Tackling the 1/3 Variant of the Time and Space Assembly Line Balancing Problem by Means of a Multiobjective Genetic Algorithm

Manuel Chica
European Centre for Soft Computing
Mieres (Asturias), Spain
manuel.chica@softcomputing.es

Óscar Cordón
European Centre for Soft Computing
Mieres (Asturias), Spain
oscar.cordon@softcomputing.es

Sergio Damas
European Centre for Soft Computing
Mieres (Asturias), Spain
sergio.damas@softcomputing.es

***Abstract*—The time and space assembly line balancing problem (TSALBP) considers realistic multiobjective versions of the classical assembly line balancing involving the joint optimization of conflicting criteria such as the cycle time, the number of stations, and/or the area of these stations. This industrial problem is very difficult to solve and of crucial importance in the manufacturing context. As TSALBP-1/3 contains a set of hard constraints like precedences or cycle time limits for each station it has been mainly tackled using multiobjective constructive metaheuristics (e.g. ant colony optimization). Global search algorithms in general –and multiobjective genetic algorithms in particular– have shown to be ineffective to solve this family of problems up to now. The goal of this contribution is to present a new multiobjective genetic algorithm design, taking the well known NSGA-II algorithm as a base and new coding scheme and specific operators, to properly tackle with the TSALBP. An experimental study on six different problem instances is used to compare the proposal with the state-of-the-art methods.**

I. Introduction

An assembly line is made up of a number of workstations, arranged either in series or in parallel. Since the manufacturing of a production item is divided into a set of tasks, a usual and difficult problem is to determine how these tasks can be assigned to the stations fulfilling certain restrictions. Consequently, the aim is to get an optimal assignment of subsets of tasks to the stations of the plant. Moreover, each task requires an operation time for its execution.

A family of academic problems –referred to as simple assembly line balancing problems (SALBP)– was proposed to model this situation [1] [2]. Taking this family as a base and adding spatial information to enrich it, Bautista and Pereira recently proposed a more realistic framework: the time and space assembly line balancing problem (TSALBP) [3]. This framework considers an additional space constraint to become a simplified version of real-world problems. The new space constraint emerged due to the study of the specific characteristics of the Nissan plant in Barcelona (Spain).

As many real-world problems, TSALBP formulations have a multicriteria nature [4] because they contain three conflicting objectives to be minimised: the cycle time of the assembly line, the number of the stations, and the area of these stations. In this paper we deal with the TSALBP-1/3 variant which tries to minimise the number of stations and their area for a given product cycle time. TSALBP-1/3 has an important set of hard constraints like precedences or cycle time limits for each station. Thus, the use of constructive approaches is more convenient than others like local or global search procedures [5]. In [6] authors successfully tackled the TSALBP-1/3 by means of a specific procedure based on the multiple ant colony system (MACS) algorithm [7]. This approach is the state-of-the-art of TSALBP-1/3 and it outperformed a multiobjective genetic algorithm. In particular, the latter method –a multiobjective extension of an existing genetic algorithm for SALBP [8] based on the use of the NSGA-II [9]– showed a very low performance when solving the TSALBP-1/3. Such weak performance of the latter multiobjective genetic algorithm was due to its inability to deal with the inherent problem characteristics and not to any drawback of the evolutionary multiobjective optimization (EMO) approach. In fact, EMO could be a powerful tool to accurately solve this very complex problem.

Therefore, in this contribution a new design of a multiobjective genetic algorithm is developed, also based on NSGA-II but incorporating specific components to appropriately deal with the TSALBP constraints. On the one hand, a new individual representation will be proposed which is more faithful to the solution phenotype and thus more appropriate for the problem solving. On the other hand, novel crossover, repair, and mutation operators will be designed to overcome the non constructive nature of genetic algorithms when dealing with the TSALBP constraints.

Our proposal will be compared with the existing multiobjective genetic algorithm and the state-of-the-art algorithm to solve the problem, MACS for the TSALBP-1/3. We will consider six well-known problem instances from the literature for this experimental study. In order to evaluate the performance of the different methods, a detailed analysis of results will be developed considering the usual multiobjective performance indicators (metrics).

