

Interactive preferences in multiobjective ant colony optimisation for assembly line balancing

Manuel Chica · Óscar Cordon ·
Sergio Damas · Joaquín Bautista

Published online: 10 September 2014
© Springer-Verlag Berlin Heidelberg 2014

Abstract In this contribution, we propose an interactive multicriteria optimisation framework for the time and space assembly line balancing problem. The framework allows decision maker interaction by means of reference points to obtain the most interesting non-dominated solutions. The principal components of the framework are the g -dominance preference scheme and a state-of-the-art memetic multiobjective ant colony optimisation approach. In addition, the framework includes a novel adaptive multi-colony mechanism to be able to handle the preferences in an interactive way. Results show how the multiobjective framework can interactively obtain the most useful solutions with higher convergence than previous a priori methods. The experimentation also makes use of original data of the Nissan Pathfinder engine and practical bounds to define industrially feasible solutions in a set of scenarios. By solving the problem in these scenarios, we show the search guidance advantages of

using an interactive multiobjective ant colony optimisation method.

Keywords Interactive preferences · Multiobjective ant colony optimisation · Assembly line balancing problem

1 Introduction

A set of workstations forms an assembly line arranged either in series or in parallel. Each workstation performs a specific set of tasks that require a specific amount of time to complete. Assembly line balancing (ALB) efficiently assigns these tasks to workstations, balance task times, and consider constraints, such as precedence relationship (Boysen et al. 2007; Battaia and Dolgui 2013). The simple assembly line balancing problem (SALBP) (Baybars 1986; Scholl and Becker 2006) belongs to the ALB family of problems and considers the assignment of a task to a station in such a way that all the precedence constraints are satisfied and no station workload time is greater than the line cycle time.

The time and space assembly line balancing problem (TSALBP) is a realistic SALBP extension that considers additional space constraints. It is based on the observation of the Nissan plant of Barcelona (Bautista and Pereira 2007). The TSALBP-m/A¹ is one of the TSALBP variants that minimises the number of stations and the area of these stations. The TSALBP-m/A has a multicriteria nature as many other real-world problems that favours the application of multiobjective metaheuristics (MOMHs) (Chica et al. 2012).

Normally, when dealing with these kinds of industrial multicriteria decision-making problems, it is necessary to estab-

Communicated by V. Loia.

This work has been supported by Ministerio de Economía y Competitividad under project SOCOVIF12 (TIN2012-38525-C02-01 and TIN2012-38525-C02-02), and under PROTHIUS-III: DPI2010-16759, both including EDRF funding.

M. Chica (✉) · Ó. Cordon · S. Damas
European Centre for Soft Computing, Mieres, Spain
e-mail: manuel.chica@softcomputing.es

S. Damas
e-mail: sergio.damas@softcomputing.es

Ó. Cordon
DECSAI and CITIC-UGR, University of Granada, Granada, Spain
e-mail: oscar.cordon@softcomputing.es

J. Bautista
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: joaquin.bautista@upc.edu

¹ Originally, this TSALBP variant is referred as TSALBP-1/3 (Bautista and Pereira 2007).

lish preferences that explicitly account for expert knowledge to guide the search. The work of [Chica et al. \(2011b\)](#) explored how the decision maker can bias the search process using a MOMH with a priori preference methods for the TSALBP-m/A. This work also showed that when using a priori methods, the decision maker needs a deep understanding of the problem beforehand. Some difficulties also arise if the expert does not know the limitations of the problem instance and set optimistic or pessimistic preferences.

Interactive methods are conceived to overcome these drawbacks by allowing the decision maker to provide wiser and more effective preferences, participate interactively in the search process, and learn about the problem characteristics on-line ([Branke et al. 2008](#)). These advantages are important in the case of real-world problems such as the TSALBP where the decision maker can even ignore the feasible space of solutions.

In this work, we propose the use of an interactive framework based on reference points for solving the TSALBP-m/A. Our proposal allows decision makers to guide and interact with the algorithm during the optimisation process. g -dominance ([Molina et al. 2009](#)) is the preference elicitation method of the framework which has shown a good behaviour in different real-world problems ([Rodríguez et al. 2012](#); [Greiner et al. 2011](#)). In conjunction with g -dominance, we use an existing memetic multiobjective ant colony optimisation (MOACO) approach for the TSALBP. Particularly, we analyse the performance of two MOACO algorithms, multiple ant colony system (MACS) ([Barán and Schaerer 2003](#)) and Unsort Bicriterion (Unsort) ([Iredi et al. 2001](#)), with the incorporation of local search methods.

However, the majority of the interactive methods for MOMHs are designed for evolutionary algorithms and not for MOACO. For instance, [Kamalian et al. \(2004\)](#) suggested a posteriori method followed by an interactive multiobjective evolutionary algorithm. Another example is the evolutionary algorithm of [Phelps and Koksalan \(2003\)](#) that allows the user to provide preference information about pairs of solutions during the run. In [Deb et al. \(2010\)](#), authors propose an interactive evolutionary scheme to use the function value for directing the search to the preferred solutions.

Hence, the combined application of g -dominance to our MOACO algorithms has also the singularity of being the first interactive framework based on MOACO in the literature. Another important novelty of the work is that our MOACO designs include a mechanism to adapt their multi-colony approach to the selection of a specific reference point during the interactive process. We present a novel method to adapt these ants' thresholds to the initial approximation of the Pareto front and to the reference point selected by the decision maker.

We carry out three main experiments to validate the global proposal. The experimentation employs a real industrial case

from the Nissan Pathfinder engine manufacturing process obtained from the assembly line of Nissan in Barcelona, Spain.

First, we analyse the performance differences between MACS and Unsort with local search (MACS-LS and Unsort-LS, respectively) when using g -dominance. The second experiment compares, in terms of convergence, one of the existing a priori methods for the TSALBP-m/A ([Chica et al. 2011b](#)) with respect to g -dominance. The latter is done keeping in mind that we cannot measure by indicators the advantages of the interactive methods (applicability and usability) with respect to an a priori scheme.

Finally, we define two experimental scenarios where the decision maker interacts with the interactive framework. The evolution of the algorithm convergence is shown by attainment surfaces ([Fonseca and Fleming 1996](#)). To facilitate the interaction during the optimisation process, we have also defined and computed new practical bounds for the problem based on the industrial reality when configuring assembly lines that consider spatial information.

The remainder of the paper is structured as follows. Section 2 explains the TSALBP-m/A formulation and the main components of the algorithms. Section 3 shows the advantages and components of the novel interactive multiobjective framework. In Sects. 4 and 5 we first describe the design of the experiments and then analyse the obtained results. Finally, Sect. 6 highlights some concluding remarks.

2 Background

2.1 Time and space assembly line balancing

A set J of n tasks divides the manufacturing of a production item. Each task j requires an operation time for its execution $t_j > 0$ 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 ($S_k \subseteq J$), called workload. Each task j can only be assigned to a single station k .

Each task j has a set of direct "preceding tasks" P_j which must be accomplished before starting it. An acyclic precedence graph represents these constraints where its vertices are the tasks. A directed arc (i, j) of the graph 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. Each station k also presents a station workload time $t(S_k)$ equals to the sum of the processing times of the tasks assigned to k . The optimisation problem focuses on grouping tasks in workstations by an efficient and coherent way.

The need of introducing space constraints in the latter ALB model leads us to a family of problems called

TSALBP (Bautista and Pereira 2007) (an extension of SALBP). TSALBP defines temporal and spatial attributes (t_j and a_j , respectively) for all the n tasks which are also constrained by a precedence graph. A task must be assigned to a single station such that: (i) we satisfy every precedence constraint, (ii) the station workload time ($t(S_k)$) is lesser than the cycle time (c), and (iii) the area required by any station ($a(S_k)$) is lesser than the available area A .

