The OS varies between [0,1]. If OS is equal to 0 the instance has no precedence relations but if OS takes value 1, there is just one feasible sequence of tasks.

The result after running the NTIGen software is a structured text file describing the generated instance with the list of tasks, their operating times and area, and their precedence relations. The precedence relations form the transitive reduction of the graph in order to minimize computational resources.

In addition, by changing the number of tasks, their processing time and area we can generate instances having different time variability (TV) and area variability (AV). Descriptors about the generated instance are listed after its creation to show the complexity of the graph, TV, AV, and the number of checkpoints, isolated, initial, and final tasks.

Table 5
Used parameter values for the *advanced TSALBP-NSGA-II*.

Parameter	Value	Parameter	Value
Random seed	1212	Stopping criteria	300 s
Population size	100	Ishibuchi's similarity based mating γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
α values for scramble mutation	{0, 0.8}		

5.3. Description of the used TSALBP instances

By using the NTIGen software, a set of eight new TSALBP real-like instances have been created to be used in this study. The features of these real-like instances are shown in Table 4. The NTIGen software and this set of TSALBP instances are publicly available at ⟨http://www.prothius.com/TSALBP⟩.

Notice that, the number of precedences in Table 4 has been calculated from the transitive reduced graph. Besides, the random seed numbers for the pseudo-random generator have been randomly obtained from the list of the first $2^{17}-1$ prime numbers in order to ensure that all instances are reproducible.

6. Experiments and analysis of the robustness results

In this section we present the results of the experimentation and the analysis of them. The goal is to show how robustness functions are used a posteriori for providing an important additional information about the convenience of selecting some non-dominated solutions for the TSALBP-m/A instead of others according to their robustness.

To generate the non-dominated solutions considered in the experiments we have selected the *advanced TSALBP-NSGA-II* (Chica et al., 2011) as MOO method. The *advanced TSALBP-NSGA-II* will generate the non-dominated solution sets for all the TSALBP instances described in Section 5.3 and the production plans of Section 2.2 when demand changes. The parameters of the algorithm are presented in Table 5. We would also like to remark that the complete TSALBP framework is available on-line at ⟨http://www.prothius.com/TSALBP⟩ for ensuring the reproducibility of the experimentation.

The obtained results and the robustness attainment surfaces of the non-dominated solutions are presented in Figs. 5–10. The figures collect the number of non-dominated solutions (cardinality), a table with the values of R_1, R_2, and R_3, which denote the robustness of each solution in the Nissan production plans, and the robustness attainment surfaces.

In these robustness attainment surfaces, function R_2 is symbolized by the green-red color map (a solution will have a red colored point when R_2 is over 3% of the total cycle time although this value can be changed). R_3 is represented by the diameter of each non-dominated solution. Then, smaller and reddish points mean lower robustness. As explained in Section 4, the graphical information of the model is complementary to the robustness functions and they both constitute the proposed model. In this section we will show how a valuable analysis can be derived from it.

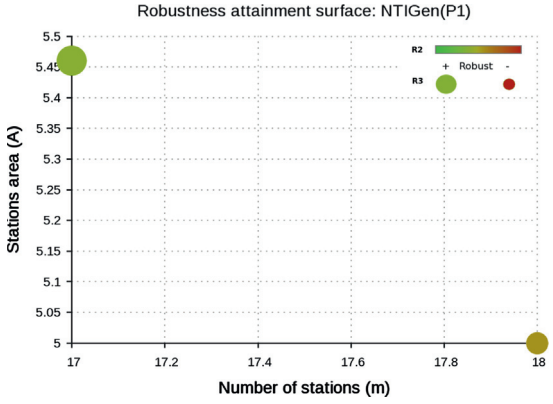
The first instance tackled by the algorithm is P1 (Fig. 5) where only two non-dominated solutions are found. However, even when having small non-dominated solution sets, the robustness information is important for the DM. Solution #2 reports robustness values of $R_1=0.149$, $R_2=3.33$, and $R_3=0.12$. These values mean that, when demand varies, the assembly line should support an average station overload of 0.149 time units (R_1), 3.33 time units in the most overloaded station (R_2), and that a 12% of the stations are overloaded (R_3). Solution #2 is thus less robust than solution #1. Then, if a DM can afford stations with an area of 5.4, solution #1 shall be the best option.

