

A multiobjective model and evolutionary algorithms for robust time and space assembly line balancing under uncertain demand

Manuel Chica^{*,a}, Joaquín Bautista^b, Óscar Cordon^{a,c}, Sergio Damas^a

^a*European Centre for Soft Computing, 33600 Mieres, Spain*

^b*ETSEIB, Universitat Politècnica de Catalunya, 08028 Barcelona, Spain*

^c*DECSAI and CITIC-UGR, University of Granada, 18071 Granada, Spain*

Abstract

Changes in demand when manufacturing different products require an optimization model that includes robustness in its definition and methods to deal with it. In this work we propose the r-TSALBP, a multiobjective model for assembly line balancing to search for the most robust line configurations when demand changes. The robust model definition considers a set of demand scenarios and presents temporal and spatial overloads of the stations in the assembly line of the products to be assembled. We present two multiobjective evolutionary algorithms to deal with one of the r-TSALBP variants. The first algorithm uses an additional objective to evaluate the robustness of the solutions. The second algorithm employs a novel adaptive method to evolve separate populations of robust and non-robust solutions during the search. Results show the improvements of using robustness information during the search and the outstanding behavior of the adaptive evolutionary algorithm for solving the problem. Finally, we analyze the managerial impacts of considering the r-TSALBP model for the different organization departments by exploiting the values of the robustness metrics.

Key words: Robust Optimization, Assembly Line Balancing, Multiobjective Evolutionary Algorithms, Uncertain Demand

*Corresponding author

Email addresses: manuel.chica@softcomputing.es (Manuel Chica),
joaquin.bautista@upc.edu (Joaquín Bautista), oscar.cordon@softcomputing.es
(Óscar Cordon), sergio.damas@softcomputing.es (Sergio Damas)

1. Introduction

Finding the most robust solutions for operation research problems is a very active research field where we can find optimization algorithms that consider robustness such as those for portfolio planning (Ogüzsoy and Güven 2007) and product design problems (Wang and Curry 2012). Robust optimization tries to find flexibility by its way of solving problems. Particularly, flexibility is an important asset to manufacturing firms to respond to changes in the environment (Bengtsson and Olhager 2002) and this flexibility also applies to the automotive industry and assembly line balancing. An assembly line consists of a set of workstations and different tasks requiring an operation time for their execution. These tasks divide the manufacturing of a production item and one usual and difficult problem, called assembly line balancing (ALB) (Boysen et al. 2007; 2008, Battaia and Dolgui 2013), is to determine how these tasks can be assigned to the stations fulfilling certain restrictions.

The simple assembly line balancing problem (SALBP) (Scholl 1999, Scholl and Becker 2006) belongs to the ALB family of problems and optimally partitions tasks to stations with respect to some objective (such as line cycle time) in such a way that all the precedence constraints are satisfied. Bautista and Pereira (2007) proposed an SALBP extension aiming to design a more realistic ALB model. They defined the time and space assembly line balancing problem (TSALBP) by considering an additional space constraint to get a simplified but closer version to existing real-world situations. 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).

All the latter models assume the balance of the assembly lines when producing mixed products. Nevertheless, product demands are not usually fixed and certain. This is a usual situation in many managerial and operation research problems where demand is uncertain. See for instance supply chain coordination (Gao et al. 2014) or bus scheduling (Yan and Tang 2009)). For the TSALBP, for instance, when the assembly line produces mixed products in a given sequence the model considers the operation time of the tasks as an average of the times of the different products and their demand. If the demand changes, the operation time also changes and the line configuration may need a re-balancing. This re-balancing may cause production losses

because those workers assigned to workstations will have to comply with new tasks and increase their learning curve to work in the line.

These difficulties have a high managerial impact in the production system of the plant and have encouraged us to propose the Robust model for Time and Space Assembly Line Balancing Problem (r-TSALBP). The r-TSALBP is a new ALB model that integrates the concept of robust solutions (Roy 2010, Beyer and Sendhoff 2007) to provide practitioners with the most efficient assembly line configurations when demand changes, also having a low impact on the management of the plant. The model links robustness with the flexibility of an assembly line configuration when demand changes based on a set of real production plans. The goal is to identify how robust a line configuration is for a set of production plans according to both operation time and linear area. This is carried out by using temporal and spatial robustness functions, based on overload, for each station and production plan.

Although overload in assembly lines is not a new concept in production and assembly lines (Chica et al. 2013, Xu and Xiao 2009; 2011), the r-TSALBP is the first model that incorporates this concept as part of its definition. Our r-TSALBP model is based on straight mixed-model lines and uses uncertain values for task attributes based on a scenario definition. The r-TSALBP model also involves single-objective and multiobjective problem variants and is parameterized to let the decision maker specify the desired robustness level. The model is also versatile and allows the use of diverse optimization methods that can deal with robustness in different ways. To illustrate the mentioned versatility we design two evolutionary methods to solve one of the r-TSALBP multiobjective variants by using two mechanisms: including a third objective for the robustness of the solutions and controlling the feasibility of the population with respect to their robustness.

The first method is based on redesigning the existing *TSALBP-NSGA-II* (Chica et al. 2011) for handling the three-objective problem. The second method is a novel adaptive version of the Infeasibility Driven Evolutionary Algorithm (IDEA) (Singh et al. 2008). The original IDEA version was explicitly designed for industrial constrained optimization problems such as ALB and was already applied to solve the TSALBP in Rada-Vilela et al. (2013). In the current contribution we propose an extension of the original version to search for robust TSALBP solutions by making IDEA adaptive. This behavior is achieved by dividing the population of the algorithm in robust and non-robust sub-populations of solutions and by adapting the size of both populations depending on the robustness of the Pareto archive every

generation. Although Rada-Vilela et al. (2013) and the current proposal considers the same original algorithm, the corresponding goals and approaches are neither equivalent nor comparable.

The experimentation comprises the performance evaluation of the methods in a set of seven instances generated by the NTIGen software (Chica et al. 2014). These instances belong to a diverse set of TSALBP instances that incorporates the real data and industrial features of the Nissan industry plant of Barcelona. Additionally, both methods are also compared with the original *TSALBP-NSGA-II* to analyze the effect of including the robustness computation during the search process or afterwards. The analysis employs commonly used multiobjective performance indicators and robustness graphical representations of the non-dominated solutions.

The discussion of the paper is structured as follows. First, in Section 2 we present background information on the TSALBP and a complete literature review on robust ALB. We describe the r-TSALBP model in Section 3. Then, Section 4 explains the proposed evolutionary methods for solving the r-TSALBP and Section 5 describes the experimental results.

Finally in Section 6 we discuss the managerial implications of considering uncertain demand when balancing the assembly line and the meaning of the robustness functions included in the model. By starting from the managerial effects on the organization we highlight the role of these temporal and spatial robustness functions as indicators of the actions impact. These indicators are six hierarchical metrics due to their relationships with the departments and hierarchy of the industrial organization. The practical limitations of the model, metrics, and computational methods in conjunction with some future works to alleviate them are discussed in Section 7.

2. Background

2.1. Time and space assembly line balancing problem description

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 manufacturing technologies and the employed resources. Each station k ($k = 1, 2, \dots, m$) is assigned to a subset of tasks S_k ($S_k \subseteq J$) which is called workload of the station. Besides, each station k has a workload time $t(S_k)$ which is equal to the sum of the processing times of its assigned tasks (workload of the station).

Each task j can only be assigned to a single station k and has a set of direct “preceding tasks” P_j which must be accomplished before j is started. These constraints are normally represented by means of an acyclic precedence graph. The vertexes of the graph represent the tasks where a directed arc (i, j) indicates that, on the production line, task i must finish before the start of task j . Then, task j cannot be assigned to a station that is ordered before the one where task i was assigned. k .

SALBP, a family of ALB problems, focuses on grouping tasks in workstations by an efficient and coherent way. Because of the need of introducing space constraints in ALB, a required area a_j is associated to each task j and an available area A_k to each station k ; identical for every station and equal to $A = \max_{k=1,2,\dots,m} A_k$. Each station k has then an available station area $a(S_k)$ which is equal to the sum of areas required by the tasks assigned to the station k .

TSALBP (Bautista and Pereira 2007) is the family of problems that introduced this space features in ALB. Time and Space Assembly Line Balancing Problem (TSALBP) states that, for a set of n tasks with their temporal t_j and spatial a_j attributes ($1 \leq j \leq n$) 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).

According to the ALB taxonomy given by Battaïa and Dolgui (2013), TSALBP is a single-model line with a basic straight line layout; it has tasks with constant values for both temporal (t_j) and spatial attributes (a_j); it includes three types of constraints: assignment (given by a precedence graph), cycle time, and linear area; and it defines three functions to minimize in a multiobjective fashion (cycle time c , area of stations A , and number of stations m).

2.2. Robust optimization for assembly line balancing

One of the most common ways of finding robust solutions for ALB is to search for the solutions that perform well across all possible scenarios (Battaïa and Dolgui 2013). Using this approach Xu and Xiao (2011) dealt with the mixed ALB problem variant and proposed a lexicographic-order on the α -worst case scenario. A previous work of the same authors used a min-max relative regret mechanism instead, which is a more conservative approach (Xu and Xiao 2009). Saif et al. (2014) presented a similar

work but for a multiobjective problem by defining a Pareto lexicographic α -robust dominance relationship.

Chica et al. (2013) also defined a set of scenarios and proposed novel robustness functions and a graphical representation to respectively measure and represent how robust the assembly line configuration is on this set of scenarios (product plans). The values of these functions were computed *a posteriori* once a multiobjective optimization method obtained a non-dominated solutions set. The work of Papakostas et al. (2014) also evaluated *a posteriori* the solutions of a model for minimizing time and cost through a set of demand profiles but they used single-objective particle swarm optimization.