TSALBP presents eight variants depending on three optimisation criteria. In this work, we tackle one multiobjective variant, the TSALBP-m/A, that consists of minimising the number of stations m and the station area A , given a fixed value of the cycle time c . More information about the problem can be found in Chica et al. (2010a). Mathematically, the TSALBP-m/A has the following two objectives:

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

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

where UB_m is the upper bound for the number of stations m (see Sect. 4.1 for more details about the bounds of the model), a_j is the area information for task j , x_{jk} is a decision variable taking value 1 if task j is assigned to station k , and n is the number of tasks.

2.2 Memetic MOACO algorithms for the TSALBP

A MOACO algorithm was the first successful proposal to tackle the TSALBP-m/A (Chica et al. 2010a). Then, some posterior studies presented the use of other MOMHs (Chica et al. 2010b, 2011a) and also showed that memetic algorithms are the best approaches to solve the problem (Chica et al. 2012). In Sect. 2.2.1 we first summarise two MOACO designs for the TSALBP-m/A, MACS and Unsort Bicriterion (Rada-Vilela et al. 2013). Then, in Sect. 2.2.2, we explain their memetic variants and the two local search operators (Chica et al. 2012) that lead to MACS-LS and Unsort-LS, respectively.

2.2.1 MACS and Unsort Bicriterion

MACS (Barán and Schaerer 2003) is an extension of the ant colony system (Dorigo and Gambardella 1997) to deal with multiobjective problems. The algorithm uses a pheromone trail matrix and several heuristic information functions. Unsort Bicriterion (Iredi et al. 2001) is a bicriterion optimisation algorithm with multiple colonies where the number of colonies is defined by the user and is independent from the number of objectives. The pheromone updates of Unsort are done by origin and performed by all ants.

Since the number of stations of the solutions is not fixed in advance, both MOACO algorithms follow a constructive and station-oriented scheme to face the precedence problem (this is a typical approach to solve the SALBP Scholl and Becker 2006). The process first opens a station and sequentially selects tasks to fill the station using a transition rule. The algorithm repeats this process for the current station until a stopping criterion is reached. Then, the algorithm opens a new station to be filled and iterates the loop until it assigns all the tasks.

Regarding the heuristic information, the work of Rada-Vilela et al. (2013) clearly showed that, to solve the TSALBP-m/A, MOACO algorithms yield better performance when they are only guided by pheromone trail information. Therefore, we will not consider the heuristic information. The pheromone information will have to memorise tasks that are the most appropriate to be assigned to a specific station. Hence, a pheromone trail associates a pair station-task (k, j).

The specific design of the algorithms for solving the TSALBP-m/A introduces an additional mechanism in the construction procedure. This mechanism randomly decides when to close the current station by considering a station closing probability distribution and a filling threshold $\alpha_i \in [0, 1]$ for each ant i . The probability distribution is defined by a station filling rate which is the overall processing time of its assigned set of tasks.

At each construction step, the MOACO algorithm computes the current station filling rate. If it is lower than the ant's filling threshold α_i (i.e. when it is lower than $\alpha_i \cdot c$), the station remains opened. Otherwise, the algorithm updates the station closing probability distribution and uniformly generates a random number in $[0, 1]$ to decide whether the station is closed or not. If the decision is to close the station, the algorithm creates a new station to allocate the remaining tasks.

Once decided whether opening or closing a station, the algorithm has to choose a task for current stations among all the candidate tasks using the MOACO transition rule. The procedure continues until it cannot assign any task. Thus, the higher the ant's threshold, the higher is the probability of filling fully a station, and vice versa. Different ants' filling thresholds make ants in the colony have a different search behaviour and they lead a better intensification-diversification trade-off. They also contribute to a highly diverse search behaviour of the ant population.

2.2.2 Multicriteria local search structure and components

The integration of global and local search operators in the memetic design of MACS-LS and Unsort-LS is done using scalarisations of the objective function vector (Gandibleux and Freville 2000; Jaskiewicz 2002). The memetic algorithm uses a weighted sum scalarisation of the two objectives of our problem, A and m , calculated by $\min(\lambda^1 A + \lambda^2 m)$.

The local search of the memetic MOACO algorithms has to optimise this multicriteria function. The local search generates the weight vector $\lambda = (\lambda^1, \lambda^2)$ at random for each constructed solution as usually done when designing memetic algorithms (Jaszkiewicz 2002).

We consider two different neighbour generation operators based on tasks moves. They are selected depending on the weight vector λ . If $\lambda^1 > \lambda^2$, the local search will apply the operator for minimising the A objective because the local search is biased towards the improvement of this objective. Otherwise, the algorithm will first consider the neighbour operator focused on improving m . When the initial neighbour operator does not succeed minimising the weighted sum scalarisation, the algorithm will apply the other operator.

According to the study performed in Chica et al. (2012), the best configuration of the memetic MOACO algorithms for the TSALBP-m/A is to apply the local search to all the generated solutions (as usual in MOACO with local search) and generate an intermediate intensification–diversification trade-off (some iterations are enough to lead to a proper convergence). We will use this configuration for the memetic algorithms considered in the current contribution.

3 The interactive framework proposal

3.1 The need for modelling interactive preferences

The way to incorporate the decision maker information into the multiobjective search process is of great interest in the MOMH research field (Coello et al. 2007). One of the most important questions is when to inject preference information in the multiobjective optimisation process. There are three approaches: (a) prior to the search (a priori approaches), (b) during the search (interactive approaches), and (c) after the search (a posteriori approaches).

The main contribution to the use of preferences for solving the TSALBP is presented in Chica et al. (2011b). In that work, authors proposed two different a priori approaches and included them into the MACS algorithm: goals and units of importance. Both elicitation schemes can provide solutions of the Pareto front approximation close to her/his preferences. For instance, the use of goals as a preferences scheme modifies the goal programming problem to incorporate them into the objective function $f_i(x)$ using different goal (t_j) relations: $f_i(x) \leq t_j$, $f_i(x) \geq t_j$, $f_i(x) = t_j$, or $f_i(x) \in [t_j^l, t_j^u]$ (Deb 1999).

However, these a priori methods have shown different drawbacks. First, the decision maker has difficulties to a priori express her/his preferences in real-world problems and cannot control the search process interactively when the results do not satisfy her/his aspirations. Also for the TSALBP, the expert must operate on labour or industrial

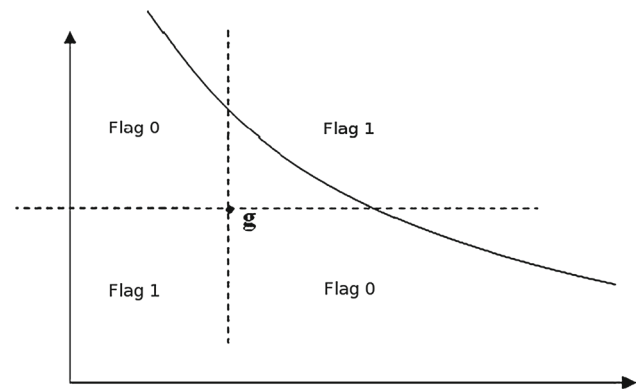


Fig. 1 Illustrative graph to show how to calculate g -dominance flag values from a reference point (Molina et al. 2009)

space costs and not directly with the objective values, this makes the preference elicitation even more complex.

In contrast, interactive methods allow the user to adjust preferences directly during the search and continue the optimisation process. They are simpler and require less expert effort than a priori methods, need moderate computational requirements, and the decision maker can effectively control the search process (Branke et al. 2008). As already stated, one of the most extended methods to introduce preferences interactively in MOMHs is using reference points (Molina et al. 2009; Ben Said et al. 2010; Figueira et al. 2010). The advantage of this approach is that it can return an approximation to the Pareto set and not simply projected solutions.