P2 is the original Nissan instance having 140 tasks. The number of non-dominated solutions obtained by the algorithm is five (see Fig. 6). The graphical points and the numerical values of R_1, R_2, and R_3 allow us to conclude that solutions #1 and #2, those with objective values (19,5) and (18,6.09), respectively, are less robust than the remainder when demand changes. The DM is able to rapidly infer from the attainment surface of Fig. 6 that, if the number of stations (and then, workers) is not restricted, the best approach in terms of robustness is always choosing a solution with more than 19 stations.

Instances P3 and P5 are the cases in which the most homogeneous and robust solutions are found. In both instances, almost all the solutions seem to be robust enough for the production plans. There are even some solutions with no station overload ($R_1=R_2=R_3=0$). As these instances do not provide any difference with respect to their robustness values, we have not included their graphical representations in the section to focus our analysis just on those instances presenting a higher robustness variability.

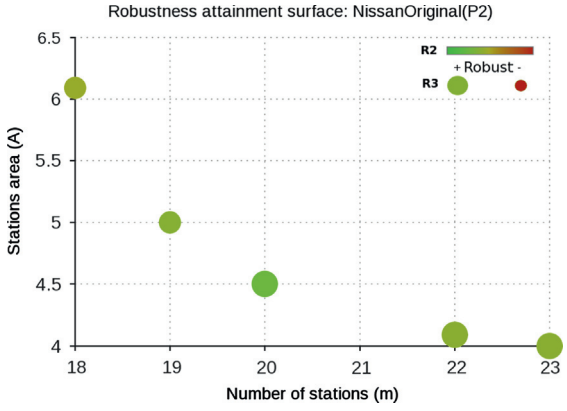
On the contrary, the rest of the problem instances present important robustness differences among the solutions, thus showing the importance of our proposed model. For example, there are seven non-dominated solutions for instance P4 (Fig. 7). Among them, there are two solutions which are less robust than the remainder. These are solutions #1 and #2 with 15 and 16 stations, respectively. In the latter pair of solutions, the 11% and 5.2% of the stations are overloaded by different production plans (R_3 function) and the worst overloaded stations have an overload of 3.48 and 4.21 time units (R_2).

Problem instances P6, P7, and P8 are those having the highest number of tasks (320, 376, and 420) and reflect more robustness differences. The *advanced TSALBP-NSGA-II* has produced non-dominated sets with a high number of solutions: 14 in P6 and 11 in P7-P8. The robustness attainment surfaces are again complementary to numerical data. The graphical representation is necessary to easily find the most robust solutions (Figs. 8–10). In these



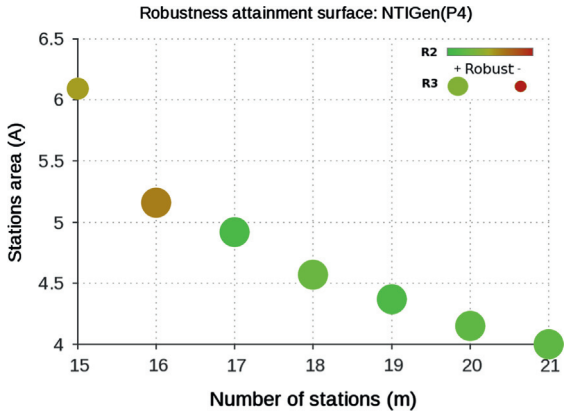
Sols.	R_1	R_2	R_3
S1	0.022	2.30	0.01
S2	0.149	3.33	0.12

Fig. 5. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the NTIGen instance of 100 tasks (P1).