This paper is structured as follows. In Section II the formulation of the TSALBP-1/3 is given. Then, in Section III, the MACS algorithm and the multiobjective extension of

978-1-4244-7835-4/11/$26.00 ©2011 IEEE

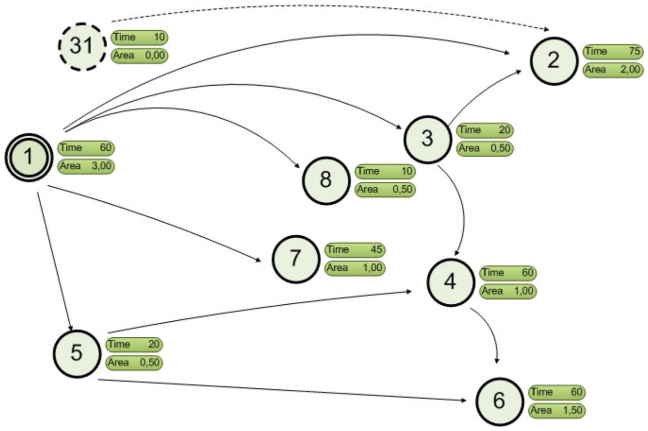

Fig. 1: A precedence graph representing the first 8 tasks of a problem instance. Time and area information are shown next to each task.

the genetic algorithm for SALBP are reviewed. Our novel multiobjective genetic algorithm design for the problem is described in Section IV. The used performance indicators, problem instances, and the analysis of the obtained results are presented in Section V. Finally, in Section VI, some concluding remarks are provided.

II. The Time and Space Assembly Line Balancing Problem

The manufacturing of a production item is divided into a set V 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 is assigned to a subset of tasks S_k ($S_k \subseteq V$), called workload. A task j is assigned to a station k.

Each task j has a set of direct predecessors, P_j, which must be accomplished before starting it. These constraints are normally represented by means of an acyclic precedence graph, whose vertices stand for the tasks and where a directed arc (i, j) indicates that task i must be finished before starting task j on the production line. Thus, if $i \in S_h$ and $j \in S_k$, then $h \leq k$ must be fulfilled. Each station k presents a station workload time $t(S_k)$ that is equal to the sum of the tasks' lengths assigned to the station k. SALBP [2] focuses on grouping tasks in workstations by an efficient and coherent way. There is a large variety of exact and heuristic problem-solving procedures for it [10].

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

This leads us to a new family of problems called TSALBP in [3]. It may be stated as: given 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).

TSALBP presents eight variants depending on three optimization criteria: m (the number of stations), c (the cycle time) and A (the area of the stations). Within these variants there are four multiobjective problems and we will tackle one of them, the TSALBP-1/3. It consists of minimising the number of stations m and the station area A, given a fixed value of the cycle time c, mathematically formulated as follows:

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

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

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

We chose this variant because it is realistic in the automotive industry since the annual production of an industrial plant (and therefore, the cycle time c) is usually set by some market objectives. For more information we refer the interested reader to [6].

III. Previous Approaches to Solve the TSALBP-1/3

The specialised literature includes a large variety of exact and heuristic problem-solving procedures as well as metaheuristics for solving the SALBP [10], [11]. Mainly, the use of genetic algorithms [8], [12], tabu search [13], simulating annealing [14], and ant colony optimization [3] have been considered. Besides, multicriteria formulations of the SALBP have also been tackled using genetic algorithms [15] and ant colony optimization [16].

However, there are not many proposals for solving the multiobjective 1/3 variant of the TSALBP [6]. A MACS algorithm and a multiobjective extension of the SALBP genetic algorithm (first introduced in [8]) can be found. These two algorithms will be briefly reviewed in the next subsections, as will be compared with our proposal in the experimental study developed in Section V.

A. The MACS algorithm

MACS [7] was proposed as an extension of ant colony system (ACS) [17] to deal with multiobjective problems. The original version of MACS uses one pheromone trail matrix and several heuristic information functions. However, in the case of the TSALBP-1/3, the experimentation carried out in [6] showed that the performance was better when MACS was only guided by the pheromone trail information. Therefore, the heuristic information functions were not used.

Since the number of stations is not fixed, the algorithm uses a constructive and station-oriented approach [2] to face the precedence problem (as usually done for the SALBP [10]). Thus, the algorithm will open a station and select one task till a stopping criterion is reached. Then, a new station is opened to be filled and the procedure is iterated till all the existing tasks are allocated.

The pheromone information has to memorise which tasks are the most appropriate to be assigned to a station. Hence, a pheromone trail has to be associated to a pair $(station_k, task_j)$, $k = 1...n$, $j = 1...n$, with n being the number of tasks, so the pheromone trail matrix has a bi-dimensional nature. Since MACS is Pareto-based, the pheromone trails are updated using the current non-dominated set of solutions (Pareto archive). Two station-oriented single-objective greedy algorithms were used to obtain the initial pheromone value τ_0.