In Simaria et al. (2009) authors presented an ant colony optimization (ACO) algorithm for a flexible U-shaped assembly line where workstations remain unchanged (tools and equipment) but the number of operators and the tasks are adjusted to meet the demand. Again, the goal of the model was to design an assembly line flexible enough to cope with the different demand scenarios. Dolgui and Kovalev (2012) proposed an ALB model and a dynamic programming method to minimize the cycle time by following a worst scenario approach; while Li and Gao (2014) characterized unstable demand in manual mixed-model assembly lines by several representative scenarios.

Another well-known way of considering uncertainty and robustness in ALB is by assuming that task times have uncertain values by defining intervals or known distributions. For instance, Gurevsky et al. (2012) dealt with the SALBP-E when having task times within intervals and proposed a way to find a compromise between the objective function minimization and a stability ratio (a parameter to reflect the risk aversion coefficient of a decision maker and the conservatism). A related stability study was done in Gurevsky et al. (2013) but for the case of an ALB problem where a workstation can have several workplaces, there are exclusion constraints, and the processing times of the tasks can vary during the line life cycle. Hazır and Dolgui (2013) recently presented two robust SALBP-2 models having interval uncertainty for operation times and solved them by a decomposition method and enhancement strategies.

To sum up, the majority of the approaches existing in the literature for robust ALB are based on considering uncertain tasks attributes by defining interval values or by setting different scenarios. The most used robust criteria rely on the worst case by using traditional min-max or variations of it (Dolgui and Kovalev 2012, Simaria et al. 2009, Xu and Xiao 2011; 2009, Saif et al. 2014). The existing approaches are summarized in Table 1.

	Assembly line	Task attributes		Objectives	Robustness criteria	Solving method
		Uncertain	Scenarios			
Xu and Xiao (2009)	Mixed		✓	Workload variance	Min-max, α -worst	Genetic algorithm
Xu and Xiao (2011)	Mixed		✓	Workload variance	Lexicographic α -worst	Genetic algorithm
Chica et al. (2013)	Single		✓	No. stations and their area	Time overload	MO Genetic algorithm
Simaria et al. (2009)	U-shaped		✓	Idle time and workload	α -worst, min-max regret	ACO
Dolgui and Kovalev (2012)	Single		✓	Cycle time	Worst scenario	Dynamic programming
Gurevsky et al. (2012)	Single	✓		No. stations and time most loaded	Stability radius	Multi-start heuristic
Hazır and Dolgui (2013)	Single	✓		Cycle time	-	Decomposition method
Gurevsky et al. (2013)	Multiple workplaces	✓		Weighted no. of stations	Stability radius	Heuristic procedure
Li and Gao (2014)	Mixed		✓	Labor costs	Time overload	Branch and bound
Papakostas et al. (2014)	Single		✓	Cycle time	Posterior evaluation	PSO
Saif et al. (2014)	Single		✓	Cycle time and smoothness	Pareto lexicographic α -worst	Pareto dominance ‡
‡ Authors presented an illustrative example without solving the problem with a specific search method.						

Table 1: Publications with robust models for ALB and their elements according to the taxonomy given in Battaia and Dolgui (2013).

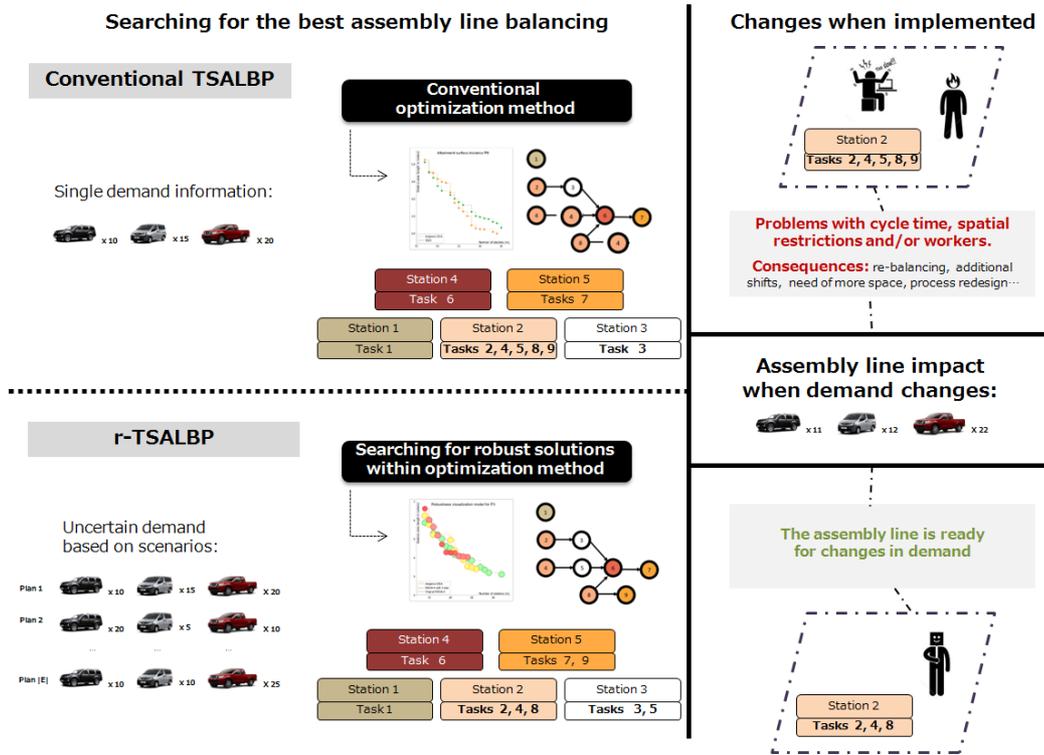


Figure 1: Graphical representation of a comparison between TSALBP and r-TSALBP. A robust tasks-stations assignment is performed by an appropriate model and method. In r-TSALBP, the resulting assembly line is ready for changes in demand.

3. The r-TSALBP model

TSALBP assumes both a constant demand and fixed operation time and required area (t_j and a_j). Real assembly lines normally assemble more than a single product and when demand changes, the operation time and area of the tasks change in consequence.

The demand of a set of mixed products is defined by production plans. For instance, the engine assembly line of the Nissan Spanish Industrial Operations plant has nine production plans: the first three engine plans are built for 4×4 vehicles, the fourth and fifth for VANs; and the remaining for medium tonnage trucks. When demand is balanced (identical for the products), the cycle time is 3 minutes, and the assembly line is divided into 21

workstations; then the maximum required linear area is 4 m¹.

However, it is not realistic to always have the same uniform demand for all the engines within a global demand. For instance, the processing time of a specific operation can be different when an engine is assembled for a truck or a VAN. Meanwhile, if demand changes and more products with higher linear area requirements have to be assembled, the area of the tasks will necessary increase. See Figure 1 for a diagram example of the problem situation.

A robust line configuration means that no reconfiguration is required whenever a task attribute varies. This fact can save efficiency loss and time for workers to get used to working under the new production scenario. Through the definition of r-TSALBP in Sections 3.1 to 3.5 we propose different single and multiobjective problem variants to solve ALB with robust solutions when demand changes in mixed products lines.

3.1. Mathematical definition

Being J the set of tasks for assembling a set I of product types, a task $j \in J$ requires a processing time of t_{ji} for assembling product $i \in I$. Similarly, we denote the required area of j for processing product i by a_{ji} . We also call Ψ to the set of assembly line configurations and ψ a specific line configuration which belongs to the set.

We define E as the set of realistic production plans to model the demand variation of the mix of products to be assembled. One of the plans of E is called the reference production plan, ε^0 , and ψ^0 is its reference line configuration. For the Nissan case this reference plan ε^0 is the one having a balanced demand for the products of I .

Given a production plan $\varepsilon \in E$, defined by a demand vector $\vec{d}_\varepsilon = (d_{1\varepsilon}, d_{2\varepsilon}, \dots, d_{|I|\varepsilon})$, we can determine the average processing time and required area of task $j \in J$ for this plan ε by Equations 1 and 2, respectively:

$$\bar{t}_{j\varepsilon} = \frac{1}{D_\varepsilon} \sum_{i=1}^{|I|} t_{ji} d_{i\varepsilon}, \quad (1)$$

$$\bar{a}_{j\varepsilon} = \frac{1}{D_\varepsilon} \sum_{i=1}^{|I|} a_{ji} d_{i\varepsilon}, \quad (2)$$

¹The required area is specified by two-dimensional units but the variable for optimization is the length of the tasks and the station's length, measured in linear meters.

where D_ε is the global demand of plan ε given by $D_\varepsilon = \sum_{i=1}^{|I|} d_{i\varepsilon}$.

The used variables belong to the set of parameters (Table 2) and variables (Table 3) of the r-TSALBP model specification. Table 4 shows the associated restrictions.