In one of the latter works, Molina et al. (2009) defined a new dominance relation, g -dominance, that can be easily incorporated into different MOMHs. Using g -dominance the search process works without any scalarisation function and the decision maker can interactively set her/his preferences by specifying a reference point. That will be the scheme used in our work.

3.2 The selected interactive method: g -dominance

g -dominance modifies the multiobjective Pareto dominance definition. Being p the number of objectives, given a reference point $v = (v_1, \dots, v_p) \in \mathfrak{R}^p$ and a generic point $w = (w_1, \dots, w_p) \in \mathfrak{R}^p$, the g -dominance relation $\text{Flag}_v(w)$ is defined as:

$$\text{Flag}_v(w) = \begin{cases} 1, & \text{if } w_i \leq v_i, \forall i = 1, \dots, p, \\ 1, & \text{if } v_i \leq w_i, \forall i = 1, \dots, p, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

This means that, given a reference point g , the dominance relation divides the space into four sub-spaces with different flag values (see Fig. 1). Based on the reference point and these flags, the conventional dominance relation changes. Given reference point g , two points $w = (w_1, \dots, w_p)$, $w' =$

$(w'_1, \dots, w'_p) \in \mathfrak{R}^p$ and assuming a minimisation problem, w' is g -dominated by w if either of the following Equations holds:

$$\text{Flag}_g(w) > \text{Flag}_g(w'), \quad (4)$$

$$\text{Flag}_g(w) = \text{Flag}_g(w'), \text{ satisfying that} \\ w_i \leq w'_i \quad \forall i = 1, \dots, p \\ \wedge \exists j : w_j < w'_j. \quad (5)$$

Thanks to the transformation of the traditional dominance scheme, g -dominance can drive the search to the desired area of the Pareto front ignoring the feasibility of the reference point. In a minimisation problem, a MOACO algorithm uses g -dominance as shown in Algorithm 1. M has to be a big enough number to make the solution dominated.

Algorithm 1: The pseudo-code of the evaluation method of a MOACO when using g -dominance.

```

1 Evaluate  $f_i(x)$ ,  $\forall i = 1, \dots, p$ 
2 Compute  $\text{Flag}_g(f)$ 
3 if  $\text{Flag}_g(f) = 0$  then
4    $f_i(x) = f_i(x) + M \quad \forall i = 1, \dots, p$ 
5 end

```

3.3 Adaptive multi-colony thresholds for using interactive preferences in a MOACO

One of the advantages of using g -dominance is the absence of changes in the algorithm design. However, driving the search towards a certain region of the search space may cause conflicts with one of the most important components of MOACO for TSALBP: the multi-colony approach based on filling thresholds.

As explained in Sect. 2.2.1, the station filling threshold of each ant (α_i) determines the probability of closing a station in the constructive station-oriented scheme. This threshold modifies the search behaviour of the ants of the multi-colony algorithm to achieve a correct intensification–diversification trade-off. The interactive preferences guide the algorithm to a certain region, but not all the ants are biased to explore that region in the original MOACO design. We need a new mechanism to adapt the ants' thresholds to the preferences of the decision maker regardless the approach followed to elicit these preferences.

The proposed adaptive method is to divide the search space into a user-dependent parameter ϕ of sub-spaces to focus the ants' search in one of them depending on the preference location (reference point, goals, etc.). Then, each sub-space will have a specific colony to explore this region.

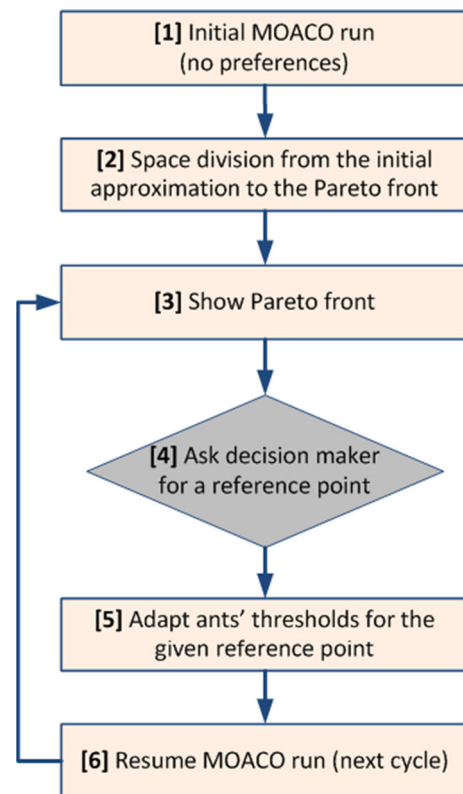


Fig. 2 Flowchart of the novel adaptive mechanism for MOACO

The run of the algorithm is based on cycles. A cycle stands for a certain number of evaluations of the MOACO to show the non-dominated solutions found until this moment allowing the decision maker to interact and resume the algorithm. Therefore, the interaction cycle length, specified by the user, sets the interaction frequency; i.e. it determines how often an user action is required.

Mainly, the adaptive mechanism is composed of the following six components (see Fig. 2 for a flowchart of the proposal):

1. *Initial MOACO run:* The memetic MOACO algorithm runs for the first cycle length without using any preferences.
2. *Space division:* The space division method calculates ϕ sub-spaces by taking into account the non-dominated solutions of the initial run. More details of the method are given at the end of this section.
3. *Show Pareto front:* The current approximation to the Pareto-optimal set given by the algorithm is shown. Information about the practical bounds of the problem is also presented (see Sect. 4.1).
4. *Ask decision maker for a reference point:* While the search process is stopped, the framework lets the decision maker to interactively indicate her/his preferences.

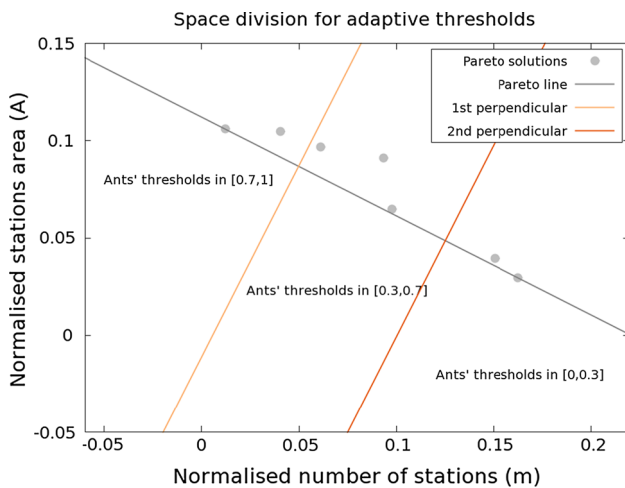


Fig. 3 Search space division using $\phi = 3$ and $\delta = 0.5$. Each of the three regions has different ants' filling thresholds to make the MOACO algorithm adaptive

For the case of g -dominance, (s)he just has to choose a reference point in the search space.

5. *Adapt ants' thresholds*: The reference point is located in one of the ϕ sub-spaces. Consequently, the colony will be modified with the ants' thresholds assigned to this specific sub-space.
6. *Resume MOACO run*: Once the MOACO algorithm is adapted, it continues for the next cycle. The algorithm will be again stopped every time an interaction is required. Then, the new approximation to the Pareto front is shown, the decision maker will select a new reference point and the process will continue.

We should remark that the algorithm only calculates the space division once, before the first interaction. A 2D search space is considered for our bi-objective TSALBP-m/A although the mechanism can be extended to higher dimensional problems.