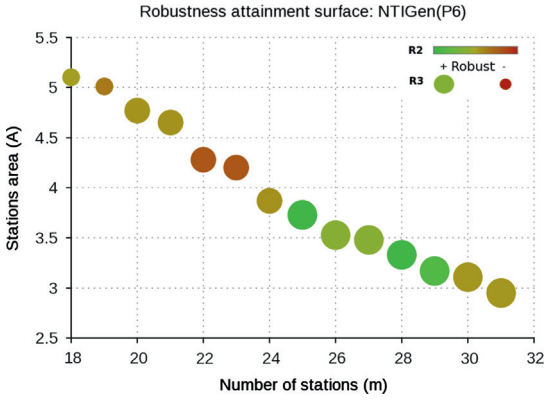
Sols.	R_1	R_2	R_3
S1	0.126	2.4	0.139
S2	0.103	1.98	0.149
S3	0.039	1.33	0.092
S4	0.044	1.05	0.083
S5	0.018	0.94	0.029

Fig. 6. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the original Nissan instance of 140 tasks (P2).



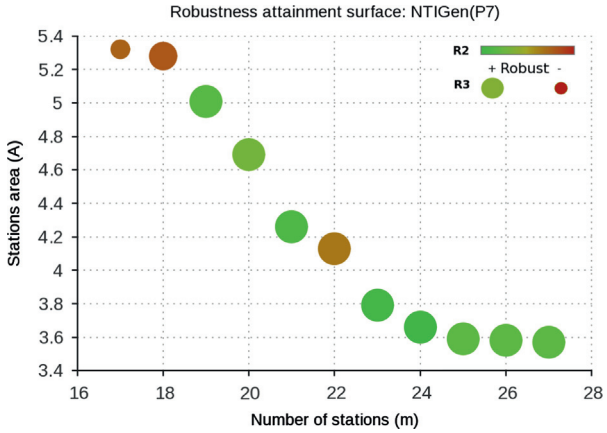
Sols.	R_1	R_2	R_3
S1	0.131	3.48	0.111
S2	0.137	4.21	0.052
S3	0.012	0.58	0.03
S4	0.035	1.59	0.046
S5	0.009	0.72	0.017
S6	0.023	1.33	0.033
S7	0.013	1.33	0.032

Fig. 7. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the NTIGen instance of 220 tasks (P4).



Sols.	R_1	R_2	R_3
S1	0.231	2.63	0.185
S2	0.226	3.29	0.158
S3	0.095	2.79	0.075
S4	0.072	2.82	0.079
S5	0.093	4.04	0.083
S6	0.084	4.04	0.072
S7	0.066	2.85	0.055
S8	0	0	0
S9	0.019	1.93	0.019
S10	0.018	1.19	0.018
S11	0	0	0
S12	0.02	1.36	0.029
S13	0.029	2.74	0.017
S14	0.028	2.74	0.016

Fig. 8. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the NTIGen instance of 320 tasks (P6).



Sols.	R_1	R_2	R_3
S1	0.211	4.15	0.157
S2	0.142	4.42	0.093
S3	0.021	0.86	0.035
S4	0.014	1.61	0.017
S5	0.005	0.54	0.024
S6	0.065	3.67	0.038
S7	0.04	2.02	0.022
S8	0	0	0
S9	0	0	0
S10	0.008	0.97	0.013
S11	0.008	0.97	0.012

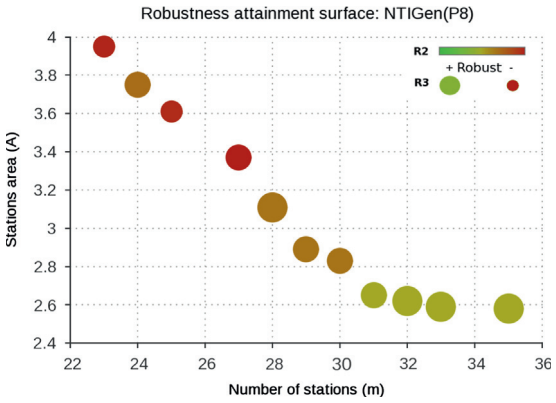
Fig. 9. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the NTIGen instance of 376 tasks (P7).

instances it is also possible to discover different robust areas in the Pareto front.