J	Set of all the elementary processing tasks of the line ($j = 1, \dots, J $)
n	Number of tasks of the line: $n = J $
K	Set of workstations ($k = 1, \dots, K $)
c	Cycle time (it can also be a variable)
Ψ	Set of line configurations ($\psi = 1, \dots, \Psi $)
E	Set of demand production plans ($\varepsilon = 1, \dots, E $)
$\bar{t}_{j\varepsilon}$	Average processing time of the elementary task $j \in J$ (measured at normal work pace) in production plan $\varepsilon \in E$
$\bar{a}_{j\varepsilon}$	Average linear area (length) required to perform task $j \in J$ for the production plan $\varepsilon \in E$
P_j	Set of immediate “preceding tasks” which must be accomplished before j is started
UB_m	Upper bound of the number of stations. It is equal to the number of tasks
γ^c, γ^A	Flexibility control parameters for exceeding cycle time and linear area
Δ^c	Maximum exceeding time for all the stations $k \in K$ at normal work pace. $\Delta^c = \gamma_c c$
Δ^A	Maximum exceeding linear area available for all the station $k \in K$. $\Delta^A = \gamma_A A$

Table 2: Parameters of the r-TSALBP model

x_{jk}	Binary variable being 1 if task $j \in J$ is assigned to station $k \in K$. Otherwise its value is 0
S_k	Subset of tasks assigned to each station $k \in K$: $S_k = \{j \in J : x_{jk} = 1\}$ (referred as the workload of the station)
m	Number of workstations of the line: $m = K $ (it can also be a parameter)
A	Linear area of each station k of the assembly line (it can also be a parameter)
$y_{k\varepsilon}^c$	Binary variable being 1 if the processing time required in station $k \in K$ for the production plan $\varepsilon \in E$ ($\sum_{j \in S_k} \bar{t}_{j\varepsilon}$) exceeds the cycle time c . Otherwise, 0
$y_{k\varepsilon}^A$	Binary variable being 1 if the linear area required in station $k \in K$ for the production plan $\varepsilon \in E$ ($\sum_{j \in S_k} \bar{a}_{j\varepsilon}$) exceeds the available area A . Otherwise, 0

Table 3: Variables of the r-TSALBP model

Binary condition of the station-task assignment variable:

$$x_{jk} \in \{0, 1\}, \quad (j = 1, \dots, |J|; k = 1, \dots, |K|) \quad (3)$$

Binary conditions to denote variables exceeding time and/or linear area, respectively:

$$y_{k\varepsilon}^c \in \{0, 1\}, y_{k\varepsilon}^A \in \{0, 1\} \quad (k = 1, \dots, |K|; \varepsilon = 1, \dots, |E|) \quad (4)$$

Every task must be assigned to just one single station:

$$\sum_{k=1}^{|K|} x_{jk} = 1, \quad (j = 1, \dots, |J|) \quad (5)$$

Every station must contain at least one task:

$$\sum_{j=1}^{|J|} x_{jk} \geq 1, \quad (k = 1, \dots, |K|) \quad (6)$$

The assignment cannot violate the immediate precedence relations:

$$\sum_{k=1}^{|K|} k(x_{ik} - x_{jk}) \leq 0, \quad (i \in P_j, j = 1, \dots, |J|) \quad (7)$$

The station workload time cannot exceed the maximum cycle time (including the defined allowance):

$$\sum_{j=1}^{|J|} \bar{t}_{j\varepsilon} x_{jk} \leq (c + \Delta^c y_{k\varepsilon}^c), \quad (k = 1, \dots, |K|; \varepsilon = 1, \dots, |E|) \quad (8)$$

The length of the station workload cannot exceed the maximum linear area (including the defined allowance):

$$\sum_{j=1}^{|J|} \bar{a}_{j\varepsilon} x_{jk} \leq (A + \Delta^A y_{k\varepsilon}^A). \quad (k = 1, \dots, |K|; \varepsilon = 1, \dots, |E|) \quad (9)$$

Table 4: Restrictions of the r-TSALBP model

Property 1. *If $\gamma_c = 0$ and $\gamma_A = 0$ the r-TSALBP variant is converted to the original TSALBP model.*

Equations 10 to 12 define the main objective functions of the r-TSALBP model. The first equation represents the number of stations of the line configuration, the second one the cycle time of the line, and the third one the available linear area of the stations of the line. The second and third equations are calculated using the reference production plan ε^0 .

$$f^1(x) = m = \sum_{k=1}^{UB_m} \max_{j \in J} \{x_{jk}\}, \quad (10)$$

$$f^2(x) = c = \max_{k \in K} \left\{ \sum_{j=1}^{|J|} \bar{t}_{j\varepsilon^0} x_{jk} \right\}, \quad (11)$$

$$f^3(x) = A = \max_{k \in K} \left\{ \sum_{j=1}^{|J|} \bar{a}_{j\varepsilon^0} x_{jk} \right\}. \quad (12)$$

3.2. Temporal robustness functions

Furthermore, the r-TSALBP formulation adds secondary functions which can be divided into temporal and spatial robustness functions. Both groups of functions are normalized to $[0, 1]$. Within the temporal functions we note the following:

- Rate of overloaded production plans with respect to the allowed workload time (Equation 13).

$$g_c^1 = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \max_{k \in K} y_{k\varepsilon}^c. \quad (13)$$

- Rate of overloaded stations with respect to the allowed workload time (Equation 14).

$$g_c^2 = \frac{1}{m} \sum_{k=1}^{|K|} \max_{\varepsilon \in E} y_{k\varepsilon}^c. \quad (14)$$

- Exceeding processing time of the stations in all the plans with respect to the maximum exceeding time and the number of overloaded stations (Equation 15).

$$g_c^3 = g_c^3(x) = \frac{1}{\Delta^c \sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} y_{k\varepsilon}^c} \sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} (\max\{0, \sum_{j=1}^{|J|} \bar{t}_{j\varepsilon} x_{jk} - c\}). \quad (15)$$

3.3. Spatial robustness functions

Analogously, the secondary spatial robustness functions are defined as follows:

- Rate of overloaded production plans with respect to the available linear area (Equation 16).

$$g_A^1 = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \max_{k \in K} y_{k\varepsilon}^A. \quad (16)$$

- Rate of overloaded stations with respect to the available linear area (Equation 17).

$$g_A^2 = \frac{1}{m} \sum_{k=1}^{|K|} \max_{\varepsilon \in E} y_{k\varepsilon}^A. \quad (17)$$

- Over-assigned linear area of the stations in all the plans with respect to the maximum exceeding area and the number of overloaded stations (Equation 18).

$$g_A^3 = g_A^3(x) = \frac{1}{\Delta^A \sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} y_{k\varepsilon}^A} \sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} (\max\{0, \sum_{j=1}^{|J|} \bar{a}_{j\varepsilon} x_{jk} - A\}). \quad (18)$$

3.4. *r*-TSALBP variants when considering robustness as a restriction

We introduce in Table 5 the single and multiobjective variants of *r*-TSALBP that define robustness as a restriction of the model. The additional restrictions for temporal robustness are given by Equation 19 and the additional spatial restrictions are specified by Equation 20.

$$g_c^1 \leq \tilde{g}_c^1; g_c^2 \leq \tilde{g}_c^2; g_c^3 \leq \tilde{g}_c^3, \quad (19)$$

$$g_A^1 \leq \tilde{g}_A^1; g_A^2 \leq \tilde{g}_A^2; g_A^3 \leq \tilde{g}_A^3, \quad (20)$$

where $\{\tilde{g}_c^1, \tilde{g}_c^2, \tilde{g}_c^3\}$ and $\{\tilde{g}_A^1, \tilde{g}_A^2, \tilde{g}_A^3\}$ are parameters defined in $[0, 1]$ that restrict the temporal robustness functions (g_c) and the spatial robustness functions (g_A). For instance, $\tilde{g}_c^1 = 0.4$ means that a solution is robust in 60% of the production plans (according to the workload of the stations).

Normally, a decision maker can use minimum temporal robustness parameters given by $\tilde{g}_c^1 = 1 - \tilde{r}_c^1$, $\tilde{g}_c^2 = 1 - \tilde{r}_c^2$, and $\tilde{g}_c^3 = 1 - \tilde{r}_c^3$. The same parameters can be used for spatial robustness (i.e. $\tilde{g}_A^1 = 1 - \tilde{r}_A^1$, $\tilde{g}_A^2 = 1 - \tilde{r}_A^2$, and $\tilde{g}_A^3 = 1 - \tilde{r}_A^3$).

3.5. *r-TSALBP variants when considering robustness as an objective function*

These model variants can handle weighted optimization functions with respect to time and linear area by using the robustness functions defined in Equations 13 to 15 and Equations 16 to 18. The resulting two objective functions are $G_c(x)$ (Equation 21) and $G_A(x)$ (Equation 22).

$$G_c(x) = \sum_{i=1}^3 (\mu_c^i \max\{0, g_c^i - \tilde{g}_c^i\}), \quad (21)$$

$$G_A(x) = \sum_{i=1}^3 (\mu_A^i \max\{0, g_A^i - \tilde{g}_A^i\}), \quad (22)$$

where μ_c^i and μ_A^i are linear combination weights $\in [0, 1]$ satisfying that $\sum_{i=1}^3 \mu_c^i \leq 1$ and $\sum_{i=1}^3 \mu_A^i \leq 1$.

These objective functions extend the original optimization functions $f^1(x)$, $f^2(x)$, and $f^3(x)$ of the classical TSALBP model and can be dealt with in a combined way by optimization methods.

Variant	Parameters	# objs.	Minimization objectives	Temporal restrictions	Spatial restrictions	Non linearity
r-TSALBP-m †	A, c	1	$f^1(x)$ (Eq. 10)	✓	✓	-
r-TSALBP-A	m, c	1	$f^3(x)$ (Eq. 12)	✓	✓	Eq. 9
r-TSALBP-c †	m, A	1	$f^2(x)$ (Eq. 11)	✓	✓	Eq. 8
r-TSALBP-m/c	A	2	$f^1(x) \wedge f^2(x)$ (Eqs. 10 and 11)	✓	✓	Eq. 8
r-TSALBP-m/A	c	2	$f^1(x) \wedge f^3(x)$ (Eqs. 10 and 12)	✓	✓	Eq. 9
r-TSALBP-c/A	m	2	$f^2(x) \wedge f^3(x)$ (Eqs. 11 and 12)	✓	✓	Eqs. 8 and 9
r-TSALBP-m/c/A	-	3	$f^1(x) \wedge f^2(x) \wedge f^3(x)$ (Eqs. 10 to 12)	✓	✓	Eqs. 8 and 9
†Note that when $A \rightarrow \infty$, r-TSALBP-m and r-TSALBP-c are equivalent to r-SALBP-1 and r-SALBP-2, respectively.						

Table 5: Characteristics of the different r-TSALBP variants.

