Adaptive IDEA for Robust Multiobjective Optimization. Application to the r-TSALBP-m/A

Manuel Chica, Joaquín Bautista, Óscar Cordón and Sergio Damas

Abstract—Robust optimization tries to find flexible solutions when solving problems with uncertain scenarios and vague information. In this paper we present a multiobjective evolutionary algorithm to solve robust multiobjective optimization problems. This algorithm is a novel adaptive method able to evolve separate populations of robust and non-robust solutions during the search. It is based on the infeasibility driven evolutionary algorithm (IDEA) and uses an additional objective to evaluate the robustness of the solutions. The original and adaptive IDEAs are applied to solve the r-TSALBP-m/A, an assembly line balancing model that considers a set of demand production plans and includes robustness functions for measuring the temporal overloads of the stations of the assembly line with respect to the plans. Our results show that the proposed adaptive IDEA gets more robust non-dominated solutions for the problem. Also, we show that, for the case of the r-TSALBP-m/A, we can obtain Pareto fronts with a higher convergence by using the adaptive version of the algorithm.

I. Introduction

Real-world applications normally involve uncertainties because of operating conditions or manufacturing process [1]. Robust optimization tries to find flexibility by its way of solving the latter problems. Particularly, flexibility is an important asset to manufacturing firms to respond to changes in the environment [2] and this flexibility applies to the many operation research problems and industrial fields such as automotive industry, assembly line balancing, or scheduling. Robustness can be applied to many components in an optimization process: noise in constraints, objective function, or uncertainties in data variables [3], [4], [5].

In this work we integrate the concept of robust solutions [4], [5] during the search of an evolutionary multiobjective optimization (EMO) method to directly provide practitioners with robust solutions for their problems. The original infeasibility driven evolutionary algorithm (IDEA) [6] was explicitly designed for industrial constrained optimization problems. In the current contribution we propose a novel variant of the IDEA, an extension of the original version to search for robust solutions during the search. The novel IDEA is adaptive and 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 at every generation.

In order to show the potential of the adaptive IDEA to deal with robust optimization problems we apply it to a well-known industrial problem, the assembly line balancing (ALB) [7], [8]. 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 the ALB is one usual and difficult problem which determine how these tasks are efficiently assigned to the stations by fulfilling certain restrictions. The simple assembly line balancing problem (SALBP) [9], [10] 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.

The r-TSALBP is a new SALBP model which searches for the most efficient assembly line configurations when demand changes. 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 robustness functions which are based on the overloads for each station and production plan. Concretely in this work we apply the original and adaptive IDEA to the r-TSALBP-m/A model which optimizes m (the number of stations) and A (the area of the stations) when having production plans with different operation times for the tasks of the assembly line.

The experimentation comprises the performance evaluation of both IDEAs in a set of seven instances generated by the NTIGen software [11]. These instances belong to a diverse set of r-TSALBP-m/A instances that incorporates the real data and industrial features of the Nissan industry plant of Barcelona. Both methods are compared to analyze the effect of including the robustness computation during the search process and how the adaptive variant performs. The analysis employs commonly used multiobjective performance indicators and robustness graphical representations of the non-dominated solutions.

In Section II we discuss the introduction of robustness in multiobjective optimization and production problems. Our adaptive IDEA proposal is given in Section III. The utility and mathematical description of the r-TSALBP-m/A is detailed in Section IV. The specific design of both IDEAs for solving the r-TSALBP-m/A is presented in Section V and the analysis of results in Section VI. Finally, the conclusions of the study and some future works are discussed in Section VII.

II. Robust optimization in EMO and assembly line balancing

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 [12], vehicle routing problem [13], and product design problems [14]. The search for optimal robust designs often appears as a multi-criteria decision making (MCDM) problem optimizing conditional expectation and variance. For example, one of the proposals in this line was the multiobjective six sigma of [15].

The work of [16] was the first and one of the most important contributions in introducing robustness in EMO. Authors defined 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, [16] suggested 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 [17]: 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 [18], [19], [20]. An example of a robust optimization model for a multi-site production planning problem was developed in [21]. In this work, authors assumed a future economic scenario with an associated probability. An optimal production plan less sensitive to the change in the noisy and uncertain data was returned by using a stochastic non-linear programming model.

