International Journal of Production Research, 2018
https://doi.org/10.1080/00207543.2018.1427899

Modelling and solving a cost-oriented resource-constrained multi-model assembly line balancing problem

Jordi Pereira*

Faculty of Engineering and Sciences, Universidad Adolfo Ibáñez, Viña del Mar, Chile

(*Received 14 March 2017; accepted 27 December 2017*)

A line balancing problem considers the assignment of operations to workstations in an assembly line. While assembly lines are usually associated to mass production of standardised goods, their advantages have led to their widespread use whenever a product-oriented production system is applicable and the benefits of the labour division and specialisation are significant, even when some of its characteristics may deviate from classical assembly lines. In this work, we study a line balancing problem found in the textile industry in which the line must be balanced for multiple types of goods taking into account resource requirements. In order to solve the problem, a hybrid method that combines classical methods for line balancing with an Estimation of Distribution Algorithm is proposed. Computational experiments show that the new procedure improves upon the state of the art when compared using a benchmark set derived from the literature, as well as when compared using data from the manufacturer that originated this research work.

Keywords: assembly line balancing; combinatorial optimisation; lower bound; heuristics; estimation of distribution algorithm

1. Introduction

Assembly line balancing constitutes an integral part of the proper design and management of an assembly line. An assembly line is a product-oriented manufacturing method in which the product flows through each of its workstations (also referred to as stations) until the finished product is obtained. The assembly line balancing problem (ALBP) considers the division of the assembly operations (also referred to as tasks) among stations in order to maximise the total efficiency of the line.

Different ALBPs have been extensively studied in the literature. The most thoroughly studied problem is a simplified version known as the simple assembly line balancing problem (SALBP). In a SALBP, the assembly of a product is divided into elementary operations that need to be assigned to the workstations. Each operation requires a known, deterministic task time and needs to be performed after some other tasks, imposing precedence constraints among them. In addition to this, workstations have an identical amount of time to perform their assigned operations, which is known as cycle time and defines the production rate of the line. The objective is to find an assignment of tasks to workstations that optimises the efficiency of the line.

Summarising, the SALBP considers minimising the total idle time of the line under two sets of constraints: (1) cumulative constraints on the operation times on each workstation and (2) precedence constraints among tasks. When additional constraints are considered, or the objective takes into account additional features, the problem is considered to be a general ALBP or GALBP.

GALBP formulations build upon the SALBP formulation by taking into account additional considerations and by adapting the objective and the constraints to the requirements of real-life line balancing problems, see Boysen, Fliedner, and Scholl (2007) and Dolgui and Battaïa (2013) for a literature review and a classification scheme for the GALBP.

For instance, the problem investigated in this paper takes into account the particularities of a clothing company based in Chile. As their product is targeted towards the higher end market, the company chose to keep production local and in-house, focusing its strategy on responsiveness to variety and service times. In order to cope with the increasing product variety, while still benefiting from division of labour, specialisation and standardisation, the manufacturing process of different items of clothing is performed in a single assembly line in small batches, each corresponding to a few hours worth of work, leading to a multi-model assembly line.

Moreover, in order to reduce operational costs, the workstations of the line are not equally equipped and the skills of the workers differ. Consequently, operations that require specific machinery and skills are grouped together so as to reduce wages and investment costs, leading to a line balancing problem with resource assignment features and a cost-oriented objective.

*Email: jorge.pereira@uai.cl

These factors are not exclusive of the company under study, as recent literature reviews (ElMaraghy et al. 2013; Hazir, Delorme, and Dolgui 2015) point out. ElMaraghy et al. (2013) highlights the relevance of managing product variety throughout the enterprise, including production. Along similar lines, Boysen, Fliedner, and Scholl (2009) reviews integrated production planning within assembly lines, highlighting the relevance of jointly considering line balancing along with other production planning decisions, and Tonelli et al. (2013) provides an example of the application of such an integrated approach. Similarly, Hazir, Delorme, and Dolgui (2015) highlights the increasing importance of considering costs and equipment decision within assembly line balancing in all phases of the product life cycle. Moreover, Hazir, Delorme, and Dolgui (2015) stresses that the current availability of accurate data within many companies allows a more detailed line balancing problem as the one proposed in the present work. Another line of work that stresses the importance of the concepts tackled in the present paper can be found in the review of workforce planning with skills (De Bruecker et al. 2015).