In addition, a novel mechanism was introduced in the construction procedure in order to achieve a better search intensification-diversification trade-off able to deal with the problem difficulties. This mechanism randomly decides when to close the current station taking as a base both a station closing probability distribution and an ant filling threshold α_i. The probability distribution is defined by the station filling rate (i.e., the overall processing time of the current set of tasks S_k assigned to that station) as follows:

$$p\left(closing\ k\right) = \frac{\sum_{i \in S_k} t_i}{c} \tag{3}$$

At each construction step, the current station filling rate is computed. In case it is lower than the ant's filling percentage threshold α_i (i.e., when it is lower than $\alpha_i \cdot c$), the station is kept opened. Otherwise, the station closing probability distribution is updated and a random number is uniformly generated in $[0, 1]$ to take the decision whether the station is closed or not. If the decision is to close the station, a new station is created to allocate the remaining tasks. Otherwise, the station will be kept opened. Once the latter decision has been taken, the next task is chosen among all the candidate tasks using the MACS transition rule to be assigned to the current station as usual. The procedure goes on till there is no more remaining task to be assigned.

Thus, the higher the ant's threshold, the higher the probability of a totally filled station, and *vice versa*. This is due to the fact that there are less possibilities to close it during the construction process. In this way, the ant population will show a highly diverse search behaviour, allowing the algorithm to properly explore the different parts of the optimal Pareto front by appropriately distributing the generated solutions.

The interested reader is referred to [6] for a complete description of the MACS proposal for the TSALBP-1/3.

B. A multiobjective extension of a single-objective genetic algorithm

An extension of an existing single-objective genetic algorithm for the SALBP was proposed in [6] to deal with the TSALBP-1/3. The authors chose the proposal introduced in [8] and adapted it by means of the state-of-the-art multiobjective NSGA-II approach. In short, the features of this TSALBP-NSGA-II designed can be summarised as follows:

- Coding: The original order-based encoding scheme proposed in [8] is considered. The length of the chromosome is equal to the number of tasks. The task-station assignment is implicitly encoded in the genotype and it is obtained by using a simple station-oriented constructive mechanism [2] guided by fulfilling the available cycle time of each station. A station is opened and sequentially filled with the tasks listed in the chromosome order while the overall processing time of the set of assigned tasks does not exceed the assembly line cycle time. Once there is not available time to place the next task in the current station, this station is closed and a new empty one is opened to assign the remaining tasks. The procedure stops when all the tasks are allocated.
- Initial population: it is randomly generated by assuring the feasibility of the precedence relations.
- Crossover: A kind of order preserving crossover [18], [19] is considered to ensure that feasible offsprings are obtained satisfying the precedence restrictions. This family of order-based crossover operators emphasises the relative order of the genes from both parents. In our case, two different offspring are generated from the two parents to be mated, proceeding as follows. Two cutting points are randomly selected for them. The first offspring takes the genes outside the cutting points in the same sequence order as in the first parent. That is, from the beginning to the first cutting point and from the second cutting point to the end. The remaining genes, those located between the two cutting-points, are filled in by preserving the relative order they have in the second parent. The second offspring is generated the other way around, i.e. taking the second parent to fill in the two external parts of the offspring and the first one to build the central part. Notice that, preserving the order of the genes of the other parent in the central part will guarantee the feasibility of the obtained offspring solution in terms of precedence relations. The central genes also satisfy the precedence constraints with respect to those that are in the two external parts.
- Mutation: The same mutation operator considered in the original single-objective genetic algorithm [8], a scramble mutation, is used. A random cut-point is selected and the genes after the cut-point are randomly replaced (scrambled), assuring feasibility.
- Diversity: the similarity-based mating scheme for EMO proposed in [20] to recombine extreme and similar parents was used in this algorithm to try to improve the

diversity and spread of the Pareto set approximations.

This NSGA-II design for the TSALBP-1/3 showed poor results in comparison with MACS [6]. The generated Pareto front approximations showed a very low cardinality and converged to a narrow region located in the left-most zone of the objective space (i.e. solutions with small values of the number of stations, m). The latter fact is justified by the TSALBP-1/3 nature as a strongly constrained combinatorial optimization problem, which was not properly tackled by the global search algorithm considered (a multiobjective genetic algorithm) and by the basic order encoding used.

Nevertheless, in the next section we will propose an advanced EMO design able to overcome the problems of the latter basic multiobjective genetic algorithm and to successfully solve the TSALBP-1/3.

IV. An advanced NSGA-II-based approach for the TSALBP-1/3

The weak performance of the previous EMO algorithm (Section III-B) when solving the TSALBP-1/3 cannot be explained because of the chosen multiobjective genetic algorithm. It is well known that NSGA-II has shown a large success when solving many different multiobjective numerical and combinatorial optimization problems (see Chapter 7 in [21] for a detailed review classified in different application areas). On the contrary, that weak behaviour was due to the inherent characteristics of the combinatorial optimization problem being solved. In principle, the use of global search procedures as genetic algorithms could be less appropriate than constructive metaheuristics to deal with the TSALBP-1/3 because of the hard constraints (precedence relations and stations' cycle time limitation). In addition, the representation used does not seem to be adequate because it is not a natural coding for the problem.