Specifically for ALB we can find the work of [22] where authors proposed novel robustness functions and a graphical representation to respectively measure and represent how robust the assembly line configuration is. The values of these functions were computed once a multiobjective optimization method obtained a non-dominated solutions set. Other ways of considering uncertainty in ALB were by assuming that task times are uncertain and not deterministic [23], [24].

[25] proposed an ALB model with uncertain operation execution times. Operation execution times were uncertain in the sense that their sets belonged to a given set of scenarios. Following this research line, [26] recently presented two robust SALBP-2 models having interval uncertainty for operation time.

III. Adaptive IDEA

IDEA [6] is a multiobjective evolutionary algorithm for dealing with constrained optimization problems. In [27] we considered the use of IDEA to solve the conventional TSALBP. However, in this work we propose a novel approach based on the IDEA but to be adaptive and to use robustness measures as additional constraints. The goal of the new adaptive design of IDEA is to be able to distinguish between robust and non-robust solutions during the search.

The design of the original IDEA emphasizes the search for optimal solutions near the constraint boundaries by maintaining and evolving a small proportion of unfeasible solutions [6]. One of the strengths of this algorithm is to divide solutions (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 or lower convergence to the optimal Pareto front depending on the α_I value.

Our adaptive IDEA incorporates a novel mechanism to tune the latter fraction value by calculating the global robustness of the non-dominated solutions. 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 P as the ratio of solutions that are robust according to the problem specific measure $m(z)$ where z is a non-dominated solution. Then, for a given a Pareto set P we can calculate its robustness ratio $r(P)$ as in Equation 1:

$$r(P) = \frac{1}{|P|} \sum_{\forall z \in P} (m(z) = 0). \tag{1}$$

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 non-robust 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 non-robust 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 non-robust solutions. Finally, the adaptive IDEA uses the process defined by Equation 2 to adapt the α_I parameter at iteration t within the merge process of the robust and non-robust (feasible and unfeasible) solutions.

$$\alpha_I^t = \begin{cases} \alpha_I^{t-1} - \Delta_\alpha \alpha_I^{t-1}, & \text{if } r(P^t) \leq \tau_r, \\ \alpha_I^{t-1} + \Delta_\alpha \alpha_I^{t-1}, & \text{if } r(P^t) > \tau_r, \end{cases} \tag{2}$$

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 robust and non-robust individuals

as usual and the flow of the algorithm continues as in the original version of the IDEA.

IV. DESCRIPTION OF THE R-TSALBP-M/A

In this section we introduce the ALB problem and the r-TSALBP formulation. Then, in Sections IV-A and IV-B we briefly describe the mathematical formulation and robustness measures of the problem to be tackled by our algorithms.

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 ($k = 1, 2, ..., m$) 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.

TSALBP [28], [29] is a family of problems that focuses on grouping tasks in workstations by an efficient way and introduces spatial features. It means that 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 $A = max_{k=1,2,...,m} A_k$. 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 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).

All the latter models assume the balance of the assembly lines when producing mixed products. Nevertheless, product demands are not usually fixed and certain. 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. The r-TSALBP-m/A is an ALB model that integrates the concept of robust solutions by defining a set of production plans (scenarios) by minimizing m and A (see Figure 1 for a graph defining r-TSALBP).

A. r-TSALBP-m/A mathematical formulation

In r-TSALBP-m/A we define $i \in I$ as the set of product types to be assembled and 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.
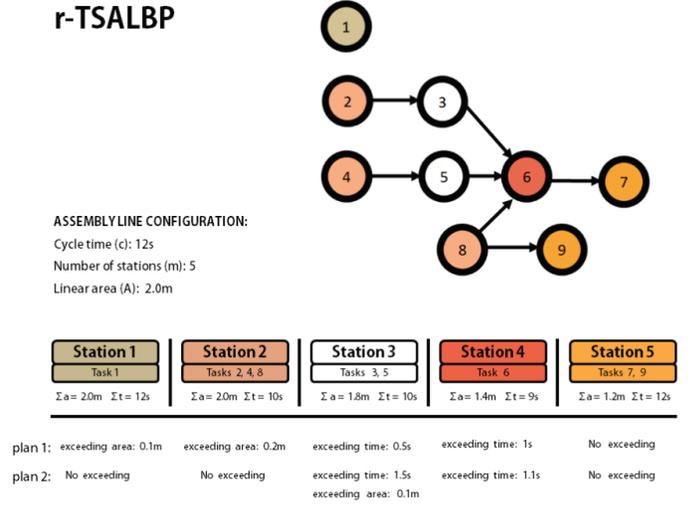

Fig. 1. Graphical representation of a r-TSALBP example. The optimization problem must configure a robust tasks-stations assignment when having different production plans.