When compared to the SALBP, the proposed problem includes a cost-oriented objective; focuses on the assignment of the tasks, the production resources and workforce to the workstations; and takes into account the multiple models within a single assembly line. Consequently, the problem is denoted by cost-oriented, resource-constrained, multi-model assembly line balancing problem, in short C-R-MMALBP, and its formulation models the line balancing problem that originated this research as well as other heterogeneous assembly lines with similar characteristics.

1.1 *Literature review*

The literature on line balancing is fairly extensive, (see Boysen, Fliedner, and Scholl 2007; Dolgui and Battaïa 2013; Hazir, Delorme, and Dolgui 2015; Scholl and Becker 2006, for a comprehensive treatment of different previous work on assembly line balancing). Consequently, on the one hand, we focus our attention on works featuring similar characteristics to the proposed problem, i.e. line balancing with multiple models, and heterogeneity among stations leading to equipment decisions and a cost-oriented objective, and on the other hand, we also provide a state of the art on solution procedures for line balancing problems. Additionally, we refer to other sources, like Swift, Booker, and Edmondson (2004) and Tonelli et al. (2013) for a discussion of other issues of relevance to this work, like technological aspects and the integration of line balancing and equipment selection with other planning decisions.

1.1.1 *Multi-model assembly line balancing*

The literature covers two basic approaches on assembly lines with multiple models, mixed-model and multi-model assembly line balancing, see Dolgui and Battaïa (2013). The main difference between a mixed-model and a multi-model line lies in the variability among the manufactured units. In a mixed-model assembly line, a single product with multiple variations (models) is assembled. While the variability among the models needs to be taken into account during line balancing, the models are scheduled in a specific order to minimise possible work overload. A classical example corresponds to an automotive assembly line in which different models (e.g. with/without sun roof) of the same car are produced. In a multi-model assembly line, differences among models are more significant and force the line to rely on batch production. An example corresponds to the line investigated in the case study, in which different garments are manufactured in a single line. In this case, line balancing takes into account distinct but similar assembly operations.

The literature has mostly focused on mixed-model assembly lines (Bock 2008; Bukchin and Rabinowitch 2006; Emde, Boysen, and Scholl 2010; Kimms 2000; Yagmahan 2011, among others). These studies can be divided according to the method used to tackle the cumulative (cycle time) constraints of different products. Some methods consider a weighted average of the task times for all of the models (Yagmahan 2011) while others consider a separate constraint for each model (Bukchin and Rabinowitch 2006). Note that other approaches are possible (Bock 2008; Emde, Boysen, and Scholl 2010).

Studies on multi-model assembly lines are less usual, and they tend to address the disturbances caused by product changeover, like set-up times (Scholl, Boysen, and Fliedner 2008), or the buffers required for smooth operations (Chakravarty and Shtub 1985).

In this work, the model considers batched production, and thus the multi-model nature of the problem is represented as cumulative constraints like in the mixed-model problem found in Bukchin and Rabinowitch (2006). Note that line balancing problems with multiple cumulative constraints have been studied in the literature, both in abstract terms (Scholl, Fliedner, and Boysen 2010) and in specific applications (Bautista and Pereira 2011).

1.1.2 *Station heterogeneity*

Differences among stations have also been considered in the literature and greatly affect how the problem is formulated. For instance, the literature covers works in which heterogeneous manpower is considered (Borba and Ritt 2014; Vilà and Pereira 2014; Moreira et al. 2015; Moreira et al. 2017), or robotic machinery is to be selected (Gao et al. 2009).

These works focus on efficiency based objectives, such as minimising the cycle time for a given workforce or minimising the workforce for a given throughput. Problems with cost- or profit-oriented objectives are significantly different than their efficiency-oriented counterparts. Moreover, there are fewer references on the topics, as pointed out by a recent review, (Hazir, Delorme, and Dolgui 2015).

Heterogeneity has also been tackled from a cost perspective. For example, in some cases the equipment of the workstations is to be selected (Pekin and Azizoglu 2008), the tasks require specific skill levels to be performed (Amen 2006; Roshani et al. 2012), the costs of the robots installed in each workstation are to be minimised (Nilakantan et al. 2017), or the processing times of the tasks and the costs vary according to the station resources, be it equipment or personnel (Hamta et al. 2011; Sungur and Yavuz 2015). Other heterogeneous problems include the design of reconfigurable manufacturing systems (Youssef and Elmaraghy 2006; Dou, Dai, and Meng 2010) and the transfer line balancing problem (Battaïa et al. 2014; Borisovsky, Delorme, and Dolgui 2014) in which tool selection is an integral part of the problem. Also, in some cases (Scholl, Fliedner, and Boysen 2010), equipment is implicitly considered in the problem.