4. Multiobjective evolutionary algorithms for solving the r-TSALBP-m/A

In this section we describe three evolutionary multiobjective optimization methods for solving one of the multiobjective variants defined in the last section: the r-TSALBP-m/A ². The common operators for the methods are described in Section 4.1. In Section 4.2 we redesign the TSALBP-NSGA-II for including a robustness measure as a third objective. Section 4.3 explains the original and adaptive IDEA that deal with robustness as a restriction.

4.1. Common operators of the algorithms

4.1.1. Representation scheme

The scheme is an order-based representation that explicitly considers task-station assignments regardless the cycle time of the assembly line. By introducing separators (they are dummy genes) we allocate the tasks to the different stations. Separators do not represent any specific task and are inserted into the chromosome defining groups of tasks that are assigned to a specific station. The maximum possible number of separators is $n - 1$ (with n being the number of tasks) as it would correspond to an assembly line configuration with n stations of one single task. Tasks take value in $\{1, \dots, n\}$ while separators do in $\{n + 1, \dots, 2 \cdot n - 1\}$.

The number of separators included in the genotype is variable and it depends on the number of existing stations in the current solution. Therefore, the algorithm works with a variable-length coding scheme, although its order-based representation nature avoids the need of any additional mechanism to deal with this issue. The maximum size of the chromosome is $2 \cdot n - 1$ to allow the presence of separators for the maximum number of possible stations. On the other hand, the representation scheme ensures the encoded solutions are feasible with respect to the precedence relations constraints.

4.1.2. Crossover operator

The crossover operator is based on a classical order-based one: the PMX (Poon and Carter 1995). PMX generates two offspring from two parents by means of the following procedure: a) selection of two random cut

²To facilitate the analysis of the results we have restricted the r-TSALBP-m/A variant to the temporal robustness by only considering objective $G_c(x)$ (Equation 21) and restrictions given by Equation 19. The same analysis can be performed when also using the spatial robustness (objective $G_A(x)$ and restrictions of Equation 20).

1	3	2	9	5	7	8	10	4	6
---	---	---	---	---	---	---	----	---	---

Figure 2: Variable coding scheme of the chromosome for an eight-task example. Colored genes denote separators which always have integer values greater than the total number of tasks.

points, b) for the first offspring, copy of the genes outside the random points directly from the first parent, and c) copy of the genes inside the two cut points but in the order they appear in the second parent. The algorithm follows the same mechanism with the second offspring but with the opposite parents. Figure 3 shows an application example of the operator.

The feasibility of the offspring with respect to precedence relations is assured. However, since information about the tasks-stations assignment is encoded inside the chromosome, it is compulsory to assure that: a) there is not any station exceeding the fixed cycle time limit, and b) there is not any empty station in the configuration of the assembly line. Therefore, a repair operator must be applied for each offspring after crossover. The two methods of the repair operator are:

- Redistribute spare tasks among available stations: the repair operator must reallocate the spare tasks of stations with exceeding cycle time in other stations. First, these critical stations are found. Then, the operator finds the feasible available stations to reallocate each task of the critical station, fulfilling precedence and cycle time restrictions.
- Removing empty stations: no empty stations are allowed within the individual. For the genotype of the individual, this means that two or more genes representing separators cannot be placed together. Thus, the repair operator will find and remove them to only keep compulsory separators.

4.1.3. Mutation operators

In addition, we specifically design two mutation operators. The first one, the scramble operator, reorders a part of the sequence of tasks and reassigns them to stations. The second mutation operator, the divider operator, additionally increase the diversity of the search to obtain better distributed Pareto front approximations by randomly selecting one station with more

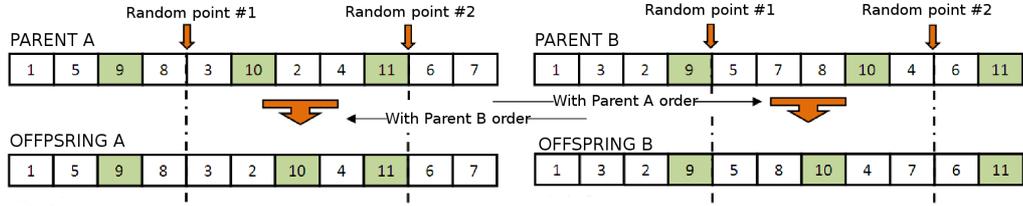


Figure 3: Variable-length crossover for a r-TSALBP example. Separators are the colored genes.

than one task and placing a separator, at a random position, to split up the station into two new stations.

4.2. NSGA-II with an additional objective

The design of this algorithm comes from the original NSGA-II search scheme (Deb et al. 2002) with more effective representation and operators (those described in the previous section) to solve the r-TSALBP-m/A. We call the algorithm *TSALBP-NSGA-II* because of its problem-specific design and potential application to other TSALBP variants (Chica et al. 2011). To deal with the robustness variable of the model we re-design the algorithm to manage three objective functions while maintaining the common operators previously described (i.e. representation, crossover, and mutations operators).

4.3. Adaptive IDEA

IDEA (Singh et al. 2008) is a multiobjective evolutionary algorithm for dealing with constrained optimization problems. In Rada-Vilela et al. (2013) we considered the use of IDEA to solve the conventional TSALBP. However, in this work we propose a novel approach based on the IDEA not to only embed the specific operators for our problem (those described in Section 4.1) but to be adaptive and to use the r-TSALBP-m/A robustness measures as additional constraints. This is an important difference with respect to the design of *TSALBP-NSGA-II* because IDEA will distinguish between robust and non-robust solutions during the search.

The design of IDEA emphasizes the search for optimal solutions near the constraint boundaries by maintaining and evolving a small proportion of unfeasible solutions. One of the strengths of this algorithm is to divide solutions (both in parent and offspring populations) into feasible and unfeasible

sets (robust and non-robust sets). The non-dominated sorting and crowding distance methods rank these sets independently. Then, the algorithm selects the solutions for the next generation from both feasible and unfeasible sets to maintain a pre-specified number of unfeasible solutions.

To do this, the algorithm uses a predefined parameter α_I to get the unfeasible solutions as a fraction of the size of the population. The effect of this α_I parameter is sensitive and this fact was considered in the seminal paper where authors claimed that the performance of IDEA is consistent over a wide range of α_I . However, we can achieve a higher (similar than *TSALBP-NSGA-II*) or lower convergence to the optimal Pareto front depending on the α_I value.

Our adaptive IDEA incorporates a novel mechanism. It consists of modifying the α_I parameter through the algorithm run depending on the robustness of the solutions of the Pareto set approximation obtained until that moment. First, we define the robustness of the Pareto set approximation as the ratio of solutions that are robust according to $G_c(x)$ (see Equation 21). Given a Pareto front approximation P we calculate its robustness ratio $r(P)$ as in Equation 23.

$$r(P) = \frac{1}{|P|} \sum_{\forall z \in P} (G_c(z) = 0). \quad (23)$$

Initially, the adaptive IDEA sets α_I to its starting value. Then, each time the algorithm modifies the solutions of the Pareto archive the adaptive process computes $r(P)$ and adjusts the α_I parameter in consequence. The goal of the adaptive process is to balance the importance of the unfeasible solutions of the population. If all the solutions of the Pareto set approximation are robust ($r(P) = 1$) the algorithm will increase the number of allowed unfeasible solutions of the population. When the number of non-robust solutions of the Pareto set approximation increases, the adaptive process will shrink the available space for unfeasible solutions. To sum up, the adaptive IDEA uses Equation 24 to adapt the α_I parameter at iteration t within the merge process of the feasible and unfeasible solutions.

$$\alpha_I^t = \begin{cases} \alpha_I^{t-1} - \Delta_\alpha \alpha_I^{t-1}, & \text{if } r(P^t) < \tau_r, \\ \alpha_I^{t-1} + \Delta_\alpha \alpha_I^{t-1}, & \text{if } r(P^t) > \tau_r, \end{cases} \quad (24)$$

where $\tau_r \in [0, 1]$ is a threshold for deciding whether the Pareto set approximation is robust to increase or decrease α_I and Δ_α is the step value for

modifying α_I at each generation. Note that if Δ_α equals to 0 the adaptive IDEA will turn into the original one.

After IDEA computes the new α_I^t the population is filled with the required number of feasible and unfeasible individuals as usual and the flow of the algorithm continues as in the original version of the IDEA.

5. Computational experience

5.1. Description of the TSALBP instances and experimental setup

The experimentation uses a set of seven real-like TSALBP instances. Table 6 shows the main features of these instances. We used the NTI-Gen software to generate them from a real instance of the Nissan plant in Barcelona (Spain). The interested reader can find the set and the software at www.prothius.com/TSALBP.

Features	NTI-Gen instances						
	P1+	P2	P3	P4	P5	P6	P7
Random seed	N/A	117017	21277	113683	56399	5869	73553
No. of tasks	140	190	220	280	320	376	420
Cycle time	180	207.07	222.42	221.62	169.552	186.65	137.751
Order strength	0.9	0.7	0.5	0.3	0.6	0.25	0.95
Precedences	293	314	304	407	435	548	608
Precs. window	N/A	5	1	2	1	3	2
Time variability	24	41.75	151.45	224.29	2742.28	901.34	1003.77
Linear area variability	513.86	266.67	300	400	200	300	133.33
Initial tasks	1	6	33	59	32	87	6
Final tasks	5	7	20	42	31	49	8
Isolated tasks	0	5	3	0	5	0	3
Checkpoints	N/A	0	6	7	1	12	0

+Original NISSAN instance.

Table 6: Main characteristics of the TSALBP instances.

We run each algorithm 10 times with different random seeds setting the run time as the stopping criterion. This run time is set to 300 seconds which were enough to get good solutions when running *TSALBP-NSGA-II* in previous publications (Chica et al. 2013). All the algorithms were launched in the same computer: Intel XeonTM E5530 with two CPUs at 2.40GHz, 3.7 Gbytes of memory, and Scientific Linux 6.4 as operating system. We use the same framework and programming language (C++) for the development of the algorithms. Table 7 shows the considered parameter values.