Given a production plan $\varepsilon \in E$, defined by a demand vector $\overrightarrow{d}_\varepsilon = (d_{1\varepsilon}, d_{2\varepsilon}, ..., d_{|I|\varepsilon})$, we can determine the average processing time of task $j \in J$ for this plan ε by Equation 3[1]:

$$\overline{t}_{j\varepsilon} = \frac{1}{D_\varepsilon} \sum_{i=1}^{|I|} t_{ji} d_{i\varepsilon}, \qquad (3)$$

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

The used variables of the r-TSALBP-m/A model belong to the set of parameters (Table I) and variables (Table II) of the r-TSALBP model specification. Table III shows the associated restrictions.

TABLE I. PARAMETERS OF THE R-TSALBP-M/A MODEL

J	Set of all the elementary processing tasks of the line ($j = 1, .., |J|$)
n	Number of tasks of the line: $n = |J|$
K	Set of workstations ($k = 1, .., |K|$)
c	Cycle time (it can also be a variable)
E	Set of demand production plans ($\varepsilon = 1, .., |E|$)
$\overline{t}_{j\varepsilon}$	Average processing time of the elementary task $j \in J$ (measured at normal work pace) in production plan $\varepsilon \in E$
a_j	Linear area (length) required to perform task $j \in J$
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	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. $\Delta^c = \gamma_c c$

Equations 10 to 11 define the main objective functions of the r-TSALBP-m/A model. The first equation represents the number of stations of the line configuration and the third one

---

[1] We assume the required area of all the tasks $j \in J$ remain constant for all the plans $\in E$

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)
$y_{k\varepsilon}^c$	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

Binary condition of the station-task assignment variable:

$$x_{jk} \in \{0, 1\}, \qquad (j = 1, .., |J|; k = 1, .., |K|) \qquad (4)$$

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

$$y_{k\varepsilon}^c \in \{0,1\}, y_{k\varepsilon}^A \in \{0,1\} \qquad (k = 1, .., |K|; \varepsilon = 1, .., |E|) \quad (5)$$

Every task must be assigned to just one single station:

$$\sum_{k=1}^{|K|} x_{jk} = 1, \qquad (j = 1, .., |J|) \qquad (6)$$

Every station must contain at least one task:

$$\sum_{j=1}^{|J|} x_{jk} \geq 1, \qquad (k = 1, .., |K|) \qquad (7)$$

The assignment cannot violate the immediate precedence relations:

$$\sum_{k=1}^{|K|} k(x_{ik} - x_{jk}) \leq 0, \qquad (i \in P_j, j = 1, .., |J|) \qquad (8)$$

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

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

the available linear area of the stations of the line. The second equation are calculated using the reference production plan ε^0.

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

$$f^2(x) = A = \max_{k \in K} \{\sum_{j=1}^{|J|} a_j x_{jk}\}. \qquad (11)$$

B. Temporal robustness functions

Furthermore, the r-TSALBP-m/A formulation adds temporal functions, normalized to $[0, 1]$. The three functions are the following:

- Rate of overloaded production plans with respect to the allowed workload time (Equation 12).

$$g_c^1 = \frac{1}{|E|} \sum_{\varepsilon=1}^{|E|} \max_{k \in K} y_{k\varepsilon}^c. \qquad (12)$$

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

$$g_c^2 = \frac{1}{m} \sum_{k=1}^{|K|} \max_{\varepsilon \in E} y_{k\varepsilon}^c. \qquad (13)$$

- Exceeding processing time of the stations in all the plans with respect to the maximum exceeding time and the number of overloaded stations (Equation 14).