Hence, a novel design is proposed, also based on the original NSGA-II search scheme [9]. However, a more appropriate representation and more effective operators are used to solve the TSALBP-1/3. From now on, the new algorithm will be referred as *advanced TSALBP-NSGA-II* because of its problem-specific design and potential application to other TSALBP variants. The previous method will be referred to as *basic TSALBP-NSGA-II* in order to stress the difference between both approaches. The main features and operators of the *advanced TSALBP-NSGA-II* are described in the next subsections.

A. Representation scheme

The most important problem of the *basic TSALBP-NSGA-II* method was the representation scheme, based on that usually considered by the existing genetic algorithm approaches for the SALBP. We should note that the SALBP is a single-objective problem and thus it is not strictly necessary to represent a solution as an assignment of tasks to stations to solve it. Instead, an order encoding is used to define a specific task ordering in a chromosome and the latter assignment is determined in a constructive fashion, as seen in Section III-B.

However, the latter representation is not a good choice for the TSALBP-1/3. It carries the problem of biasing the search to a narrow area of the Pareto front (as demonstrated by the experimental results in [6] and in the current contribution). Here is where our new proposal, the *advanced TSALBP-NSGA-II*, takes the biggest step ahead with respect to the existing basic algorithm. The new coding scheme introduced will explicitly represent task-station assignments regardless the cycle time of the assembly line, thus ensuring a proper search space exploration for the joint optimization of the number and the area of the stations. Furthermore, the representation will also follow an order encoding to facilitate the construction of feasible solutions with respect to the precedence relations constraints.

The allocation of tasks among stations is made by employing separators[1]. Separators are thus dummy genes which do not represent any specific task and they are inserted into the list of genes representing tasks. In this way, they define groups of tasks being 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, each one composed of a single task. Tasks are encoded using numbers in $\{1, \ldots, n\}$, as in the previous representation, while separators take values in $\{n+1, \ldots, 2 \cdot n - 1\}$. Hence, the genotype is again an order-based representation.

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. However, the cycle time limitation could be violated and it will be a task of the genetic operators to ensure feasibility with respect to that constraint.

In summary, the proposed representation shows two advantages. On the one hand, it is clear and natural and thus it fulfils the rule of thumb that the genetic coding of a problem should be a natural expression of it. On the other hand, the genotype keeps on being a permutation, thus allowing us to consider the extensively used genetic operators for this representation.

B. The crossover operator

The main difficulty when using non-standard representations is the design of a suitable crossover operator able to combine relevant characteristics of the parent solutions into a valid offspring solution. Nevertheless, as our representation is order-based, the crossover operator can be designed from a classical order-based one. Crossover operators of the latter kind

[1] We should notice that, although this representation is not very extended, the use of separators in an order encoding was previously considered in a document clustering application [22].

which have been suggested in the literature include partially mapped crossover (PMX), order crossover, order crossover # 2, position based crossover, and cycle crossover, among others [23]. We have selected one of the most extended ones, PMX, which has been already used in other genetic algorithm implementations for the SALBP (for example in [8]).

PMX generates two offspring from two parents by means of the following procedure: a) two random cut points are selected, b) for the first offspring, the genes outside the random points are copied directly from the first parent, and c) the genes inside the two cut points are copied but in the order they appear in the second parent. The same mechanism is followed up with the second offspring but with the opposite parents.

Thanks to our advanced coding scheme and to the use of a permutation-based crossover, 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 use of these kinds of operators is very extended in evolutionary computation when dealing with combinatorial optimization problems with hard restrictions [24]. They should be carefully developed as a poor design of the repair operator can bias the convergence of the genetic algorithm or can make the crossover operator lose useful information from the parents. The goals and methods of our repair operator are the following:

- Redistribute spare tasks among available stations: changing the order of the genes in the parents to generate the offspring can cause the appearance of stations with an excessive cycle time. The repair operator must reallocate the spare tasks in other stations. First, the critical stations (those exceeding the cycle time) and their tasks are localised. Then, the feasible stations to reallocate each task of the critical station, fulfilling the restrictions, are calculated. If one spare task can be reallocated in more than one different station, the algorithm will choose one of them randomly for the reallocation. This process is repeated till either the critical station satisfies the cycle time restriction or there is no feasible movement. In the latter case, the critical station will be randomly divided in two or more stations by adding the needed separators to balance the load.
- Removing empty stations: no empty stations are allowed. 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 the necessary separators [2].

C. Mutation operator

A mutation operator has been specifically designed and applied uniformly to the selected individuals of the population. It is based on reordering a part of the sequence of tasks and reassigning them to stations. It will be called scramble mutation operator.

The scramble mutation operator works as follows: after choosing two points randomly, the tasks between those points are scrambled forming a new sequence of tasks in such a way the mutated solution keeps on being feasible with respect to the precedence relations. The existing separators among the two mutation points are ignored and a new reallocation of those tasks is considered by randomly generating new separator locations within the task sequence.