In view of the robustness function values for problem instances P6 and P7 (Figs. 8 and 9) as well as the counterpart figures, the most robust solutions for instance P6 are those having a number of stations between 25 and 29. The least robust ones are those with less than 20 stations as well as those between 21 and 23. Moreover, having the information that solutions #3 and #4 with 20 and 21 stations are more robust than their closest solutions is valuable for the DM. In this case, if the number of stations (workers) is not totally restricted, he/she can choose the most robust solution from the set of all non-dominated ones.

The least robust solutions of instance P7 are solutions #1, #2, and #6. These solutions have high values (low robustness) with respect to the others. Since the robust solutions are distributed along the entire surface of the Pareto front, the DM could select an assembly line configuration without taking into account these least robust options but always having more than 18 stations.

The last instance is P8 and we can find significant robustness differences among the non-dominated solutions. Solutions #1, #3 and #4 have very high R_1, R_2, and R_3 values and then, they are not recommended if the DM is looking for robust configuration lines for demand changes. If implementing these solutions, there could be overloaded stations with more than four time units each. If the



Sols.	R_1	R_2	R_3
S1	0.193	4.02	0.145
S2	0.093	2.86	0.07
S3	0.199	3.97	0.127
S4	0.119	4.16	0.074
S5	0.068	2.76	0.047
S6	0.092	2.76	0.069
S7	0.081	2.76	0.055
S8	0.059	2.01	0.054
S9	0.028	2.01	0.026
S10	0.026	2.01	0.025
S11	0.025	2.01	0.024

Fig. 10. Robustness attainment surface (representing R_2 and R_3) and robustness values for the non-dominated solutions when solving the NTIGen instance of 420 tasks (P8).

number of stations (and then, workers) is restricted, the DM can choose solution #2 having 24 stations which is more robust under an uncertain environment. Apart from this solution, solutions with more than 31 stations (#9, #10, and #11) are the most robust in comparison with the others.

In summary, there are important differences in terms of robustness in all the instances but P3 and P5. In particular, the graphical representation by means of robustness attainment surfaces helped to find robust solutions close to others which are not. Examples of this fact are instances P2, P5, P7, and P8 where the DM can guide her/his decision to robust solutions without a loss of optimized objective values and without the need of additional computations within the MOO method run.

7. Concluding remarks

The existing TSALBP formulation and previous ALB works do not cover an important real scenario where the same assembly line is devoted to produce mixed products and their demand is not fixed. In this paper we have presented a new robustness model to add important information in the MCDM process by evaluating the most robust assembly line configurations when future demand conditions can vary. The model comprises robustness functions and a graphical representation.

Three robustness functions, R_1, R_2, R_3, are defined based on the number of overloaded stations and the size of these overloads. The graphical representation of the robustness information makes use of the color and size of each non-dominated solution point to form the robustness attainment surfaces of the Pareto fronts.

The proposed model was used to analyze the non-dominated solutions provided by the state-of-the-art MOO method for the TSALBP-m/A, the *advanced TSALBP-NSGA-II*, although the nature of the robustness model allows the use of any other MOO method instead. The results of the application of the robustness model are clear. There are some solutions which are less robust than others when demand changes and the DM can take advantage of this information before making her/his decision.

Furthermore, the inclusion of the robustness information within the graphical representation of the Pareto front has shown a practical use as it clearly presents which solutions are robust and robustness areas of interest at a glance. The DM is now able to analyze the robustness information and identify robust Pareto front regions and their assembly line configuration alternatives.

In addition, the NTIGen software was presented to allow researchers to create realistic TSALBP instances and production plans for future research. The generated TSALBP instances contain many real-like industrial features, e.g. checkpoints, isolated tasks, initial and final tasks, chains of tasks, or stages, which make the NTIGen software a practical tool for simulating the industrial conditions of an assembly line.

Some future works arise from this contribution: (i) to include the robustness information within the search process of the MOO method and (ii) to design a global visualization framework also representing the assembly line configurations and the relations between the different alternative solutions.