$$g_c^3 = g_c^3(x) = \frac{1}{\Delta^c \sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} y_{k\varepsilon}^c}$$
$$\sum_{\varepsilon=1}^{|E|} \sum_{k=1}^{|K|} (\max\{0, \sum_{j=1}^{|J|} \bar{t}_{j\varepsilon} x_{jk} - c\}). \quad (14)$$

The robustness measures with respect to the processing times of the stations and the productions plans are defined by Equation 15 as they limit the values of the latter r-TSALBP-m/A temporal functions of the model (Equations 12 to 14).

$$g_c^1 \leq \tilde{g}_c^1; g_c^2 \leq \tilde{g}_c^2; g_c^3 \leq \tilde{g}_c^3, \qquad (15)$$

where $\{\tilde{g}_c^1, \tilde{g}_c^2, \tilde{g}_c^3\}$ are parameters defined in $[0, 1]$ that restrict the temporal functions (g_c). 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 $\tilde{g}_c^3 = 1 - \tilde{r}_c^3$. These measures can also be put together by a single value using combination weights as defined by Equation 16.

$$G_c(x) = \sum_{i=1}^{3} (\mu_c^i \max\{0, g_c^i - \tilde{g}_c^i\}), \qquad (16)$$

V.   ADAPTATION OF THE IDEAS FOR SOLVING THE R-TSALBP-M/A

In this section we define the design of the original and adaptive IDEA for solving the r-TSALBP-m/A. Both IDEAs have the same representation (Section V-A) and genetic operators (Sections V-B and V-C) for solving the problem (see [29] for more details) but they differ in the adaptive mechanism, only present in the proposed adaptive IDEA.

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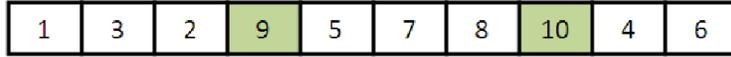
Fig. 2. Variable coding scheme of the chromosome for an eight-task example. Colored genes denote separators which always have integer values greater than the total number of tasks.
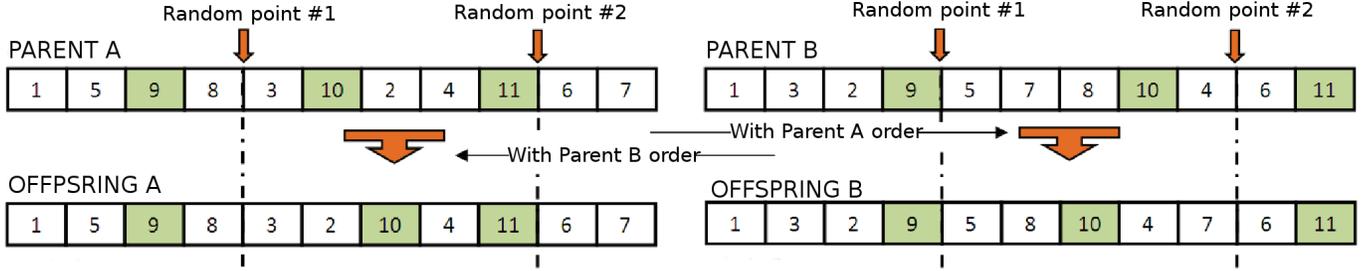

Fig. 3. Variable-length crossover of the robust IDEA. Separators are the colored genes.

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. See Figure 2 for the representation of the chromosome using separators.

B. Crossover operator

The crossover operator is based on a classical order-based one: the PMX [30]. PMX generates two offspring from two parents by means of the following procedure: a) selection of two random cut 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.

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. Figure 3 shows a graph to illustrate the crossover process

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

VI. EXPERIMENTS

In this section we analyze the results obtained by both IDEA variants when solving the r-TSALBP-m/A. First, we give the details of the experimentation (Section VI-A) and then, we compare the behavior of the novel adaptive IDEA with respect to the original version of the algorithm (Section VI-B).

A. Experimental setup

The experimentation uses a set of seven real-like TSALBP instances and 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 and all the algorithms were launched in the same computer: Intel Xeon^{TM} 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 IV shows the parameter values of the model and IDEAs.