The literature also contains some studies with cost-oriented or configuration selection objectives. The characteristics of each case study greatly affect the formulation. For instance, in Tuncel and Topaloglu (2013) costs are incurred if parallel stations are installed, while Battaïa et al. (2015) incorporates some station equipment decisions while aiming at adjusting the number of workers per workstation.

A more detailed review of alternative formulations is provided in Section 2.4, in which the proposed model is compared with those found in the literature.

1.1.3 *Solution procedures*

Among the general purpose procedures for line balancing problems, Construction-based methods are considered the state of the art. Specifically, truncated enumeration schemes like the Hoffmann heuristic (Hoffmann 1963) are to be highlighted. The quality of the solutions reported by the Hoffmann heuristic in the reference benchmark set (Sternatz 2014) and its applicability to more general line balancing problems (Sternatz 2014; Sternatz 2015), lead us to consider the heuristic as a state-of-the-art procedure heuristic for ALBPs. It is also important to highlight that branch-and-bound-based methods for the SALBP (Morrison, Sewell, and Jacobson 2014) are able to solve large-size instances to optimality. This is mostly due to the availability of very good bounding methods for the SALBP (Pereira 2015), but does not hold for GALBPs.

Note that metaheuristic approaches, like ant colony optimisation (Bautista and Pereira 2007), iterated local search (Borba and Ritt 2014), or genetic algorithms (Chen et al. 2012; Gao et al. 2009), are also common in the resolution of GALBPs, and provide some of the state-of-the-art methods for these problems.

1.2 ***Paper outline and contributions***

This paper tackles a novel multi-model assembly line balancing problem in which variability is considered in the model as constraints that need to be satisfied. Moreover, the objective tries to minimise the solution cost including workforce and resource requirements. The remainder of the paper is organised as follows:

- Section 2 provides a description of the problem in hand, which is denoted by the cost-oriented resource-constrained multi-model assembly line balancing problem (C-R-MMALBP). The problem aims at minimising the total workforce and resource (machinery) costs in a multi-model assembly line in which stations need to be equipped with specific machinery in order to perform some of the tasks.
  In order to illustrate the characteristics and limitations of the problem, the assumptions behind the model and two formulations are put forward. Moreover, the ability of the model to depict real-life requirements is also investigated in an industrial example derived from the garment industry.
- Section 3 is devoted to the description of the different algorithmic proposals of this work and discusses their corresponding solution methods.
  Specifically, a modified hybrid heuristic based on the hybridisation of the Hoffmann heuristic, the state-of-the-art method for the SALBP (Sternatz, 2014), and an Estimation of Distribution Algorithm (Larrañaga and Lozano 2001) is described.
- Section 4 reports the results of a computational experiment verifying the applicability of the proposed approach.
  The hybrid approach is shown to be robust and to outperform an adaptation of the state-of-the-art method for assembly line balancing, highlighting its ability to efficiently solve complex general line balancing problems.
- Some conclusions and future lines of work are provided in Section 5. Among the considered issues, the applicability of the proposed formulation and solution methods is discussed.

Additionally, a description of a lower bound based on the best-performing lower bound for the SALBP (Pereira, 2015) for the problem is provided in Appendix 1.

2. The C-R-MMALBP

In order to provide a formal description of the problem and its mathematical formulations, see Section 2.3, we first introduce the assumptions of the problem in Section 2.1, and describe the situation that originated the current work in Section 2.2. The proposed formulations are then compared to previous models found in the literature, see Section 2.4.

2.1 *Assumptions of the C-R-MMALBP model*

To describe the problem and its formulation, we first introduce its assumptions, see Baybards (1986) and Scholl and Becker (2006) for the original assumptions of the SALBP.

(A-1) All input parameters are known with certainty.
(A-2) Tasks are to be performed in a single station.
(A-3) The assembly sequence must comply with technological precedence requirements.
(A-4) All tasks must be processed.
(A-5) Stations are equipped according to the tasks they perform. This reflects that the labour and machinery costs differ according to the task assignments.
(A-6) Processing times are fixed and not sequence dependent.
(A-7) Tasks can be processed in any station equipped with the machinery and the workers required to perform the task.
(A-8) A single serial line is considered.
(A-9) The line is balanced taking into account multiple products. These products are performed in batches but tasks required by two or more products must be assigned to the same station.