To do so, a similar mechanism to the filling thresholds of the MACS algorithm have been followed (see Section III-A). The task sequence is analysed from left to right and each position has a random choice for the insertion of a separator. The probability distribution associated to the separator insertion depends on the current station filling rate according to the cycle time (see equation 3).

V. Experiments

This section is devoted to describe the experimental study developed to test our proposal. We first specify the problem instances, parameter values, and multiobjective performance indicators used for the computational tests. Then, our novel proposal is compared to the existing *basic TSALBP-NSGA-II* and the state-of-the-art algorithm for the TSALBP-1/3, MACS.

A. Problem instances and parameters

Six problem instances with different features have been selected for the experimentation: `arc111` (P1), `barthol2` (P2), `barthold` (P3), `lutz2` (P4), `scholl` (P5), and `weemag` (P6). The six TSALBP-1/3 instances considered are publicly available at `http://www.nissanchair.com/TSALBP`. Originally, these instances only had time information. However, their area information has been created by reverting the task graph to make them bi-objective (as done in [3]).

We executed each algorithm 10 times with different random seeds, setting a fixed run time as stopping criterion (900 seconds). This stopping criterion was chosen because it is broadly admitted and used in real-world industrial environments. Also, all the algorithms were launched in the same computer: Intel Pentium^{TM} D with two CPUs at 2.80GHz, and CentOS Linux 4.0 as operating system. Furthermore, the parameters of the developed algorithms and their operators are shown in Table I. The values of these parameters were selected after carrying out a preliminary experimentation.

B. Multiobjective performance indicators

We will consider two different multiobjective performance indicators [21] to evaluate the quality of the *advanced TSALBP-NSGA-II* proposal with respect to the TSALBP-1/3

[2] Notice that, the application of the current operator is not actually needed and it is more related to aesthetic reasons. The coding scheme, the designed genetic operators and the multiobjective fitness function would actually allow the algorithm to work with chromosomes encoding empty stations by directly ignoring them.

TABLE I: Used parameter values for the three algorithms presented in the experimentation.

Parameter	Value	Parameter	Value
Basic TSALBP-NSGA-II			
Population size	100	Ishibuchi's γ, δ values	10
Crossover probability	0.8	Mutation probability	0.1
MACS			
Number of ants	10	β	2
ρ	0.2	q_0	0.2
Ants' thresholds (2 ants per each)	$\{0.2, 0.4, 0.6,$ $0.7, 0.9\}$		
Advanced TSALBP-NSGA-II			
Population size	100		
Crossover probability	0.8	Mutation probability	0.1

TABLE II: Mean and standard deviation $\bar{x}(\sigma)$ of the HVR performance indicator values for the *advanced TSALBP-NSGA-II* (TN), and the state-of-the-art algorithms, MACS and the *basic TSALBP-NSGA-II* (BasTN) for instances P1 to P6. Higher values indicate better performance.

	HVR		
	P1	P2	P3
MACS	0.7860 (0.005)	0.7721 (0.01)	0.8104 (0.017)
BasTN	0.7854 (0.03)	0.6835 (0.070)	0.1502 (0.067)
TN	**0.9429 (0.041)**	**0.9175 (0.058)**	**0.8717 (0.049)**
	P4	P5	P6
MACS	0.7095 (0.02)	0.5682 (0.007)	0.8700 (0.009)
BasTN	0.0089 (0.011)	0.1731 (0.001)	0.4583 (0.209)
TN	**0.9747 (0.027)**	**0.9238 (0.044)**	**0.9622 (0.027)**

state-of-the-art, the MACS algorithm, and the *basic TSALBP-NSGA-II*.

On the one hand, we selected one unary performance indicator: the hypervolume ratio (HVR). The HVR [21] has become a very useful unary performance indicator. Its use is very extended as it can jointly measure the distribution and convergence of a Pareto set approximation. It can be calculated as follows:

$$HVR = \frac{HV(P)}{HV(P^*)}, \tag{4}$$

where $HV(P)$ and $HV(P^*)$ are the volume (S indicator value) of the Pareto front approximation and the true Pareto front, respectively. When HVR equals 1, then the Pareto front approximation and the true Pareto front are equal. Thus, HVR values lower than 1 indicate a generated Pareto front approximation that is not as good as the true Pareto front.

On the other hand, we have also considered the binary set coverage metric C to compare the obtained Pareto sets two by two based on the following expression:

$$C(P,Q) = \frac{|\{q \in Q\,;\, \exists p \in P : p \prec q\}|}{|Q|}, \tag{5}$$

where $p \prec q$ indicates that the solution p, belonging to the approximate Pareto set P, dominates the solution q of the approximate Pareto set Q in a minimisation problem.