We consider two well-known multiobjective performance indicators [31], [32]: a) the unary hyper-volume ratio (HVR) [33] which measures the quality of a non-dominated solution set approximation returned by an algorithm; and b) the multiplicative I_ϵ indicator [34], 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 [35].

Additionally, we also use the robustness visualization model proposed in [22] for answering the question about how robust a Pareto front is. This model associates a diameter and

Parameter	Value
IDEAs	
Population size	100
Crossover probability	0.8
Mutation probability	0.1
Unfeasibility ratio (α_I)	0.2
Adaptive IDEA	
Δ_α (step value for α_I)	0.01
robustness ratio threshold (τ_r)	0.5
r-TSALBP-m/A model	
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 $\{\mu_c^1, \mu_c^2, \mu_c^3\}$	$\{0.2, 0.4, 0.4\}$

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).

B. Analysis of the results

The robustness plot of Figure 4 collects the robustness values of the non-dominated solutions for P7 instance. 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. For example, 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 instance of the Figure 4 is high although the original IDEA obtains some non-dominated solutions that are less robust. The same occurs for the rest of the instances.

It is also important to analyze the convergence performance of the two algorithms. To do this, Table V shows the HVR values and Figure 7 contains the I_ϵ box-plots comparing both algorithms. 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. The conclusions from the analysis of the I_ϵ box-plots are 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 when tackling instance P6. Figure 6 does the same but for instance P4. As previously commented with respect to the analysis of the performance indicators, the adaptive IDEA converges better than the original algorithm. The difference is important in the top-left and bottom-right areas of the Pareto front (extreme solutions of the Pareto front). The worst performing instance for the adaptive IDEA, instance P4, is shown in Figure 6. In this case, the original version presents a higher convergence in the central area of the Pareto front.
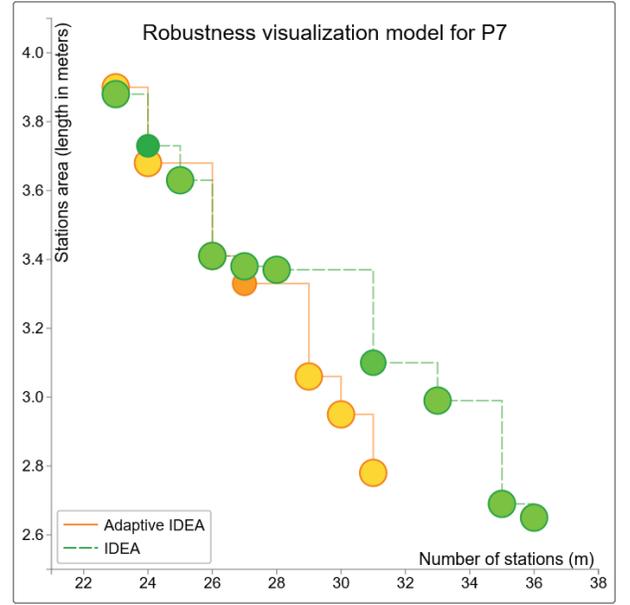

Fig. 4.   Robustness visualization plots for P7 instance when comparing both IDEA versions. Bigger circles and less intense colors stand for more robust solutions.
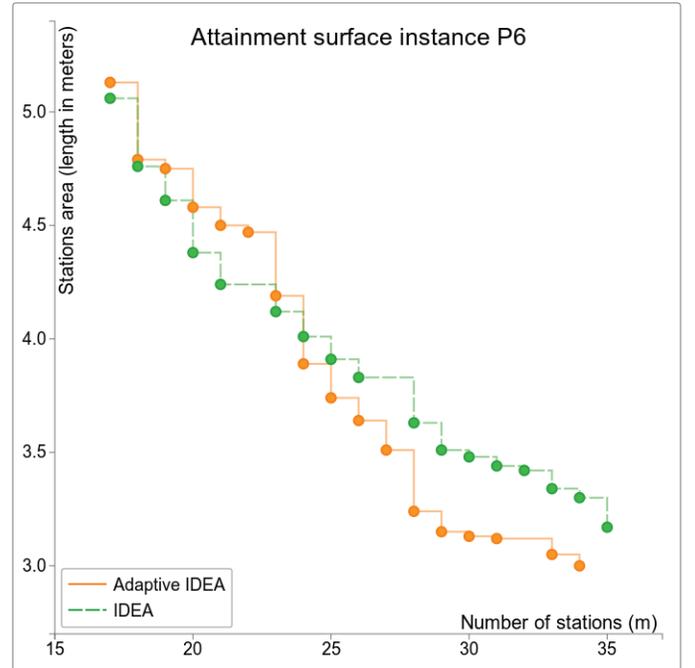

Fig. 5.   Attainment surface plots for the original and adaptive variants of the IDEA for solving instance P6.