Acknowledgements

This work has been supported by Ministerio de Economía y Competitividad under project SOCOVIFI2 (TIN2012-38525-C02-01 and TIN2012-38525-C02-02) and under PROTHIUS-III: DPI2010-16759, both including EDRF funding.

References

Battaïa, O., Dolgui, A., 2013. A taxonomy of line balancing problems and their solution approaches. International Journal of Production Economics 142, 259–277.
Bautista, J., Pereira, J., 2007. Ant algorithms for a time and space constrained assembly line balancing problem. European Journal of Operational Research 177, 2016–2032.
Baybars, I., 1986. A survey of exact algorithms for the simple assembly line balancing problem. Management Science 32, 909–932.
Benson, H.P., Sayin, S., 1997. Toward finding global representations of the efficient set in multiple-objective mathematical programming. Naval Research Logistics 44, 47–67.
Beyer, H., Sendhoff, B., 2007. Robust optimization—a comprehensive survey. Computer Methods in Applied Mechanics and Engineering 196, 3190–3218.
Bhattacharjee, T.K., Sahu, S., 1990. Complexity of single model assembly line balancing problems. Engineering Costs and Production Economics 18, 203–214.
Blasco, X., Herrero, J.M., Sanchís, J., Martínez, M., 2008. A new graphical visualization of *n*-dimensional Pareto front for decision-making in multiobjective optimization. Information Sciences 178, 3908–3924.
Bonissone, P., Subbu, R., Lizzi, J., 2009. Multicriteria decision making (MCDM): a framework for research and applications. IEEE Computational Intelligence Magazine 4, 48–61.
Boysen, N., Fliedner, M., Scholl, A., 2007. A classification of assembly line balancing problems. European Journal of Operational Research 183, 674–693.
Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: which model to use when?. International Journal of Production Economics 111, 509–528.
Chica, M., Cordón, O., Damas, S., 2011. An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. Computers and Industrial Engineering 61, 103–117.
Chica, M., Cordón, O., Damas, S., Bautista, J., 2010. Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. Information Sciences 180, 3465–3487.
Chica, M., Cordón, O., Damas, S., Bautista, J., 2012. Multiobjective memetic algorithms for time and space assembly line balancing. Engineering Applications of Artificial Intelligence 25, 254–273.
Dar-El, E.M., 1975. Solving large single-model assembly line balancing problems - A comparative study. IIE Transactions 7, 302–310.
Deb, K., Gupta, H., 2006. Introducing robustness in multi-objective optimization. Evolutionary Computation 14, 463–494.