The space division of the method (step 2) works as follows. First, the algorithm calculates a straight line between the most extreme non-dominated solutions of the current Pareto set approximation. Then, we calculate, with the support of the latter line, $\phi - 1$ perpendicular lines to the initial approximation to the Pareto front to divide the space in ϕ sub-spaces. After a preliminary study, we considered $\phi = 3$ sub-spaces.

A parameter (δ) modifies the width of the sub-spaces by adjusting the width of the central sub-space with respect to the total width of the space enclosed by the extreme solutions. The width of the other sub-spaces is proportionally obtained from the remaining $(1 - \delta)/2$. Figure 3 shows an example of space division with $\phi = 3$ and $\delta = 0.5$.

The algorithm will assign a colony (i.e. ants having similar filling thresholds) to each sub-space. Depending on these thresholds, the colony will intensify or diversify the search

by closing the stations earlier or later. Although there are different methods to distribute the ants' thresholds within the colony we have chosen a fixed uniform distribution of the filling thresholds within a range. Other options to be explored are: (a) generating thresholds' at random or (b) overlapping the threshold ranges of the sub-spaces.

4 Experimental design

4.1 Practical bounds to ease the interactive process

Some solutions found by the algorithms are theoretically feasible, but industrially infeasible. The definition and visualisation of practical bounds for the TSALBP-m/A can save decision-making efforts by focusing the search and facilitating the interactive process. In this section, we will mathematically define them for the interactive experiments. Following the notations of Sect. 2.1, we obtain the theoretical upper and lower bounds for objectives m and A by:

$$\begin{aligned} LB_m &= 1, & UB_m &= n, \\ LB_A &= \max_{j=1, \dots, n} a_j, & UB_A &= \sum_{j=1}^n a_j. \end{aligned} \quad (6)$$

We note the upper and lower practical bounds for the objectives of TSALBP-m/A by $[m_{\min}^*, m_{\max}^*]$ and $[A_{\min}^*, A_{\max}^*]$. Practical bounds for the TSALBP-m/A come from the definition of "acceptable area" for the stations of the assembly line in an industrial context. Basically, this "acceptable area" of an assembly line depends on (a) workers' movements through the continuous transportation system (limit the area to an upper bound A_{\max}^*); and (b) ergonomic factors that make stations comfortable for workers (limit stations' area to a lower bound A_{\min}^*).

To determine the practical minimum number of stations m_{\min}^* for a TSALBP-m/A instance, we computationally solve the TSALBP-m/A when $A \rightarrow \infty$ (equivalent to solve the SALBP-1 Scholl and Klein 1999). The solution of this problem will output an assembly line configuration having m_{\min}^* stations: $\vec{S}^* = (S^*(1), \dots, S^*(m_{\min}^*))$. Using this configuration \vec{S}^* , we obtain A_{\max}^* by calculating the maximum area of its workstations: $A_{\max}^* = \max_{k=1, \dots, m_{\min}^*} (\sum_{j \in S^*(k)} a_j)$.

The other two practical bounds, m_{\max}^* and A_{\min}^* , will have the same values as their theoretical counterparts, UB_m and LB_A , because in some cases it is industrially feasible to have one task per station being carried out by the associated worker or work team.

4.2 General set-up and problem instances

We run all the algorithms 20 times. They were launched using the same framework (programmed in C++) and the same computer (Intel Core™ i5-2400 with four CPUs at 3.10 GHz

Table 1 Configuration of the memetic MOACO algorithms

Parameter	Value
General parameters	
Runs (different seeds)	20
Stopping criteria	200,000 evaluations
Interaction cycle length	20,000 evaluations
β	2
ρ	0.2
q_0	0.2
MACS	
Number of ants	10
Unsort Bicriterion	
Number of colonies	10
Ants per colony	10
Adaptive multi-colony ants' thresholds	
Number of regions (ϕ)	3
Width of central sub-space (δ)	0.75
Thresholds α_i for ants (two ants per threshold value)	
★ When no preference point is defined	{0.2, 0.4, 0.6, 0.7, 0.9}
★ When point is in sub-space 1	{0, 0.08, 0.15, 0.22, 0.3}
★ When point is in sub-space 2	{0.3, 0.4, 0.5, 0.6, 0.7}
★ When point is in sub-space 3	{0.7, 0, 0.78, 0.85, 0.92, 1}
LS configuration	
Application criteria	Always
No. of iterations	50

and Scientific Linux 6.2 as operating system). The number of evaluations is set as stopping criterion. All the parameter values of the algorithms considered in the experimentation are shown in Table 1.

We have selected three real-like problem instances with different features for the experimentation (Chica et al. 2012): *barthol2* and $c = 85$, *lutz2* and $c = 16$, *scholl* and $c = 1394$. We have also considered a fourth real-world problem instance corresponding to the assembly process of the Nissan Pathfinder engine, developed at the Nissan industrial plant in Barcelona, Spain. The assembly of these engines is divided into 378 operation tasks (grouped into 140). For more details about the instances the interested reader is referred to Chica et al. (2012).

4.3 Performance indicators

An experimentation with interactive methods needs interaction during the run. To allow this behaviour, we will define a “cycle” as a run of the algorithm for a certain number of evaluations (fixed by the user). When a cycle ends, the method shows the non-dominated solutions found by the algorithm so far to allow the decision maker to define a reference point

in the search space of the problem and resume the run of the algorithm.

Experimentation uses the multiplicative I_ϵ indicator (Zitzler et al. 2003) to compare the behaviour of the algorithms. To facilitate the analysis of the I_ϵ multiobjective performance indicator we have represented its values in box-plots. Each plot contains pairs of coloured and non-coloured box-plots that show the distribution of the I_ϵ values for each pair of algorithms (I_ϵ indicator is not symmetric). Given Fig. 4 as an example, the first plot shows the box-plots comparing MACS-LS and Unsort-LS for *barthol2* instance using two reference points. As I_ϵ is a binary indicator, two box-plots (non-coloured and coloured) are drawn for each comparison. The non-coloured box-plots represent the I_ϵ (MACS, UNSORT) distribution in 20 runs, while coloured ones represent the I_ϵ (UNSORT, MACS).

Together with numerical performance indicators we use attainment surfaces in the interactive scenarios to provide visual and qualitative information, sometimes more useful than numerical values (Fonseca and Fleming 1996).

5 Analysis of the results

In Sect. 5.1, the MOACO algorithms are compared to see if there is any significant convergence difference when using g -dominance. Then, in Sect. 5.2, we will observe the convergence performance of a MOACO algorithm under two preference elicitation schemes: using g -dominance and a priori schemes. Finally, the proposed interactive framework is evaluated in two interactive scenarios (Sect. 5.3).

5.1 A comparison between MACS-LS and Unsort-LS when using g -dominance

In Fig. 4, we show the I_ϵ box-plots of the comparison between MACS-LS and Unsort-LS to analyse performance differences (if any) when using the g -dominance method. Each plot corresponds to one of the four used instances. In addition, we have defined two different kinds of reference points for each problem instance which are presented in Table 2. Half of these points falls outside the industrially feasible area determined by the practical bounds of Sect. 4.1. The other half is located inside the feasibility area.

The results of the I_ϵ indicator show that there is no big difference between both algorithms. The 20 runs of the algorithms achieved almost the same Pareto fronts for *barthol2* and *scholl* instances for the first reference point of *lutz2*. For the rest of the instances, the difference is found in the statistical distribution of the runs as the upper and lower ends of the box-plots are practically in similar places.