VII.   CONCLUDING REMARKS

Introducing robustness in the MCDM process is very valuable for decision makers since they are provided with solutions that are flexible when the initial conditions of the problem change. In the case of the tackled r-TSALBP-m/A problem, these changing initial conditions are the future demand plans of the products to be assembled in an assembly line.

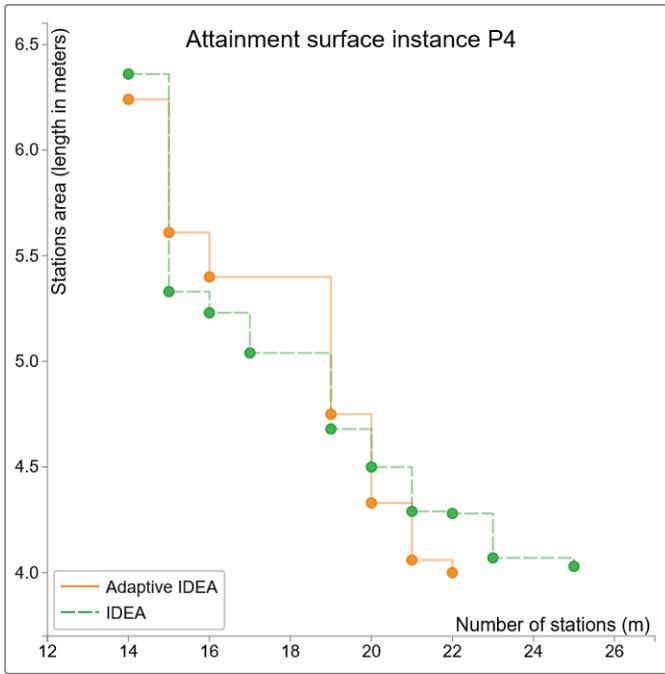In our study, we propose a novel adaptive variant of an

Fig. 6. Attainment surface plots for the original and adaptive variants of the IDEA for solving instance P4.

TABLE V. 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.

	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

existing evolutionary multiobjective optimization algorithm, IDEA. We apply it for solving the r-TSALBP-m/A model which takes into account the exceeding processing time of the stations of the configuration line for all the available production plans of the industry plant. r-TSALBP adds temporal functions and limitations for them to be considered as the robustness of the configuration line with respect to a reference plan.

Our adaptive IDEA manages robustness as a restriction and improves the original IDEA for solving the industrial problem both in robustness and optimization performance. Therefore, its convergence is shown to be better than the original version in the majority of the studied problem instances. We also observed that the adaptive IDEA, which includes robustness in the search process, adds more diversity and thus, obtains better and wider Pareto fronts. Therefore, the main conclusion is that the best way to solve the r-TSALBP-m/A is by using the adaptive IDEA, both in robustness and convergence.

Future works may focus on: a) studying how well adaptive IDEA behaves when solving other logistic and scheduling problems such as car sequencing problem and b), enriching the r-TSALBP model with some more realistic industrial features such as ergonomic factors.

REFERENCES

[1] K. Miettinen, K. Deb, J. Jahn, W. Ogryczak, K. Shimoyama, and K. Vetschera, "Future challenges," in *Multiobjective Optimization. Interactive and Evolutionary Approaches*, ser. Lecture Notes in Computer Science. Springer, 2008, vol. 5252, pp. 435–461.

[2] J. Bengtsson and J. Olhager, "The impact of the product mix on the value of flexibility," *Omega*, vol. 30, no. 4, pp. 265–273, 2002.

[3] B. Roy, "A missing link in OR-DA: Robustness analysis," *Foundations of Computing and Decision Sciences*, vol. 23, pp. 141–160, 1998.