Dolgui, A., Kovalev, S., 2012. Scenario based robust line balancing: computational complexity. Discrete Applied Mathematics 160, 1955–1963.
Ferreira, J., Fonseca, C.M., Covas, J.A., Gaspar-Cunha, A., 2008. Evolutionary multi-objective robust optimization. In: Advances in Evolutionary Algorithms, (⟨www.i-techonline.com⟩). InTech, Vienna, Austria, pp. 261–278.
Gurevsky, E., Battaïa, O., Dolgui, A., 2012. Balancing of simple assembly lines under variations of task processing times. Annals of Operations Research 201, 265–286.
Gurevsky, E., Battaïa, O., Dolgui, A., 2013. Stability measure for a generalized assembly line balancing problem. Discrete Applied Mathematics 161, 377–394.
Hamta, N., Fatemi Ghomi, S., Jolai, F., Bahalke, U., 2011. Bi-criteria assembly line balancing by considering flexible operation times. Applied Mathematical Modelling 35, 5592–5608.
Hazır, Ö., Dolgui, A., 2013. Assembly line balancing under uncertainty: robust optimization models and exact solution method. Computers & Industrial Engineering 65, 261–267.
Knowles, J., 2005. A summary-attainment-surface plotting method for visualizing the performance of stochastic multiobjective optimizers. In: 5th International Conference on Intelligent Systems Design and Applications (ISDA 05), Wroclaw (Poland), pp. 552–557.
Larichev, O., 1992. Cognitive validity in design of decision-aiding techniques. Journal of Multi-Criteria Decision Analysis 1, 127–138.
Leung, S.C.H., Tsang, S.O.S., Ng, W.L., Wu, Y., 2007. A robust optimization model for multi-site production planning problem in an uncertain environment. European Journal of Operational Research 181, 224–238.
Lim, D., Ong, Y., Sendhoff, B., Lee, B., 2006. Inverse multi-objective robust evolutionary design. Genetic Programming and Evolvable Machines 7, 383–404.
López-Ibáñez, M., Stützle, T., Paquete, L., 2010. Graphical tools for the analysis of bi-objective optimization algorithms. In: Proceedings of the 12th Annual Conference Companion on Genetic and Evolutionary Computation. ACM, New York, NY, USA, pp. 1959–1962.
Lotov, A., Miettinen, K., 2008. Visualizing the Pareto frontier. In: Multiobjective Optimization. Interactive and Evolutionary Approaches. Lecture Notes in Computer Science, vol. 5252. Springer, pp. 213–243.
Miettinen, K., Deb, K., Jahn, J., Ogryczak, W., Shimoyama, K., Vetschera, K., 2008. Future challenges. In: Multiobjective Optimization. Interactive and Evolutionary Approaches, Lecture Notes in Computer Science, vol. 5252. Springer, pp. 435–461.
Obayashi, S., Sasaki, D., 2003. Visualization and data mining of Pareto solutions using self-organizing map. In: Fonseca, C., Fleming, P., Zitzler, E., Thiele, L., Deb, K. (Eds.), Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, vol. 2632. Springer, Berlin, Heidelberg, p. 71.
Ong, Y.S., Nair, P.B., Lum, K.Y., 2006. Max–min surrogate-assisted evolutionary algorithm for robust design. IEEE Transactions on Evolutionary Computation 10, 392–404.
Otto, A., Otto, C., Scholl, A., 2011. SALBPgen—a systematic data generator for (simple) assembly line balancing. Jena Research Papers in Business and Economics 5.
Romero Zaliz, R., Rubio-Escudero, C., Cordón, O., Cobb, J., Herrera, F., Zwir, I., 2008. A multi-objective evolutionary conceptual clustering methodology for gene annotation within structural databases: a case of study on the gene ontology database. IEEE Transactions on Evolutionary Computation 12, 679–701.
Roy, B., 1998. A missing link in OR-DA: robustness analysis. Foundations of Computing and Decision Sciences 23, 141–160.
Roy, B., 2010. Robustness in operational research and decision aiding: a multi-faceted issue. European Journal of Operational Research 200, 629–638.
Scheffermann, R., Bender, M., Cardeneo, A., 2009. Robust solutions for vehicle routing problems via evolutionary multiobjective optimization. In: IEEE Congress on Evolutionary Computation (CEC), Trondheim, Norway, pp. 1605–1612.
Scholl, A., 1999. Balancing and Sequencing of Assembly Lines, 2nd edition. Physica-Verlag, Heidelberg.
Scholl, A., Becker, C., 2006. State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. European Journal of Operational Research 168, 666–693.
Shimoyama, K., Oyama, A., Fujii, K., 2005. A new efficient and useful robust optimization approach—design for multi-objective six sigma. In: IEEE Congress on Evolutionary Computation (CEC), Piscataway, pp. 950–957.
Tan, K.C., Cheong, C.Y., Goh, C.K., 2007. Robust transportation network design under demand uncertainty. Computer-Aided Civil and Infrastructure Engineering 22, 6–18.
Vargas-Quesada, B., de Moya-Anegón, F., 2007. Visualizaing the Structure of Science. Springer.
Xu, W., Xiao, T., 2010. Robust assembly line balancing with interval uncertain task times. Computer-Integrated Manufacturing Systems 16, 1202–1207.
Xu, W., Xiao, T., 2011. Strategic robust mixed model assembly line balancing based on scenario planning. Tsinghua Science & Technology 16, 308–314.