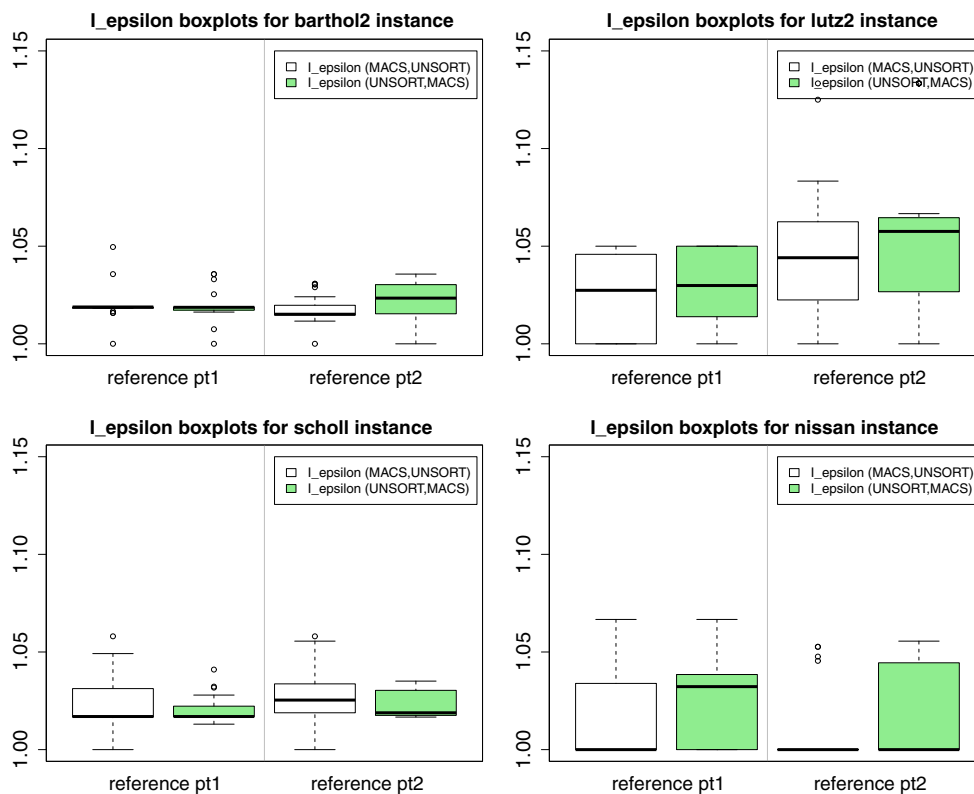


Fig. 4 I_ϵ box-plots comparing the performance of MACS-LS and Unsort-LS when using g -dominance for the four problem instances

Table 2 List of reference points used for the experimentation

Problem instance	Ref. point 1 (industrially feasible)	Ref. point 2 (industrially infeasible)
barthol2	(52, 115)	(65, 80)
lutz2	(33, 20)	(35, 8)
scholl	(59, 1386)	(50, 1200)
nissan	(24, 3)	(16, 5)

Although Unsort Bicriterion was found to achieve better results than MACS in previous works (Rada-Vilela et al. 2013), this experimentation has shown no significant difference between them. The integration of the algorithms in a memetic scheme (MACS-LS and Unsort-LS) and the use of a common interactive framework (g -dominance method and the adaptive thresholds mechanism) alleviate the minor differences previously found. Therefore, to reduce the excess of comparisons during the next experiments, we will use MACS-LS from now on.

5.2 Convergence analysis of the g -dominance-based framework against a priori goals

In this section, we will compare how the proposed interactive framework based on g -dominance behaves, in terms of

convergence, with respect to the existing a priori preferences scheme (based on goals). We have used the instances and reference points shown in the previous section (Table 2).

During this experiment, no user interaction is included during the run of the *interactive* method as both approaches in comparison have a fixed reference point (a goal for the a priori approach). In our case and for a fair comparison with g -dominance, we will use the goal relation $f_i(x) = t_j$ for each defined reference point. We will check if the convergence obtained by the interactive framework is, at least, competitive with respect to the a priori approach.

We show the I_ϵ values comparing both approaches in the box-plots of Figs. 5, 6, 7, 8. The performance indicator is checked every cycle (20,000 evaluations). Note that for the interactive approach we apply g -dominance from the start of the run of the algorithm as there is no user interaction and no need to show a first approximation to the Pareto front.

The different cycles in the box-plots show how the Pareto front approximations of the interactive framework (MACS-LS and g -dominance) evolve through the run of both approaches. By analysing these box-plots, we observe the following facts:

- Usually, the g -dominance method obtains better results although the solutions generated by the g -dominance and a priori schemes are similar for all the problem instances.

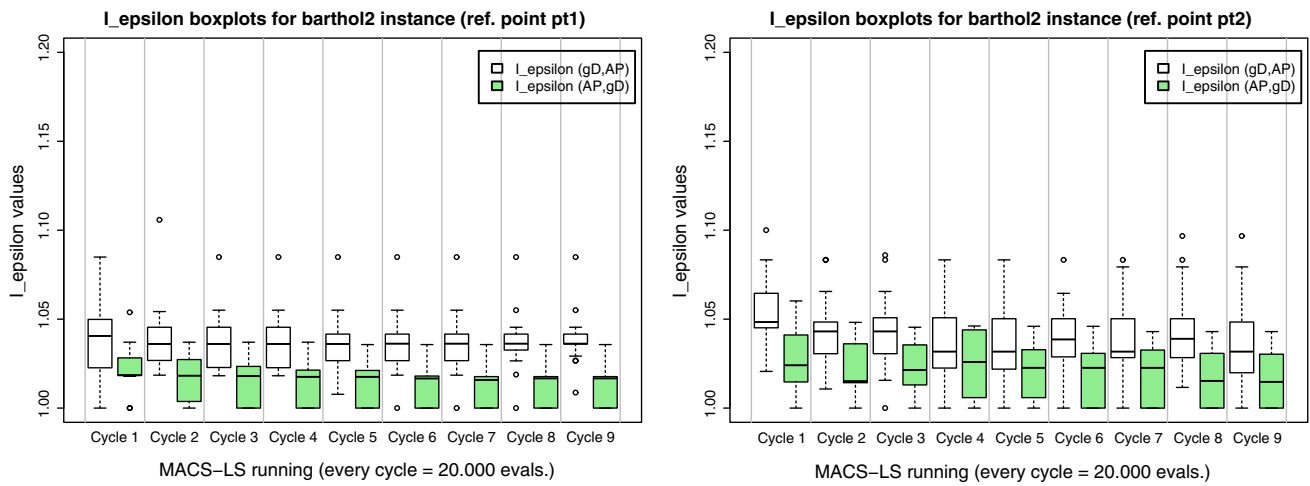


Fig. 5 I_ϵ box-plots comparing the performance of an a priori and g -dominance approaches with two different reference points for the *barthol2* instance

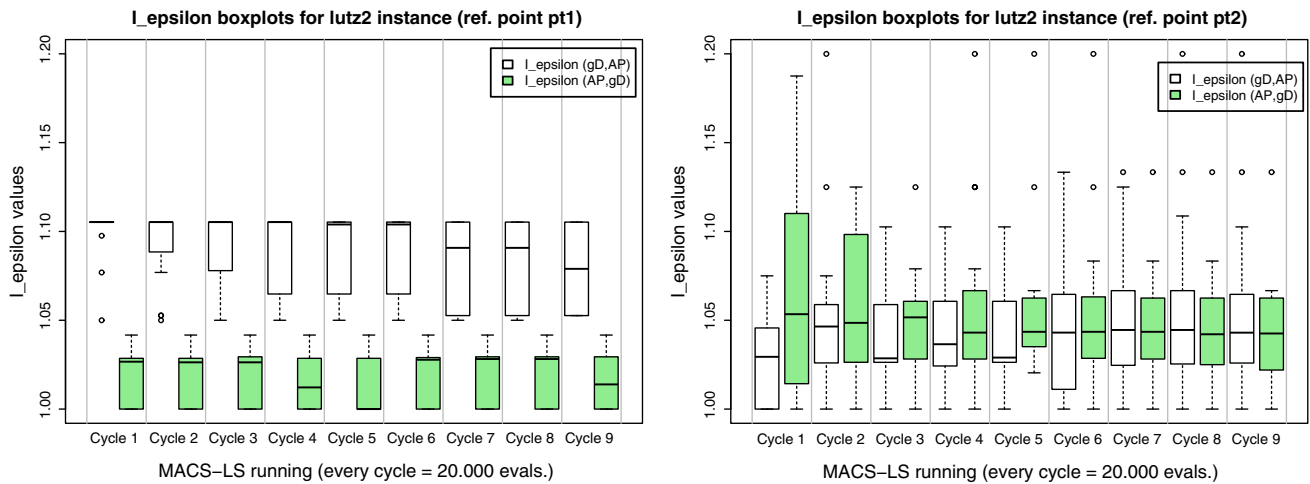


Fig. 6 I_ϵ box-plots comparing the performance of an a priori and g -dominance approaches with two different reference points for the *lutz2* instance