[4] ——, "Robustness in operational research and decision aiding: A multi-faceted issue," *European Journal of Operational Research*, vol. 200, no. 3, pp. 629–638, 2010.

[5] H. Beyer and B. Sendhoff, "Robust optimization - a comprehensive survey," *Computer Methods in Applied Mechanics and Engineering*, vol. 196, no. 33–34, pp. 3190–3218, 2007.

[6] H. K. Singh, A. Isaacs, T. Ray, and W. Smith, "Infeasibility driven evolutionary algorithm (IDEA) for engineering design optimization," in *AI 2008: Advances in Artificial Intelligence*. Springer, 2008, pp. 104–115.

[7] N. Boysen, M. Fliedner, and A. Scholl, "A classification of assembly line balancing problems," *European Journal of Operational Research*, vol. 183, no. 2, pp. 674–693, 2007.

[8] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solution approaches," *International Journal of Production Economics*, vol. 142, no. 2, pp. 259–277, 2013.

[9] A. Scholl, *Balancing and Sequencing of Assembly Lines (2nd. Edition)*. Physica-Verlag, Heidelberg, 1999.

[10] A. Scholl and C. Becker, "State-of-the-art exact and heuristic solution procedures for simple assembly line balancing," *European Journal of Operational Research*, vol. 168, no. 3, pp. 666–693, 2006.

[11] M. Chica, O. Cordón, S. Damas, and J. Bautista, "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*, ser. Lecture Notes in Management and Industrial Engineering. Berlin, Germany: Springer, 2014, vol. 2, pp. 121–128.

[12] C. B. Oğuzsoy and S. Güven, "Robust portfolio planning in the presence of market anomalies," *Omega*, vol. 35, no. 1, pp. 1–6, 2007.

[13] K. C. Tan, C. Y. Cheong, and C. K. Goh, "Solving multiobjective vehicle routing problem with stochastic demand via evolutionary computation," *European Journal of Operational Research*, vol. 177, pp. 813–839, 2007.

[14] X. J. Wang and D. J. Curry, "A robust approach to the share-of-choice product design problem," *Omega*, vol. 40, no. 6, pp. 818–826, 2012.

[15] K. Shimoyama, A. Oyama, and K. Fujii, "A new efficient and useful robust optimization approach - design for multi-objective six sigma," in *IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, Piscataway, 2005, pp. 950–957.

[16] K. Deb and H. Gupta, "Introducing robustness in multi-objective optimization," *Evolutionary Computation*, vol. 14, no. 4, pp. 463–494, 2006.

[17] J. Ferreira, C. M. Fonseca, J. A. Covas, and A. Gaspar-Cunha, "Evolutionary multi-objective robust optimization," in *Advances in Evolutionary Algorithms (www.i-techonline.com)*. Vienna, Austria: InTech, 2008, pp. 261–278.

[18] R. Scheffermann, M. Bender, and A. Cardeneo, "Robust solutions for vehicle routing problems via evolutionary multiobjective optimization," in *IEEE Congress on Evolutionary Computation (CEC)*, vol. 1, Trondheim, Norway, 2009, pp. 1605–1612.

[19] K. C. Tan, C. Y. Cheong, and C. K. Goh, "Robust transportation network design under demand uncertainty," *Computer-Aided Civil and Infrastructure Engineering*, vol. 22, pp. 6–18, 2007.