Hence, the value $C(P,Q) = 1$ means that all the solutions in Q are dominated by or equal to solutions in P. The opposite, $C(P,Q) = 0$, represents the situation where none of the solutions in Q are covered by the set P. Note that both $C(P,Q)$ and $C(Q,P)$ have to be considered, since $C(P,Q)$ is not necessarily equal to $1 - C(Q,P)$.

We have used boxplots based on the C metric for showing the dominance degree of the Pareto sets of every pair of algorithms (see Figure 2). Each rectangle contains six boxplots representing the distribution of the C values for a certain ordered pair of algorithms in the six problem instances (P1 to P6). Each box refers to algorithm A in the corresponding row and algorithm B in the corresponding column and gives the fraction of B covered by A $(C(A,B))$.

In addition, we use attainment surface plots [25] to ease the analysis of the results. The attainment surfaces plots of 2 problem instances, P2 and P5, appear in Figures 3 and 4.

The reader should notice that the true Pareto sets of the six problem instances are not known. To overcome this problem we will consider a pseudo-optimal Pareto set, i.e. an approximation of the true Pareto set. It is obtained by merging all the (approximate) Pareto sets generated for each problem instance by any algorithm in any run. Thanks to this pseudo-optimal Pareto set we can compute the HVR performance indicator.

C. Experimentation and analysis of results

In this section we analyse the performance of the *advanced TSALBP-NSGA-II* against the state-of-the-art MACS and the *basic TSALBP-NSGA-II*. We use the multiobjective performance indicators considered in the previous section.

The results corresponding to the binary C indicator values on the 6 instances are represented by means of boxplots in Figure 2. The corresponding HVR values are included in Table II. Besides, attainment surfaces for some instances are plotted in Figure 3 and 4.

In view of the boxplots generated by the C indicator values (Figure 2), a clear analysis can be drawn: the *advanced TSALBP-NSGA-II* outperforms both MACS and the *basic TSALBP-NSGA-II* without any doubt. Hardly none of the solutions generated by the *advanced TSALBP-NSGA-II* are dominated by the solutions generated with the other methods. Only in instance P6, some solutions of the *advanced TSALBP-NSGA-II* are dominated by the other methods in some of the performed runs.

The same global analysis can be observed using the results of the unary indicator, HVR (Table II). The values of the Pareto front approximations generated by the *advanced TSALBP-NSGA-II* are the highest ones. Thus, the convergence and diversity achieved by the *advanced TSALBP-NSGA-II* are higher than the state-of-the-art algorithms in all the instances. See for example the HVR values of problem instance P5 to check the big difference. On the contrary, the *basic TSALBP-*

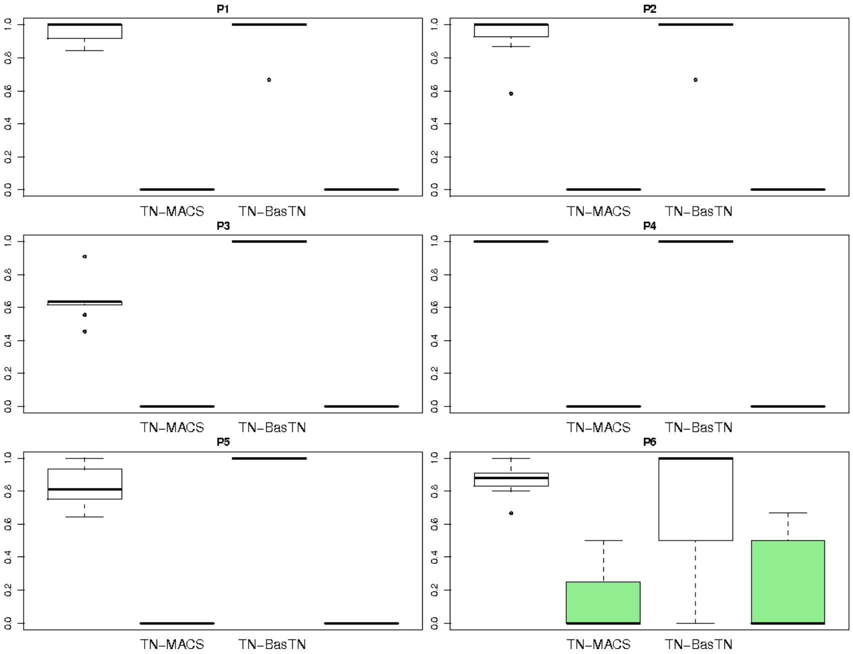

Fig. 2: Boxplots representing the binary C indicator values for comparisons between the *advanced TSALBP-NSGA-II* (TN) and the state-of-the-art algorithms, MACS and *basic TSALBP-NSGA-II* (BasTN) for instances P1 to P6. White boxplots correspond to $C(TN, MACS/BasTN)$, coloured boxplots to $C(MACS/BasTN, TN)$. Larger values indicate better performance.