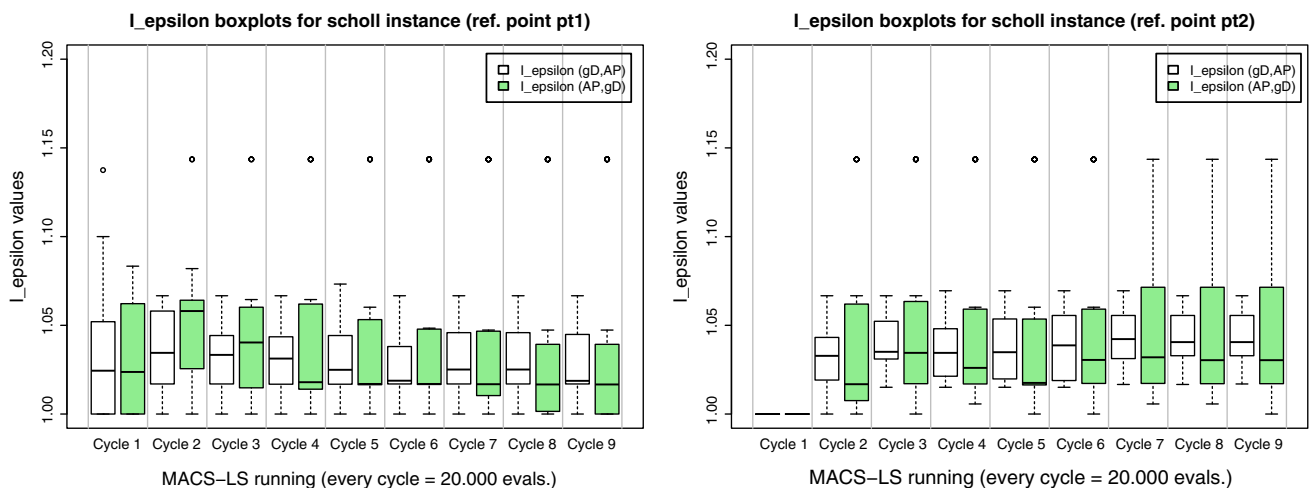


Fig. 7 I_ϵ box-plots comparing the performance of an a priori and g -dominance approaches with two different reference points for the *scholl* instance

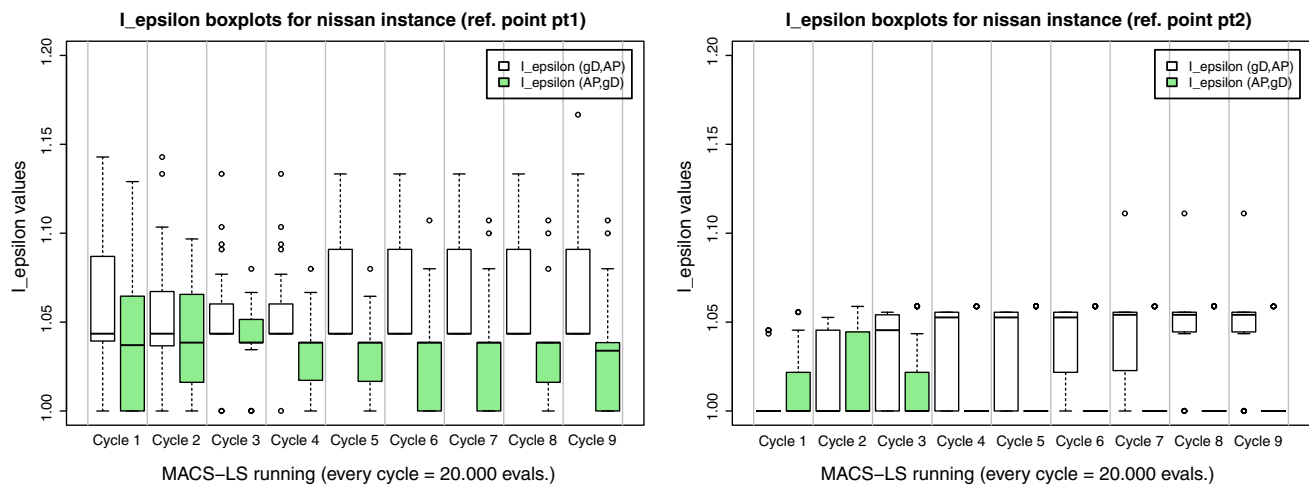


Fig. 8 I_ϵ box-plots comparing the performance of an a priori and g -dominance approaches with two different reference points for the nissan case study

If we evaluate the I_ϵ values in both directions (pair of coloured and non-coloured box-plots), we see that g -dominance obtains the best results for *barthol2* and *nissan* considering both reference points (Figs. 5, 8), and *lutz2* when selecting reference point 1 (left plot of Fig. 6). g -dominance also outperforms the a priori scheme in *scholl* (Fig. 7), but differences are lower than in other instances.

- The interactive framework has the worst behaviour in *lutz2* when using reference point 2 (right plot of Fig. 6). During the first five cycles, the convergence of the a priori method is better although after the fifth cycle results are practically the same. Therefore, the g -dominance method presents a slower convergence speed than the a priori method in this specific scenario.
- The latter behaviour also appears in *scholl* when using reference point 1 (see left plot of Fig. 7). Initially, the a priori method obtains better convergence in the first three cycles, but the interactive method finally yields better Pareto front approximations from the fourth cycle until the end of the run.

As a global analysis we can say that the convergence of the method is better than when using a priori goals in almost all the cases of the experimentation. In some cases, we have noticed that the convergence speed of g -dominance is slightly slower than goals. Despite these facts, the real advantage of using g -dominance does not just reside in its convergence, but mainly in the interaction and better control of the region of interest.

5.3 Using the framework in interactive scenarios

In this section, we have defined two interactive scenarios for two instances: *barthol2* and *nissan*. We have not

defined fixed reference points as in the previous section since the decision maker interactively selects the reference points according to her/his interest and the information (s)he receives from the algorithm in an on-line fashion. Again, we divide the run of the algorithm in cycles. The first cycle is run without preferences to show an initial approximation to the Pareto front and the representation of the practical bounds for the problem instance. Then, the decision maker interacts with the interactive framework at every cycle.

Figures 9 and 10 show the attainment surfaces of the interaction evolution for *barthol2* and *nissan*, respectively. To show statistically significant results we have plotted the non-dominated solutions obtained by all the 20 runs at every interaction cycle.

For the first scenario of *barthol2*, the algorithm is set to a length cycle of 20,000 evaluations. The decision maker first chooses (56, 90) as reference point. This first reference point has a balance between m and A (it is in the central part of the Pareto front; see the first plot of Fig. 9). We remind that the position of the reference point conditions the calculation of ants' thresholds in the multi-colony approach.² In this case, those thresholds are $\alpha_i \in [0.3, 0.7]$.