[20] Y. S. Ong, P. B. Nair, and K. Y. Lum, "Max-min surrogate-assisted evolutionary algorithm for robust design," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 4, pp. 392–404, 2006.
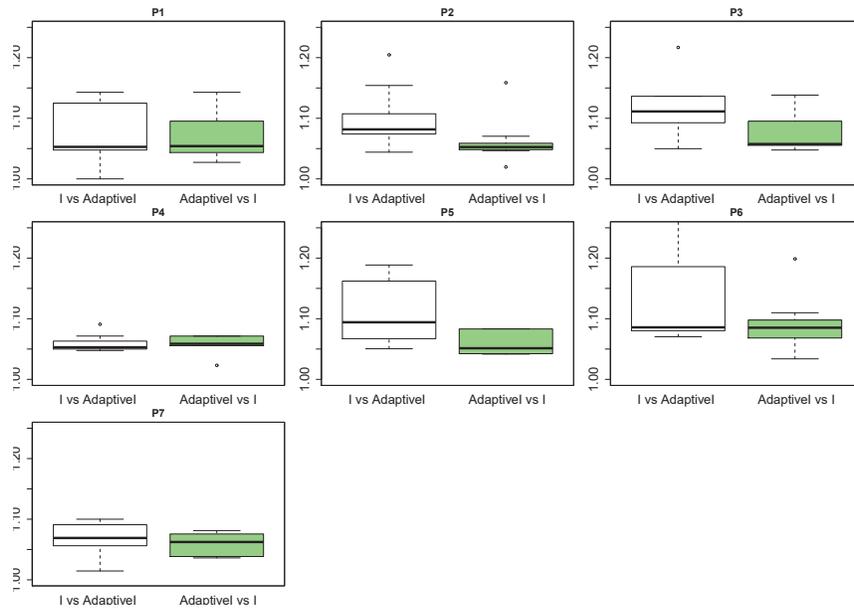
Fig. 7. Box-plots representing the I_ϵ values of the two IDEA variants for the seven problem instances.

[21] S. C. H. Leung, S. O. S. Tsang, W. L. Ng, and Y. Wu, "A robust optimization model for multi-site production planning problem in an uncertain environment," *European Journal of Operational Research*, vol. 181, pp. 224–238, 2007.

[22] M. Chica, O. Cordón, S. Damas, and J. Bautista, "A robustness information and visualization model for time and space assembly line balancing under uncertain demand," *International Journal of Production Economics*, vol. 145, pp. 761–772, 2013.

[23] E. Gurevsky, O. Battaïa, and A. Dolgui, "Balancing of simple assembly lines under variations of task processing times," *Annals of Operations Research*, vol. 201, no. 1, pp. 265–286, 2012.

[24] ——, "Stability measure for a generalized assembly line balancing problem," *Discrete Applied Mathematics*, vol. 161, pp. 377–394, 2013.

[25] A. Dolgui and S. Kovalev, "Scenario based robust line balancing: Computational complexity," *Discrete Applied Mathematics*, vol. 160, no. 13–14, pp. 1955–1963, 2012.

[26] Ö. Hazır and A. Dolgui, "Assembly line balancing under uncertainty: Robust optimization models and exact solution method," *Computers & Industrial Engineering*, vol. 65, pp. 261–267, 2013.

[27] J. Rada-Vilela, M. Chica, O. Cordón, and S. Damas, "A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem," *Applied Soft Computing*, vol. 13, no. 11, pp. 4370–4382, 2013.

[28] M. Chica, O. Cordón, S. Damas, and J. Bautista, "Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search," *Information Sciences*, vol. 180, pp. 3465–3487, 2010.

[29] M. Chica, O. Cordón, and S. Damas, "An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem," *Computers and Industrial Engineering*, vol. 61, no. 1, pp. 103–117, 2011.

[30] P. W. Poon and J. N. Carter, "Genetic algorithm crossover operators," *Computers & Operations Research*, vol. 22, no. 1, pp. 135–147, 1995.

[31] E. Zitzler, K. Deb, and L. Thiele, "Comparison of multiobjective evolutionary algorithms: Empirical results," *Evolutionary Computation*, vol. 8, no. 2, pp. 173–195, 2000.

[32] K. Deb, *Multi-objective Optimization Using Evolutionary Algorithms*. Wiley, 2001.

[33] C. A. Coello, G. B. Lamont, and D. A. Van Veldhuizen, *Evolutionary Algorithms for Solving Multi-objective Problems (2nd edition)*. Springer, 2007.

[34] E. Zitzler, L. Thiele, M. Laumanns, C. M. Fonseca, and V. G. da Fonseca, "Performance assessment of multiobjective optimizers: an analysis and review," *IEEE Transactions on Evolutionary Computation*, vol. 7, no. 2, pp. 117–132, 2003.

[35] M. Chica, O. Cordón, S. Damas, and J. Bautista, "Multiobjective memetic algorithms for time and space assembly line balancing," *Engineering Applications of Artificial Intelligence*, vol. 25, no. 2, pp. 254–273, 2012.