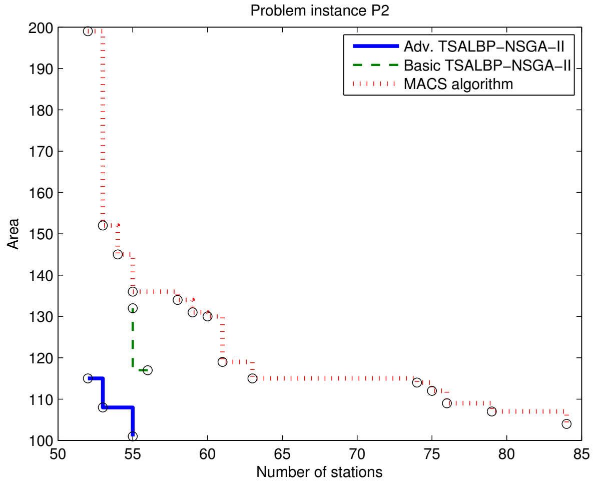

Fig. 3: Attainment surfaces plot for instance P2 (*barthol2*).

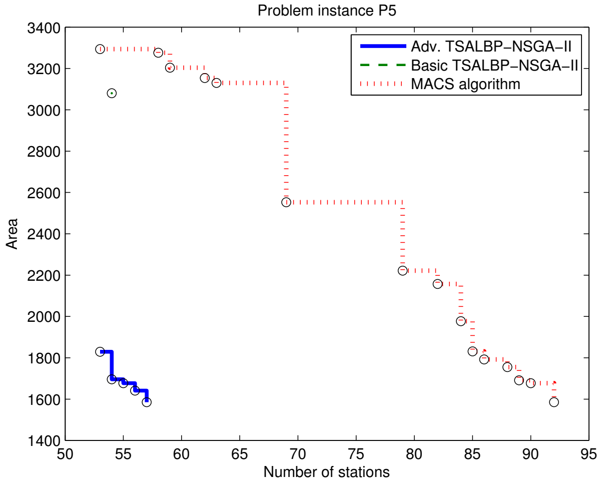

Fig. 4: Attainment surfaces plot for instance P5 (*scholl*).

NSGA-II has a very low convergence to the whole Pareto front (e.g. see the HVR values of problem instance P4).

Finally, the overall excellent performance of the *advanced TSALBP-NSGA-II* can also be remarked in the attainment surfaces of Figures 3 and 4. There is a high distance between the attainment surfaces obtained by the *advanced TSALBP-NSGA-II* and those corresponding to the other algorithms in both instances, P2 and P5. We can also see the solutions obtained by the *basic TSALBP-NSGA-II* in both figures. As explained in Section III-B, although the *basic TSALBP-NSGA-II* is able to reach better solutions than the MACS algorithm in a specific small region of the Pareto front, its performance is worse in the rest of it. The latter fact motivated the design of an EMO algorithm able to outperform both algorithms in all the Pareto front. This goal was clearly achieved by the proposal presented in this work.

In view of this experimental study, it can be seen that every included component in the *advanced TSALBP-NSGA-II*, i.e.

individual representation and operators, are more faithful to the solution phenotype and thus more appropriate for the problem solving. This fact helps to increase the performance of the algorithm and achieve the best intensification-diversification trade-off in the multiobjective search space, obtaining better results than the MACS algorithm.

VI. Conclusions

A novel multiobjective genetic algorithm design has been proposed to tackle the TSALBP-1/3 resulting in a new approach called *advanced TSALBP-NSGA-II*. The performance of this new algorithm has been compared with the state-of-the-art algorithms, the multiobjective ACO approach, MACS, and a previous multiobjective extension of an existing genetic algorithm for the SALBP, called *basic TSALBP-NSGA-II*. The comparisons were carried out using up-to-date multiobjective performance indicators. The *advanced TSALBP-NSGA-II* clearly outperformed the latter two methods when solving six TSALBP-1/3 instances considered.

It has been demonstrated that the existing *basic TSALBP-NSGA-II* showed a poor performance due to the use of non-appropriate representation and genetic operators to solve the problem. Since the TSALBP-1/3 is a very complex combinatorial optimization problem with strong constraints, a deep study of the best design options for the specific context was mandatory to get a high performance problem solving technique. Therefore, it can be asserted that EMO algorithms are suitable to solve this kind of multiobjective assembly line balancing problems, if a good design is used.

Future work is devoted to improve the performance of the EMO proposal with new diversity mechanisms to provide better Pareto approximation sets. In addition, we aim to add interactive preferences into the multiobjective algorithm to guide the search to the Pareto front regions preferred by the expert [26]–[28].