Parameter	Value	Parameter	Value
<i>r-TSALBP model</i>			
Minimum robustness $\{\tilde{r}_c^1, \tilde{r}_c^2, \tilde{r}_c^3\}$	{0.75, 0.9, 0.95}	γ_c for allowed exceeding time	0.05
Combination weights $\{\mu_c^1, \mu_c^2, \mu_c^3\}$	{0.2, 0.4, 0.4}		
<i>TSALBP-NSGA-II</i>			
Population size	100	Ishibuchi's similarity values	10
Crossover probability	0.8	Mutation probability	0.1
IDEAs		Adaptive IDEA	
Population size	100	α_I step	0.01
Crossover probability	0.8	Pareto robustness ratio (Δ_r)	0.5
Mutation probability	0.1		
Unfeasibility ratio (α_I)	0.2		

Table 7: Parameter values for the multiobjective evolutionary algorithms.

5.2. Multiobjective performance and robustness indicators

The goal of these indicators is to compare the robustness and convergence of the Pareto set approximations obtained by the multiobjective algorithms. We used moGrams platform (www.softcomputing.es/mograms) to easily compute and plot the values of the indicators. In fact, reader can access to www.softcomputing.es/mograms/omega to interactively analyze the plots of this paper. For a fair comparison, the three-objective Pareto solutions have been decomposed into bi-objective Pareto solutions. Then, we can compare the convergence of the bi-objective and three-objective algorithms. The robustness information, i.e. the third objective in some algorithms, is usually attached to the Pareto solutions.

We consider the following two well-known multiobjective performance indicators (Zitzler et al. 2000, Deb 2001): a) the unary hyper-volume ratio (*HVR*) (Coello et al. 2007) which measures the quality of a non-dominated solution set approximation returned by an algorithm; and b) the multiplicative I_ϵ indicator (Zitzler et al. 2003), a binary performance indicator that compares two different multiobjective algorithms.

Box-plots will help to represent, for each pair of algorithms, the I_ϵ values of their approximation sets in the 10 runs performed during the experimentation. For a more detailed explanation on the indicators and use of box-plots

for representing them please refer to Chica et al. (2012).

The experimentation also includes attainment surfaces (Fonseca and Fleming 1996) to allow an easy visual comparison of the performance of the algorithms. In addition, we also use the robustness visualization model proposed in Chica et al. (2013) for answering the question about how robust a Pareto front is. This model associates a diameter and an intensity color value to each non-dominated solution which are proportional to the given robustness value (big circles and light color: robust solutions; small circles and intense color: non-robust solutions).

5.3. Results and discussion

In this section we analyze the results obtained by three algorithms for the r-TSALBP-m/A. First, we compare the behavior of the novel adaptive IDEA with respect to the original version of the algorithm. Then, we analyze the robustness and convergence results for the best algorithms.

5.3.1. Performance of the adaptive IDEA

The robustness plot of Figure 4 collects the robustness values of the non-dominated solutions for two instances taken as example. The solutions were generated by both algorithms, the original IDEA and our adaptive version. These robustness values are those defined in the r-TSALBP model by $G(T)$ (Equation 21). Value 0 means that the solution is enough robust according to the required user values ($r_{min}^1(T)$, $r_{min}^2(T)$ and $r_{min}^3(T)$). The robustness achieved by both algorithms in the instances of the figure is almost complete and there are no valuable differences between them. The same occurs for the rest of the instances.

Although the non-dominated solutions generated by the algorithms are robust, it is also important to analyze the convergence performance of the two algorithms. To do this, Table 8 and Figure 6 show the HVR values and I_ϵ box-plot graphics. We can see that the adaptive IDEA converges better to the pseudo-optimal Pareto front as it gets higher HVR values in all the instances but P4. According to the I_ϵ box-plots the analysis is similar. The non-dominated solutions of the adaptive IDEA dominate those generated by the original one in the majority of the instances. The original IDEA is slightly better in instance P4 although the median values (thick lines) are similar.

Last but not least, Figure 5 shows the attainment surfaces of both algorithms tackling instance P6. As seen in the analysis of the performance

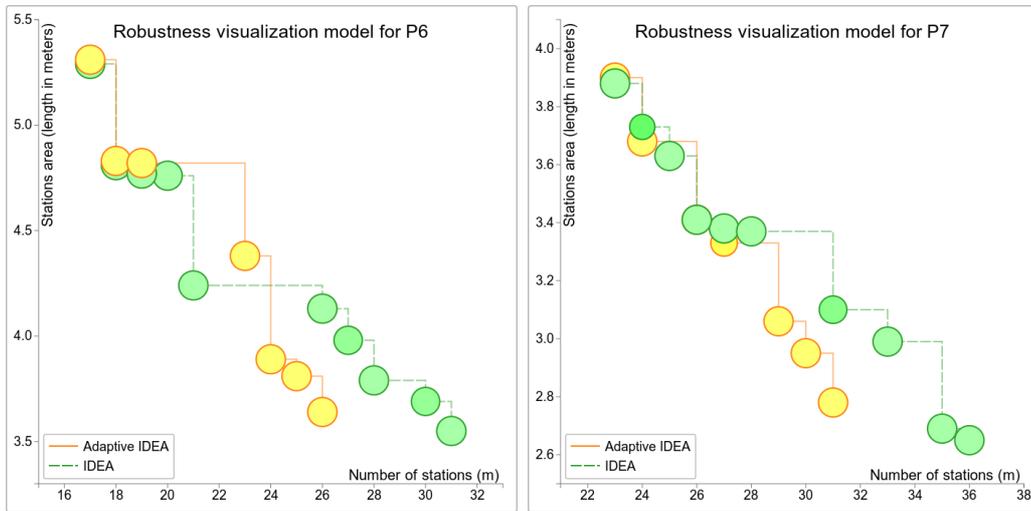


Figure 4: Robustness visualization plots for instances P6 and P7 when comparing both IDEA versions. Bigger circles and less intense colors stand for more robust solutions.

indicators the adaptive IDEA converges better than the original algorithm and the difference is important in the center and bottom-right areas of the Pareto front (solutions with low areas). The original algorithm is getting better solutions in the top-left area. A similar behavior is shown in the attainment surfaces of the remaining problem instances³.

³Reader can interactively access to other instance plots at www.softcomputing.es/mograms/omega

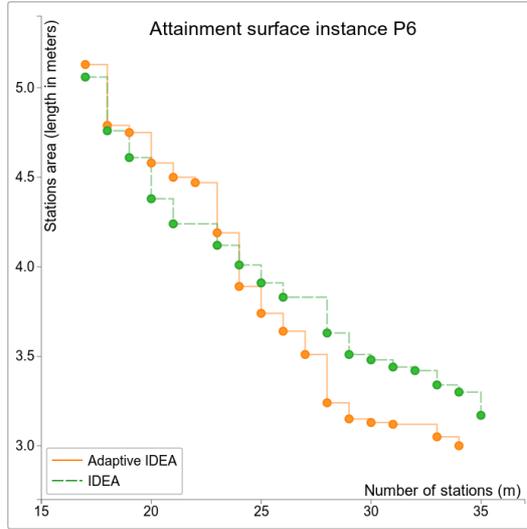


Figure 5: Attainment surface plots for the original and adaptive variants of the IDEA for solving instance P6.

	IDEA		Adaptive IDEA	
	\bar{x}	σ	\bar{x}	σ
P1	0.9071	0.2882	0.9202	0.0403
P2	0.918	0.2877	0.9464	0.0404
P3	0.9106	0.2857	0.9451	0.0494
P4	0.9617	0.2936	0.9571	0.0329
P5	0.8835	0.2783	0.9183	0.0577
P6	0.8644	0.2643	0.8996	0.0746
P7	0.8512	0.255	0.8623	0.0735

Table 8: Mean and standard deviation $\bar{x}(\sigma)$ of the HVR values for the original and adaptive IDEA. Higher values indicate better performance. Bold values correspond to the best results for instance.

5.3.2. Robustness analysis of the algorithms

We will analyze the robustness of the non-dominated solutions provided by the adaptive IDEA and the *TSALBP-NSGA-II* with and without an additional objective for the robustness. We have only included the adaptive version of IDEA in the comparison since, as seen in the previous section, it clearly outperforms the original one.

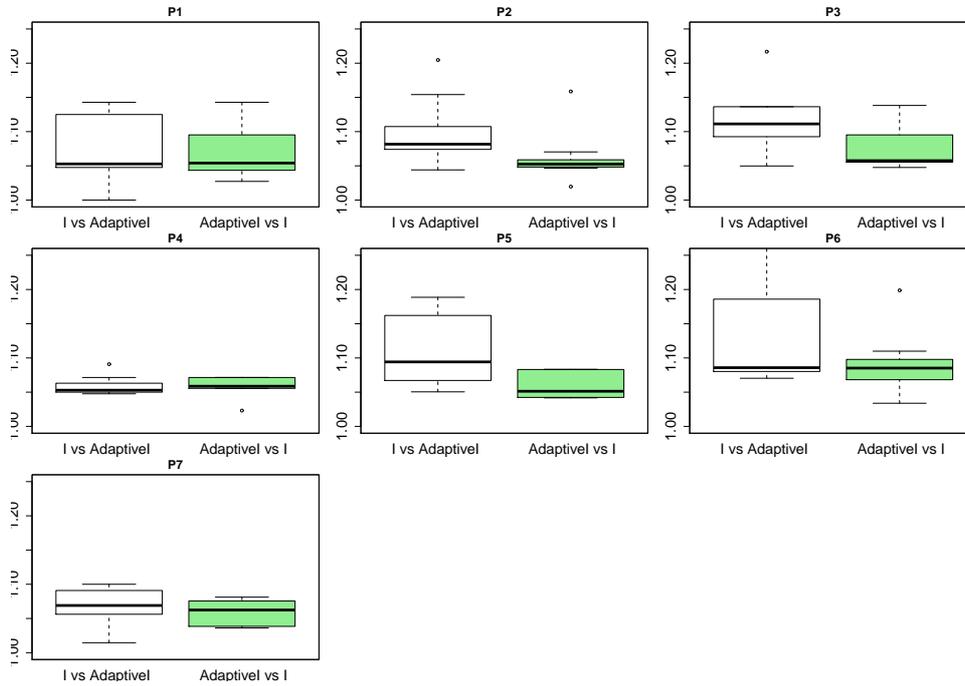


Figure 6: Box-plots representing the I_ϵ values of the two IDEA variants for the seven problem instances.