After the first cycle, the decision maker decides to explore solutions with a lower number of stations. Therefore, (s)he chooses (53, 115) as the second reference point (second plot of Fig. 9). The MACS-LS algorithm has to adapt the thresholds of the colony again because the reference point influences the search direction towards solutions having a lower number of stations. Now, ants' thresholds become $\alpha_i \in [0.7, 1.0]$.

² Please note that the space division and, therefore, the ants' thresholds depend on the first Pareto front approximation and this front can vary among the 20 runs. For simplicity, we refer here to the most common thresholds of the runs.

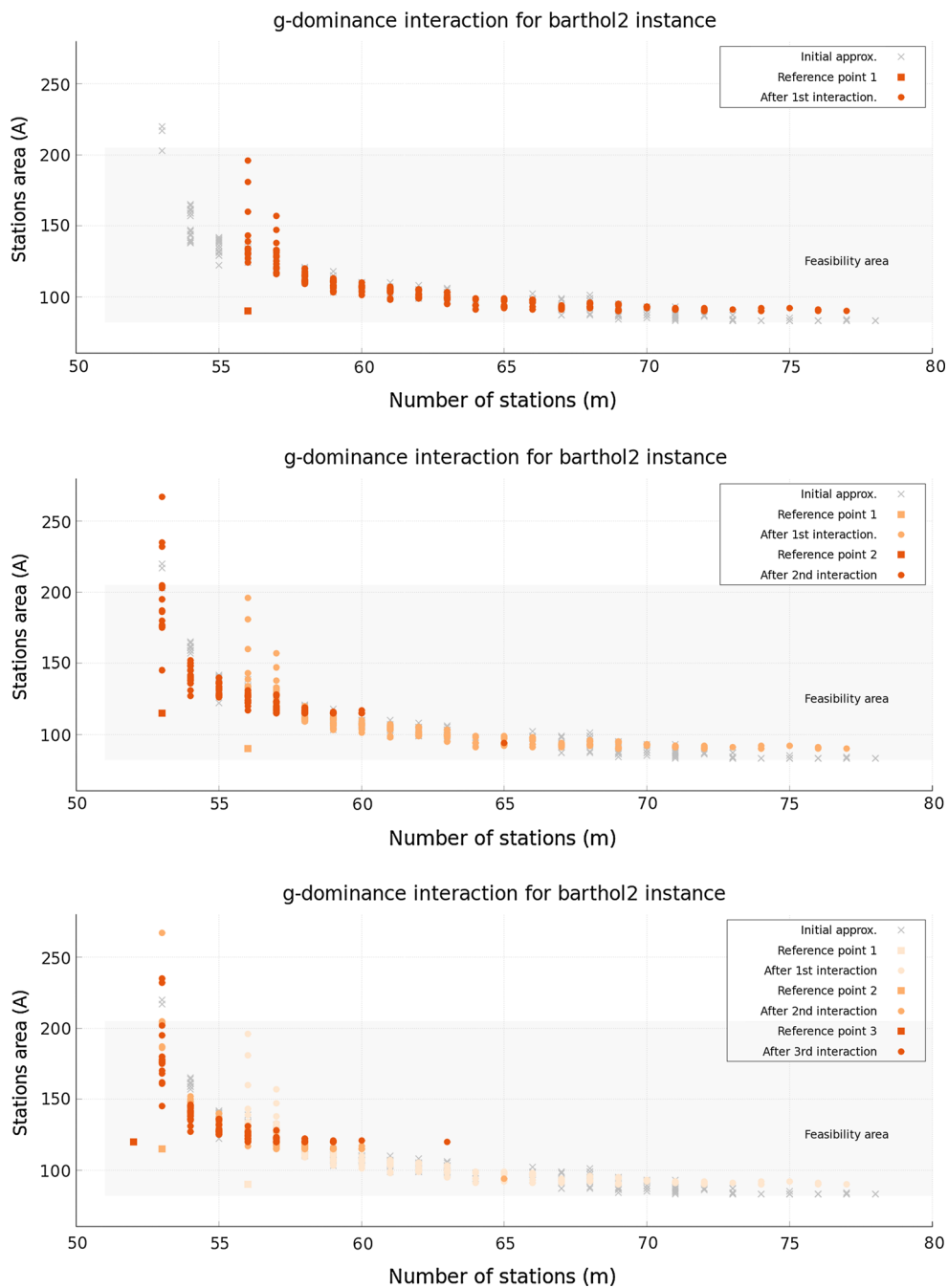


Fig. 9 Attainment surfaces of MACS-LS using *g*-dominance in an interactive scenario for *barthol2*. The three figures show the selection of the three reference points and their resulting Pareto fronts

This search exploitation originates new solutions with $m = 53$ in the left-most hand part of the Pareto front approximation that were not found beforehand. The last reference point $(52, 120)$ determines the next search direction of the algorithm and finds solutions with $m = 53$ and lower station area A (see the third plot of Fig. 9). When the algorithm has finished, the obtained Pareto set approximation has almost satisfied the expert preferences by returning solutions with 53, 54 or 55 stations with less than 150 m.

For the second scenario, we defined two reference points for the *nissan* instance through cycles of 5,000 evaluations. Both reference points $((27, 3)$ and $(24, 2))$ follow the same search direction. That means the decision maker does not change the balance of objectives during the run of the algorithm as it happened in the previous *barthol2* scenario. Although the last reference point $(24, 2)$ is not a feasible solution the performance of the algorithm is not affected. The convergence of the returned Pareto set approx-

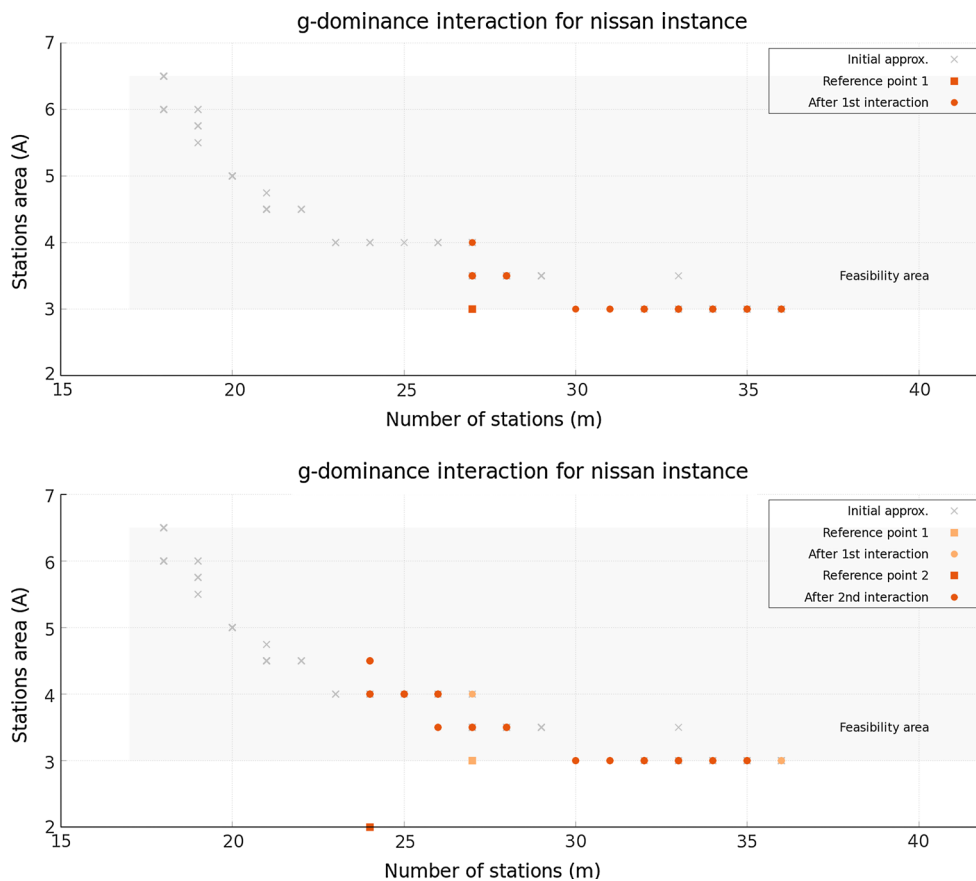


Fig. 10 Attainment surfaces of MACS-LS using g -dominance in a interactive scenario for `nissan`. Two reference points were chosen and their resulting Pareto fronts are shown

imations increases through the cycles even when the reference point is infeasible (see both plots of Fig. 10). Notice the help provided by showing the industrial feasibility area as the expert can increase the search intensification in a specific search space region by knowing the practical bounds beforehand.