Acknowledgement

This work has been supported by the Spanish Ministerio de Ciencia e Innovación (MICINN) under project TIN2009-07727, including EDRF fundings.

References

[1] I. Baybars, "A survey of exact algorithms for the simple assembly line balancing problem," *Management Science*, vol. 32, no. 8, pp. 909–932, 1986.

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

[3] J. Bautista and J. Pereira, "Ant algorithms for a time and space constrained assembly line balancing problem," *European Journal of Operational Research*, vol. 177, pp. 2016–2032, 2007.

[4] V. Chankong and Y. Y. Haimes, *Multiobjective Decision Making Theory and Methodology*. North-Holland, 1983.

[5] F. Glover and G. A. Kochenberger, Eds., *Handbook of Metaheuristics*. Kluwer Academic, 2003.

[6] 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.

[7] B. Barán and M. Schaerer, "A multiobjective ant colony system for vehicle routing problem with time windows," in *21st IASTED International Conference*, Innsbruck (Germany), February 2003, pp. 97–102.

[8] I. Sabuncuoglu, E. Erel, and M. Tayner, "Assembly line balancing using genetic algorithms," *Journal of Intelligent Manufacturing*, vol. 11, pp. 295–310, 2000.

[9] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[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] A. Scholl and S. Voss, "Simple assembly line balancing- heuristic approaches," *Journal of Heuristics*, vol. 2, pp. 217–244, 1996.

[12] Y. K. Kim, W. S. Song, and J. H. Kim, "A mathematical model and a genetic algorithm for two-sided assembly line balancing," *Computers and operations research*, vol. 36, no. 3, pp. 853–865, 2009.

[13] W. C. Chiang, "The application of a tabu search metaheuristic to the assembly line balancing problem," *Annals of Operations Research*, vol. 77, pp. 209–227, 1998.

[14] A. Heinrici, "A comparison between simulated annealing and tabu search with an example from the production planning," in *Operations Research Proceedings 1993*, H. Dyckhoff, U. Derigs, and M. Salomon, Eds., Berlin, Germany, 1994, pp. 498–503.

[15] Y. Y. Leu, L. A. Matheson, and L. P. Rees, "Assembly line balancing using genetic algorithms with heuristic-generated initial populations and multiple evaluation criteria," *Decision Sciences*, vol. 25, pp. 581–606, 1994.

[16] P. R. McMullen and P. Tarasewich, "Multi-objective assembly line balancing via a modified ant colony optimization technique," vol. 44, pp. 27–42, 2006.

[17] M. Dorigo and L. Gambardella, "Ant colony system: a cooperative learning approach to the traveling salesman problem," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 53–66, 1997.

[18] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Publishing, 1989.

[19] T. Bäck, D. Fogel, and Z. Michalewicz, Eds., *Handbook of Evolutionary Computation*. IOP Publishing and Oxford University Press, 1997.

[20] H. Ishibuchi, K. Narukawa, N. Tsukamoto, and Y. Nojima, "An empirical study on similarity-based mating for evolutionary multiobjective combinatorial optimization," *European Journal of Operational Research*, vol. 188, no. 1, pp. 57–75, 2008.

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

[22] A. M. Robertson and P. Willett, "Generation of equifrequent groups of words using a genetic algorithm," *Journal of Documentation*, vol. 50, no. 3, pp. 213–232, 1994.

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

[24] P. Chootinan and A. Chen, "Constraint handling in genetic algorithms using a gradient-based repair method," *Computers and Operations Research*, vol. 33, no. 8, pp. 2263–2281, 2006.

[25] C. M. Fonseca and P. J. Fleming, "On the performance assessment and comparison of stochastic multiobjective optimizers," in *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN)*, ser. Lecture Notes in Computer Science, vol. 1141, Berlin, Germany, 1996, pp. 584–593.

[26] M. Chica, O. Cordón, S. Damas, and J. Bautista, "Including different kinds of preferences in a multi-objective ant algorithm for time and space assembly line balancing on different nissan scenarios," *Expert Systems with Applications*, vol. 38, pp. 709–720, 2011.

[27] M. Chica, O. Cordón, S. Damas, J. Bautista, and J. Pereira, "Incorporating preferences to a multi-objective ant algorithm for time and space assembly line balancing," in *Ant Colony Optimization and Swarm Intelligence*, ser. Lecture Notes in Computer Science, vol. 5217. Berlin, Germany: Springer, September 2008, pp. 331–338.

[28] M. Chica, O. Cordón, S. Damas, and J. Bautista, "Integration of an emo-based preference elicitation scheme into a multi-objective aco algorithm for time and space assembly line balancing," in *2009 IEEE International Conference on Multicriteria Decision Making (IEEE MCDM 2009)*, Nashville (USA), April 2009, pp. 157–162.