Figure 7 shows the plots with the robustness values of the algorithms for two instances (P3 and P5) in one run. In these cases we can compare the effect of using robustness as a restriction (adaptive IDEA) and as an additional objective (*TSALBP-NSGA-II*) for obtaining robust solutions. Results show that differences between those two algorithms are minor.

Then, to validate the correct behavior of both methods we compare them against the state-of-the-art method: *TSALBP-NSGA-II* without searching for robust solutions. As expected, both adaptive IDEA and *TSALBP-NSGA-II* with an additional objective are able to obtain much more robust solutions than the conventional *TSALBP-NSGA-II*. This behavior is important because it confirms the benefits of considering robustness during the search process of both methods.

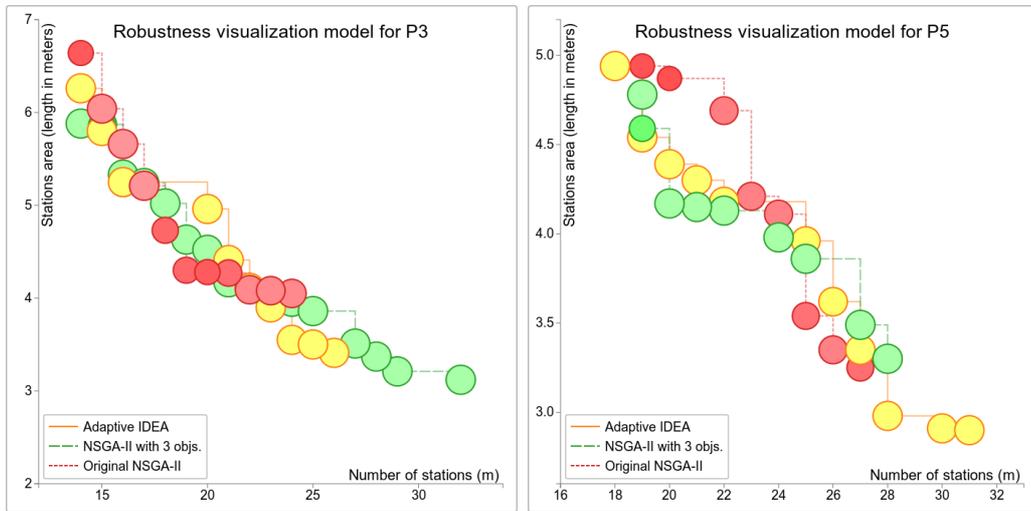


Figure 7: Robustness visualization plots for instances P3 and P5 when comparing the three algorithms. Bigger circles and less intense colors stand for more robust solutions.

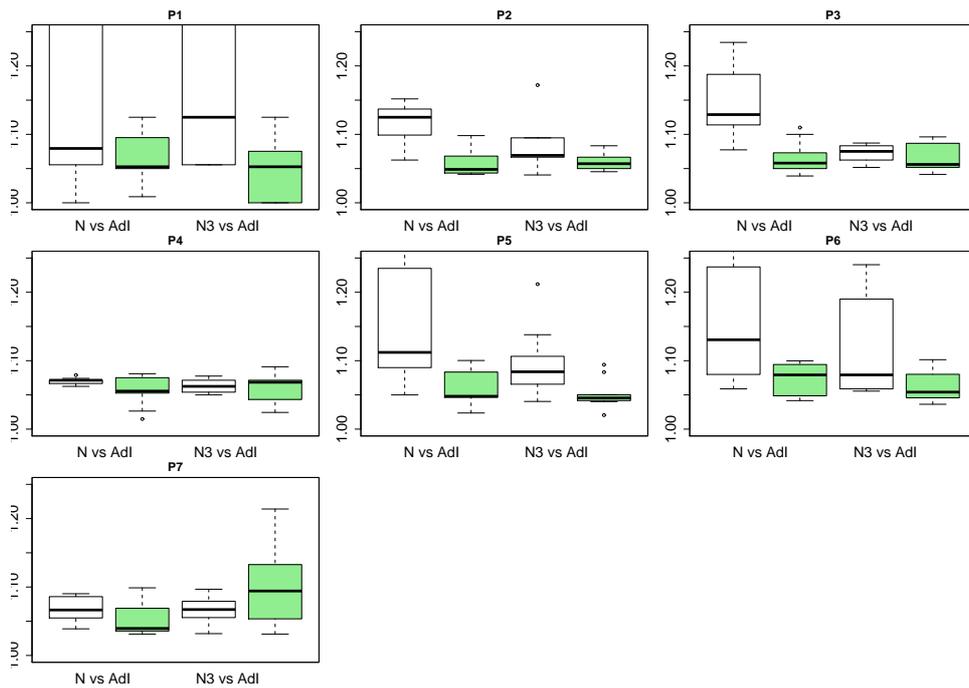


Figure 8: Box-plots representing the I_ϵ values of the *TSALBP-NSGA-II* variants (N and N3) and adaptive IDEA (AdI) for the seven problem instances.

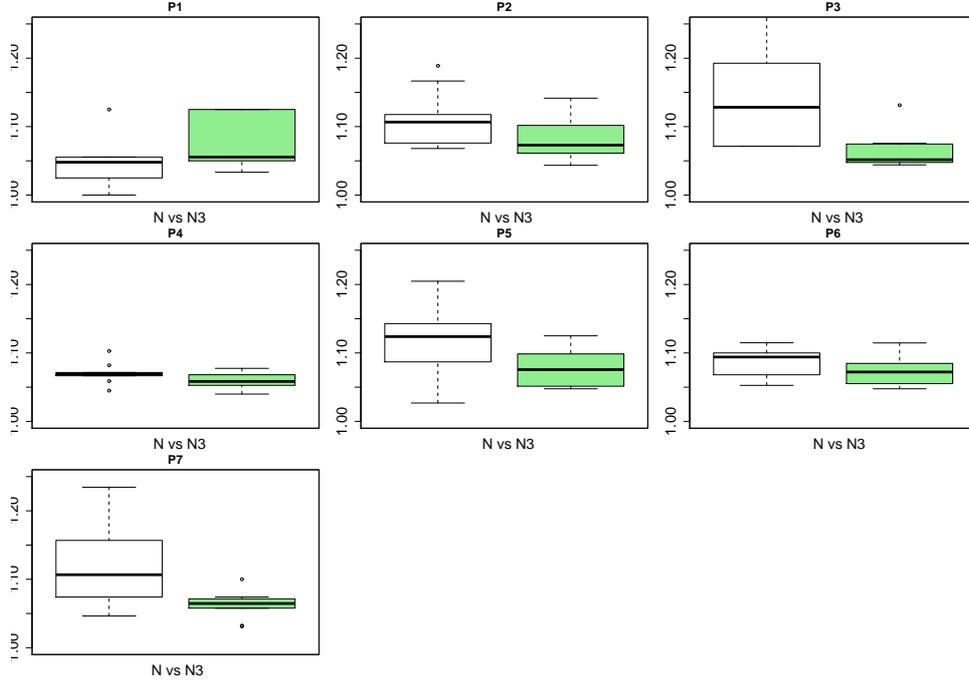


Figure 9: Box-plots representing the I_ϵ values of the two variants of *TSALBP-NSGA-II* (N and N3) for the seven problem instances.

	NSGAII		NSGAII 3Objs.		Adaptive IDEA	
	\bar{x}	σ	\bar{x}	σ	\bar{x}	σ
P1	0.8925	0.2804	0.8747	0.0431	0.9202	0.0403
P2	0.9042	0.2825	0.9206	0.0504	0.9464	0.0404
P3	0.8860	0.2741	0.9307	0.0663	0.9451	0.0494
P4	0.9328	0.2927	0.9501	0.0359	0.9571	0.0329
P5	0.8506	0.2627	0.8836	0.0685	0.9183	0.0577
P6	0.8540	0.2555	0.8698	0.0746	0.8996	0.0746
P7	0.8335	0.2509	0.8859	0.0870	0.8623	0.0735

Table 9: Mean and standard deviation $\bar{x}(\sigma)$ of the *HVR* values for the variants of *TSALBP-NSGA-II* and the adaptive IDEA. Higher values indicate better performance. Bold values correspond to the best results for instance.

5.3.3. Convergence comparison between the approaches

In the previous section we showed the robustness of the returned solutions while in the following paragraphs we will analyze the three algorithms

in terms of convergence. Table 9 shows the HVR indicator values of the algorithms. Figures 8 and 9 present the I_ϵ values by means of box-plots. The first box-plots graph compares the adaptive IDEA with the two variants of the $TSALBP-NSGA-II$ and the second one zooms the comparison just in the two $TSALBP-NSGA-II$.