6 Concluding remarks and future works

We have combined a memetic MOACO approach, using the MACS-LS and Unsort-LS algorithms, with an interactive preference elicitation method (g -dominance) for solving the TSALBP-m/A model. The combination formed a multiobjective interactive framework that allows the decision maker to interactively find a set of industrial solutions close to her/his point of preference. Up to our knowledge, the integration of an interactive preferences scheme into a MOACO algorithm is the first of its kind.

First, we compared the performance of two different MOACO algorithms when using g -dominance. Results showed hardly any difference. Then, we presented a comparison between an existing a priori preference elicitation

method and our proposed framework based on g -dominance. g -dominance obtained better convergence to the preference area for almost all the instances, an additional advantage to its better process control properties.

The definition of different interactive scenarios for solving the TSALBP-m/A allowed us to validate the framework interaction in the third part of the experimentation. One of these scenarios defined the real-world case of the Nissan Pathfinder engine. Interactive experiments showed that the framework correctly manages to converge towards the reference point. The interactive method in addition behaved well when the search direction of the expert preferences changed during the algorithm run.

Despite the good g -dominance performance we noticed the need of a user parameter to control the spread of the region of interest where preferred solutions might be found. This fact suggests us a future work focused on extending the capabilities of the current g -dominance definition. Additional future research lines can be to extend the searching for the most robust solutions with interactive preferences and to create more sophisticated graphical models to enrich visualisation and interaction (see the work proposed in Chica et al. 2013).

References

- Barán B, Schaerer M (2003) A multiobjective ant colony system for vehicle routing problem with time windows. In: 21st IASTED international conference, Innsbruck, pp 97–102
- Battaia O, Dolgui A (2013) A taxonomy of line balancing problems and their solution approaches. *Int J Prod Econ* 142(2):259–277
- Bautista J, Pereira J (2007) Ant algorithms for a time and space constrained assembly line balancing problem. *Eur J Oper Res* 177(3):2016–2032
- Baybars I (1986) A survey of exact algorithms for the simple assembly line balancing problem. *Manag Sci* 32(8):909–932
- Said Ben L, Bechikh S, Ghédira K (2010) The r-dominance: a new dominance relation for interactive evolutionary multicriteria decision making. *IEEE Trans Evol Comput* 14(5):801–818
- Boysen N, Fliedner M, Scholl A (2007) A classification of assembly line balancing problems. *Eur J Oper Res* 183(2):674–693
- Branke J, Deb K, Miettinen K, Slowinski R (eds) (2008) Multiobjective optimization, interactive and evolutionary approaches. Lecture notes in computer science, vol 5252. Springer, Berlin
- Chica M, Cerdón O, Damas S (2011) An advanced multi-objective genetic algorithm design for the time and space assembly line balancing problem. *Comput Ind Eng* 61(1):103–117
- Chica M, Cerdón O, Damas S, Bautista J (2010) Multiobjective, constructive heuristics for the 1/3 variant of the time and space assembly line balancing problem: ACO and random greedy search. *Inf Sci* 180:3465–3487
- Chica M, Cerdón O, Damas S, Bautista J (2010) A multiobjective GRASP for the 1/3 variant of the time and space assembly line balancing problem. *Trends Appl Intell Syst Lect Notes Artif Intell* 6098:656–665
- Chica M, Cerdón O, Damas S, Bautista J (2011) Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different nissan scenarios. *Exp Syst Appl* 38:709–720
- Chica M, Cerdón O, Damas S, Bautista J (2012) Multiobjective memetic algorithms for time and space assembly line balancing. *Eng Appl Artif Intell* 25(2):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. *Int J Prod Econ* 145:761–772
- Coello CA, Lamont GB, Van Veldhuizen DA (2007) Evolutionary algorithms for solving multi-objective problems, 2nd edn. Springer, Berlin
- Deb K (1999) Solving goal programming problems using multi-objective genetic algorithms. In: IEEE congress on evolutionary computation (CEC 99). Washington (USA), pp 77–84
- Deb K, Sinha A, Korhonen PJ, Wallenius J (2010) An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. *IEEE Trans Evol Comput* 14(5):723–739
- Dorigo M, Gambardella L (1997) Ant colony system: a cooperative learning approach to the traveling salesman problem. *IEEE Trans Evol Comput* 1(1):53–66
- Figueira JR, Liefvooghe A, Talbi EG, Wierzbicki AP (2010) A parallel multiple reference point approach for multi-objective optimization. *Eur J Oper Res* 205(2):390–400
- Fonseca CM, Fleming PJ (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). Lecture notes in computer science, vol 1141. Berlin, Germany, pp 584–593
- Gandibleux X, Freville A (2000) Tabu search based procedure for solving the 0–1 multiobjective knapsack problem: the two objectives case. *J Heurist* 6(3):361–383
- Greiner D, Galván B, Emperador JM, Méndez M, Winter G (2011) Introducing reference point using g-dominance in optimum design considering uncertainties: an application in structural engineering. In: Evolutionary multi-criteria optimization. Lecture notes in computer science, vol 6576. Springer, Berlin, pp 389–403
- Iredi S, Merkle D, Middendorf M (2001) Bi-criterion optimization with multi colony ant algorithms. In: First intl. conference on evolutionary multi-criterion optimization (EMO'01). Springer, Berlin, pp 359–372
- Jaszkiewicz A (2002) Genetic local search for multiple objective combinatorial optimization. *Eur J Oper Res* 137(1):50–71
- Kamalian R, Takagi H, Agonino AM (2004) Optimized design of MEMS by evolutionary multi-objective optimization with interactive evolutionary computation. Springer, Berlin, pp 1030–1041
- Molina J, Santana LV, Hernández-Díaz AG, Coello CA, Caballero R (2009) g-dominance: reference point based dominance for multiobjective metaheuristics. *Eur J Oper Res* 197(2):17–24
- Phelps S, Koksalan M (2003) An interactive evolutionary meta-heuristic for multiobjective combinatorial optimization. *Manag Sci* 49(12):1726–1738
- Rada-Vilela J, Chica M, Cerdón O, Damas S (2013) A comparative study of multi-objective ant colony optimization algorithms for the time and space assembly line balancing problem. *Appl Soft Comput* 13(11):4370–4382
- Rodríguez B, Molina J, Pérez F, Caballero R (2012) Interactive design of personalised tourism routes. *Tour Manag* 33(4):926–940
- Scholl A, Becker C (2006) State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *Eur J Oper Res* 168(3):666–693
- Scholl A, Klein C (1999) Balancing assembly lines effectively—a computational comparison. *Eur J Oper Res* 114(1):50–58
- Zitzler E, Thiele L, Laumanns M, Fonseca CM, da Fonseca VG (2003) Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Trans Evol Comput* 7(2):117–132