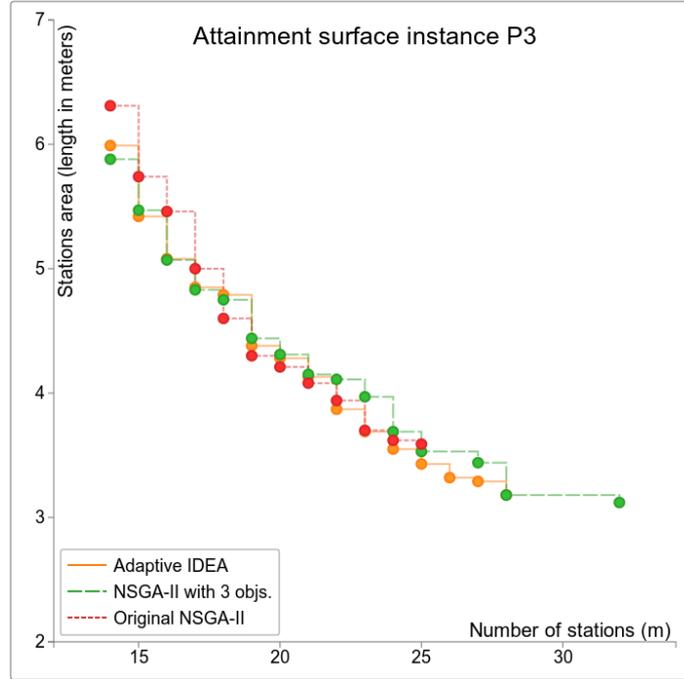


Figure 10: Attainment surface plots when solving instance P3 to compare the $TSALBP-NSGA-II$ with robustness and adaptive IDEA with the original version of the $TSALBP-NSGA-II$.

We can summarize the following conclusions drawn from the analysis of the performance indicators:

- The adaptive IDEA gets better Pareto set approximations than the conventional $TSALBP-NSGA-II$ (no use of robustness during the search process) in all the instances and according to every indicator.
- Similarly, the adaptive IDEA is also better than the $TSALBP-NSGA-II$ with an additional objective for robustness in terms of convergence. The only exception is found in instance P7 where the $TSALBP-NSGA-II$ obtains better results than the adaptive IDEA.

- Although the differences are less important when comparing the two variants of the *TSALBP-NSGA-II*, both indicators also show that the convergence of the variant with an additional objective for robustness is higher. It is only in P1 where the *HVR* and I_ϵ indicators favor the original bi-objective version of the *TSALBP-NSGA-II* without the robustness objective.

To sum up, the developed analysis exhibits that the convergence is higher when adding robustness during the search (*IDEA* and *TSALBP-NSGA-II* with an additional objective). This is a collateral effect produced by the additional diversity that derives from amplifying the search when looking for robust solutions. To illustrate this behavior we have included the attainment surface of Figure 10. In this plot we can see how the algorithms handling robustness information are the only ones that achieve non-dominated solutions in the bottom-right part of the Pareto front.

6. Managerial impact and practical relevance of the study

6.1. Effects of global demand changes on the production system

Changes in demand generate direct discrepancies between the objectives of the production department and the real capacity of the assembly line. This capacity directly depends on the line configuration with respect to the reference production plan (ϵ^0). Discrepancies bring out problems and challenges, not only for the production department, but for the organization and production system of the industrial plant. Concretely, when the global demand varies from the reference demand, the possible mid-term effects on the production system are:

- e1 Increase of the number of workstations (workers and/or linear area of the stations) to accomplish new production plans with a higher global demand than in the reference plan.
- e2 Workforce and/or spatial reduction because of demand losses, avoiding dead times in the assembly line.

We can also find more effects on the production system when, although the global demand remains fixed, there are changes in the mix of products to be assembled:

- e3** Increase/decrease of the number of workers and/or linear area to satisfy temporal and spatial constraints while keeping the current cycle time and area of each workstation.
- e4** Workload variation while keeping the same number of workers, cycle time, and available area.

Effects **e1** and **e2** lead to important production management changes. In fact, increasing the number of workstations needs the enrollment of new workers and their re-assignment. Obviously, there are other short-term actions such as subcontracting external companies, increasing shifts during the weekends, or having sets of extra hours for workers. However, the main focus of our work is to highlight the long-term and internal implications. Therefore, taking into account the need of changing the number of workers, additional training programs for both novice and expert workers might be necessary before getting a fully functional assembly line. The consequence in the productivity of the line and the production system is clear. Besides, practitioners will need to spatially re-distribute the industrial plant and will require more space for the line. When setting up a new spatial distribution a secondary problem shows up. The company will have to handle the movement and placement of heavy tools, robots, shelves, and industrial containers, among others.

Effect **e2** (workforce reduction) is also complex as organizational downsizing is now firmly established as a critical aspect of management practice (Wilkinson 2005). In human resources (HR) terms, there will be a necessary training for workers towards a multi-skilled assembly line profile, and appropriate and legal answers to “redundancies”. Effect **e3** has similar but less dramatic managerial consequences than effects **e1** and **e2**. This is because global demand remains fixed and the temporal and spatial attributes of the tasks are similar for all the products. Both effects, **e3** and **e4**, require additional training for workers and then a productivity loss during production ramp-up until reaching a full capacity utilization of the assembly line.

6.2. Using the metrics and computational methods to measure managerial impact

The r-TSALBP model presents several spatial and temporal functions which are related to the robustness of a line configuration (ψ) with respect to a set of production plans E (Equations 13 to 18). These functions can

also be seen as managerial indicators of the impact of having more or less robust line configurations. Then, these six metrics are able to summarize the impact of the above-mentioned effects in an organization:

m1 (g_c^1 , **Equation 13**) Production plans in E which cannot be accomplished by line configuration ψ^0 because of the cycle time.

m2 (g_c^2 , **Equation 14**) Number of overloaded workstations of the line. That is, the rate of stations for the line configuration ψ^0 with more workload than cycle time c allows.

m3 (g_c^3 , **Equation 15**) Additional processing time for the line configuration ψ^0 to carry out all the production plans in E .

m4 (g_A^1 , **Equation 16**) Production plans in E which cannot be accomplished by line configuration ψ^0 because of exceeding linear area A .

m5 (g_A^2 , **Equation 17**) Number of workstations of the line configuration ψ^0 that need more linear area than the allowed (A).

m6 (g_A^3 , **Equation 18**) Additional linear area generated by the line configuration ψ^0 to carry out all the production plans in E .

The diagram of Figure 11 shows the hierarchical structure of the metrics and their relationships with a departmental organization. Although there are more external implications for the organization we are not considering here the impact on external issues such as suppliers and the supply chain of the production system but only the internal processes. Metrics **m1** and **m4** provide the number of interventions on the assembly line to change its temporal and spatial characteristics. These two metrics alert about potential readjustments that would cause additional works to be re-scheduled in other shifts or during the weekends. Anyhow, these changes may cause production inefficiencies until achieving the regular capacity of the line. These issues are very related with HR actions.

Meanwhile, **m2** and **m5** are useful metrics to find the most problematic workstations because of their temporal and spatial characteristics. First, **m2** shows workstations that, under the conditions of the reference line configuration ψ^0 , need more cycle time to fulfill all the production plans of E . Manufacturing process management technologies can offer the following solutions to solve this issue: a) improve the processing time of the industrial tasks

(methods-time measurement (MTM)), b) request alternative pieces having less processing time during their assembly (product design department), and c) set a working pace over the normal activity of the line (Bautista et al. 2015) within the legal and trade union agreements (process engineering).

m5 is a metric to identify those workstations, under the conditions of the reference line configuration ψ^0 , that need more linear area to satisfy all the production plans of E . In this case, the following managerial solutions can alleviate the problem: a) reduce the linear area required by those pieces and operations involved in the operations (process design and bill of materials (BoM)), b) request smaller pieces to the product design department, and c) consider spatial and temporal windows within the workstations (Bautista et al. 2012) and set a higher working pace if necessary.

Finally, **m3** and **m6** provide temporal-spatial upper and lower limits to restrict changes in operating time and linear area to satisfy all the production plans of E . These two metrics are interdepartmental as they influence the managerial decisions of a high number of departments.

A set of different solutions (multiobjective optimization) is offered to the manager with a measure of the organizational decision impact (six metrics) for each of them:

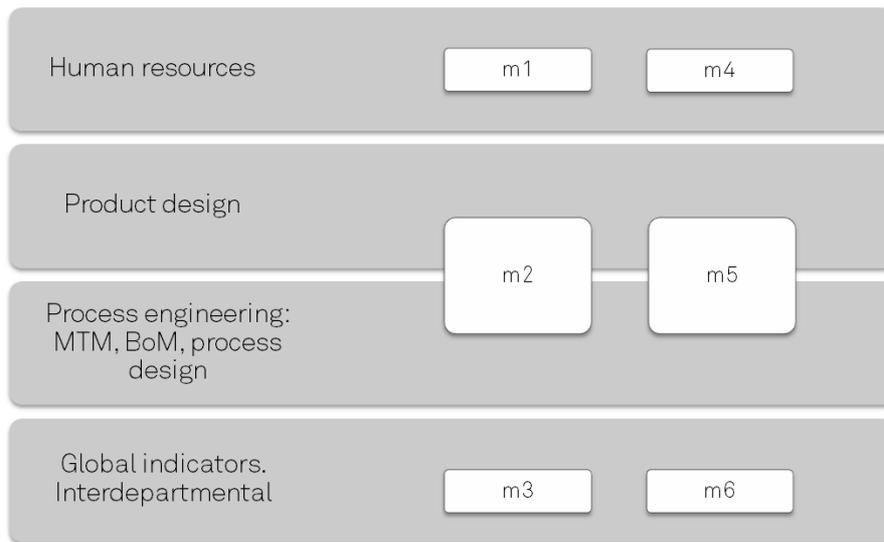


Figure 11: Diagram with the relationships between the robustness metrics and the departments of an illustrative automotive industry.

Some of the latter metrics were used in this contribution to minimize the number of stations m and their area A by different evolutionary multiobjective optimization algorithms. The metrics were included in the model by means of the temporal and spatial secondary functions and were also visualized for the obtained solutions. The multiobjective nature of the model and methods can offer the practitioner with different line configurations and their robustness values in a single run. See for instance Figure 10 that shows different solutions and robustness values for the line configuration.

We proposed two different optimization approaches to use these functions (and associated metrics) in different ways: either as constraints (Equations 8 and 9) or as additional objectives (Equations 21 and 22). In our study, we also proposed a novel adaptive variant of an existing evolutionary multiobjective optimization algorithm, IDEA. The adaptive IDEA outperformed the rest of the algorithms in the majority of the seven problem instances considered in the experimentation. This finding is also relevant for the field as we showed that the adaptive IDEA can effectively be used for solving other models where practitioners need efficient and robust solutions for balancing an assembly line.

The analysis of the experimentation also showed some important conclusions. Regardless the method used for including robustness during the search, the non-dominated solutions obtained almost a null deviation from the user defined robustness. In other words, the Pareto set approximations obtained by the adaptive IDEA were almost composed of robust solutions. Collaterally, the algorithms that included robustness in the search process added more diversity, obtaining better and wider Pareto fronts. The main conclusion is that the best way to solve the r-TSALBP is by using the adaptive IDEA, both in robustness and convergence.

The utilization of these temporal and spatial functions within the adaptive IDEA as an optimization constraint can offer managers a set of non-dominated solutions with more or less emphasis on the six defined metrics. The robustness achievement of the solutions with respect to these metrics provides information about the kind of managerial actions to apply when adopting the specific line configuration. These six metrics are indicators of the implication level and impact on the different departments of the organization and they have a hierarchical nature. Besides, it should be noted that the value of the metrics will influence the horizon of the decision as well as the level of involvement of the departments.

7. Limitations and future work

The results and conclusions of this work are only applicable to the assembly of complex products such as those of the automotive industry and in a western cultural framework (OECD countries). That means product-oriented industrial systems which are able to assemble 4×4, VANs, and trucks, in the case of the automotive industry, by using the same assembly line (mixed-model lines). The r-TSALBP model and the presented results have the following main limitations:

- Assembly lines might follow a mixed-model approach. The tasks of the different models have variations in their temporal and spatial attributes but these attributes are certain and deterministic for each defined scenario (production plan).
- Products and their components have high spatial dimensions (engines, bodywork, vehicles, and so on) with a complex BoM. Then, the linear area available for all the stations must be adjusted to the target models (from 3 to 10 meters, for instance) by avoiding long distances for the workers of the line. Any change with respect to the workload of the line configuration will require the reassignment of the tools because of the high spatial requirements of the components of the products.
- The manufacturing of any model requires high-medium technology installed in the industry plant and some minimum knowledge and skills for workers to finish their tasks. Demand changes force the reassignments of workers that necessarily imply a training process for them before achieving the maximum capacity of the assembly line. This process can take several weeks.
- Cycle times of the assembly line are high, typically between 0.5 and 15 minutes.
- Global demand changes frequently over the course of the year, even monthly. Besides, the production plan changes in some weeks even though the global demand is constant.

In other words, the r-TSALBP model and its managerial relevance do not apply under the following circumstances: a) a process oriented production

system, b) extremely similar models, and c) a constant demand. Additionally, our proposal is not totally necessary if: a) we have low cycle times which imply simple learning process for workers, b) the BoM of the products to be assembled is small, c) the products and their components have low dimensions and do not need demanding linear area, or d) the majority of the tools, equipment, and assistance robots are easily movable and their reassignment is straight-forward. To sum up, this contribution is not useful if changes in the assembly line do not require important changes and they can be easily made.

Future works may focus on two developments: enriching the model to solve some of the practical limitations of the current study and improving the computational methods. For instance, we think it is interesting to improve the TSALBP and r-TSALBP models with more realistic industrial features such as ergonomic factors (Bautista et al. 2013). Additionally, we would like to study other approaches for obtaining robust solutions through the search process of an evolutionary multiobjective algorithm. The visualization and decision-making process is, in our opinion, another important and promising line in the area. Then, a future work will be to provide practitioners with a global visualization framework for representing assembly line configurations and relationships between different alternative solutions.

Acknowledgments

This work is supported by Principality of Asturias (Programa Asturias 2014-2015, reference CT14-05-2-04) and Spanish Ministerio de Economía y Competitividad under SOCOVIFI2 (TIN2012-38525-C02-01, TIN2012-38525-C02-02) and PROTHIUS-III (DPI2010-16759), both including EDRF funding. Authors also wish to thank the anonymous referees for their comments and suggestions and Guillermo Chica from Airbus Military for his suggestions on the managerial impact of the paper.

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., Alfaro, R., Batalla, C., 2015. Modeling and solving the mixed-model sequencing problem to improve productivity. *International Journal of Production Economics* 161, 83–95.

- Bautista, J., Batalla, C., Alfaro, R., 2013. Incorporating ergonomics factors into the tsalbp, in: *Advances in Production Management Systems. Competitive Manufacturing for Innovative Products and Services*. Springer. volume 397, pp. 413–420.
- Bautista, J., Cano, A., Alfaro, R., 2012. Models for MMSP-W considering workstation dependencies: A case study of Nissan’s Barcelona plant. *European Journal of Operational Research* 223, 669–679.
- 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.
- Bengtsson, J., Olhager, J., 2002. The impact of the product mix on the value of flexibility. *Omega* 30, 265–273.
- Beyer, H., Sendhoff, B., 2007. Robust optimization - a comprehensive survey. *Computer Methods in Applied Mechanics and Engineering* 196, 3190–3218.
- 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., Cerdón, 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., Cerdón, O., Damas, S., Bautista, J., 2012. Multiobjective memetic algorithms for time and space assembly line balancing. *Engineering Applications of Artificial Intelligence* 25, 254–273.
- Chica, M., Cerdón, O., Damas, S., Bautista, J., 2013. A robustness information and visualization model for time and space assembly line balancing under uncertain demand. *International Journal of Production Economics* 145, 761–772.
- Chica, M., Cerdón, O., Damas, S., Bautista, J., 2014. NTIGen: a software for generating Nissan based instances for time and space assembly line balancing, in: *Managing Complexity: challenges for Industrial Engineering and Operations Management*. Springer, Berlin, Germany. volume 2 of *Lecture Notes in Management and Industrial Engineering*, pp. 121–128.
- Coello, C.A., Lamont, G.B., Van Veldhuizen, D.A., 2007. *Evolutionary Algorithms for Solving Multi-objective Problems* (2nd edition). Springer.
- Deb, K., 2001. *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley.

- Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6, 182–197.
- Dolgui, A., Kovalev, S., 2012. Scenario based robust line balancing: Computational complexity. *Discrete Applied Mathematics* 160, 1955–1963.
- Fonseca, C.M., Fleming, P.J., 1996. On the performance assessment and comparison of stochastic multiobjective optimizers, in: *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN)*, Berlin, Germany. pp. 584–593.
- Gao, D., Zhao, X., Geng, W., 2014. A delay-in-payment contract for Pareto improvement of a supply chain with stochastic demand. *Omega* 49, 60–68.
- 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.
- Hazır, Ö., Dolgui, A., 2013. Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering* 65, 261–267.
- Li, J., Gao, J., 2014. Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research* 52, 3552–3567.
- Ogüzsoy, C.B., Güven, S., 2007. Robust portfolio planning in the presence of market anomalies. *Omega* 35, 1–6.
- Papakostas, N., Pintzos, G., Giannoulis, C., Nikolakis, N., Chryssolouris, G., 2014. Multi-criteria assembly line design under demand uncertainty. *Procedia CIRP* 25, 86–92.
- Poon, P.W., Carter, J.N., 1995. Genetic algorithm crossover operators. *Computers & Operations Research* 22, 135–147.
- Rada-Vilela, J., Chica, M., Cordon, O., Damas, S., 2013. A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Applied Soft Computing* 13, 4370–4382.
- Roy, B., 2010. Robustness in operational research and decision aiding: A multifaceted issue. *European Journal of Operational Research* 200, 629–638.
- Saif, U., Guan, Z., Wang, B., Mirza, J., 2014. Pareto lexicographic α -robust approach and its application in robust multi objective assembly line balancing problem. *Frontiers of Mechanical Engineering* 9, 257–264.

- 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.
- Simaria, A.S., Zanella de Sá, M., Vilarinho, P.M., 2009. Meeting demand variation using flexible u-shaped assembly lines. *International Journal of Production Research* 47, 3937–3955.
- Singh, H.K., Isaacs, A., Ray, T., Smith, W., 2008. Infeasibility driven evolutionary algorithm (IDEA) for engineering design optimization, in: *AI 2008: Advances in Artificial Intelligence*. Springer, pp. 104–115.
- Wang, X.J., Curry, D.J., 2012. A robust approach to the share-of-choice product design problem. *Omega* 40, 818–826.
- Wilkinson, A., 2005. Downsizing, rightsizing or dumbsizing? quality, human resources and the management of sustainability. *Total Quality Management and Business Excellence* 16, 1079–1088.
- Xu, W., Xiao, T., 2009. Robust balancing of mixed model assembly line. *COMPEL: The International Journal for Computation and Mathematics in Electrical and Electronic Engineering* 28, 1489–1502.
- Xu, W., Xiao, T., 2011. Strategic robust mixed model assembly line balancing based on scenario planning. *Tsinghua Science & Technology* 16, 308–314.
- Yan, S., Tang, C.H., 2009. Inter-city bus scheduling under variable market share and uncertain market demands. *Omega* 37, 178–192.
- Zitzler, E., Deb, K., Thiele, L., 2000. Comparison of multiobjective evolutionary algorithms: Empirical results. *Evolutionary Computation* 8, 173–195.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C.M., da Fonseca, V.G., 2003. Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation* 7, 117–132.