Note that the above assumptions are basically the same ones described for the SALBP, with the exception of assumptions (A-5), (A-7) and (A-9). Assumption (A-5) leads to a cost-oriented objective that minimises a function of workforce and equipment costs.

Assumption (A-7) models the resource constraints. The proposed model formulates these constraints as the additional requirements that a workstation must fulfil in order to perform a task. That is, the workstation must equip a resource (be it a specific machine or a skill of the worker) in order to perform some specific tasks, and an additional cost is incurred in order to equip the task with the said resource. Consequently, and focusing on the worker skills, these resources can be seen as categorical skills according to the classification given in De Bruecker et al. (2015) within the workforce planning literature.

Assumption (A-9) describes the multi-model nature of the assembly line and implies that products may have different production rates, hence specific cycle times, since the assembly operation is organised in batches. Furthermore, assumption (A-9) also imposes that identical tasks in any batch are performed by the same worker in order to minimise possible errors while enforcing specialisation and division of labour.

2.2 *Industrial example*

This study stems from an apparel company based in Chile. The company was founded in 1977 and designs and manufactures its own clothing lines. The company targeted a high-income market segment and it traditionally worked using a vertical supply chain, in which the product was sold in different retailers owned by the company. Due to demographic changes, the competition of foreign brands and the irruption of prêt-à-porter companies, the company was forced to modify its business model. Moreover, its traditional competitor went bankrupt in 2012 and ceased operations. These conditions have led the company to sell its products through department stores in addition to its own establishments, concentrate its catalogue in approximately 50 products per season, and enforce policies to reduce its operational costs. Additionally, management has become averse to investments and inventories, even when the additional distribution channels have increased gross sales.

The company has two production units in Chile, one that performs cutting and dying operations, and a second one responsible for sewing, ironing and packaging operations. The focus of the study is the second unit, which operates as a multi-model assembly line resembling the assumptions described above. Daily operation is organised into small batches of production comprising several hours. Depending on the batches to perform, the staff is assigned to specific operations and functions, and surplus personnel is allocated to secondary operations.

Appendix 2 provides the data provided by the company consisting in the precedence graph and the processing times of the 48 elementary tasks into which the production of alpaca wool sweaters is divided. Cycle times and task times have been adjusted to a 1000-time units scale, as the adjustment has no consequences for the optimisation process (Otto, Otto, and Scholl 2013).

The data only consider the differences between the male and the female models, since other differences are not considered by the decision-maker and are not reported in the time and motion study available. Moreover, only one resource is considered, which corresponds to the use of skilled personnel. Due to differences in pay scale among workers, it is desirable to minimise the use of skilled workers, who are those workers proficient at operating an overlock sewing machine and a cutting machine. Consequently, these abilities are deemed as the resources of the problem and represent additional costs, since the wages of this skilled workers are higher. While other resources are present, the decision-maker is mainly focused on minimising personnel costs and does not take into account, or checks, the use of other resources. Therefore, these other resources are considered to be available in unlimited quantities.

The resulting problem is a two-model line balancing problem with one resource. While it only deviates slightly from the classic SALBP, its differences are still clear and stress the significance of further studying their impact within the line balancing decision process. Specifically, the lack of information regarding the differences among multiple models and the simplifying assumption neglecting other resources are partially attributed to the inability of the current decision-making process to integrate these features within its decision process. Consequently, improvements on the ability to solve the problem would raise the usefulness of information and thus lead to better models. The description of the various line balancing configurations is deferred to Section 4.3.

2.3 *Two integer linear programming formulations*

In order to provide a mathematical model for the C-R-MMALBP, some notation is required. Let $K = \{1, \dots, |K|\}$ be an ordered set of stations and let $P = \{1, \dots, |P|\}$ be the set of different products that the assembly line manufactures. Following the approach of Thomopoulos (1970), the assembly operations of all the products are described using a single precedence graph $G(V, A)$, in which the nodes, $V = \{1, \dots, |V|\}$, correspond to the elementary operations required by any product, and the arcs, $(i, j) \in A$, depict a precedence relation in any product between two operations. For convenience, precedence relations are sometimes represented using sets for each task of direct predecessors and successors, P_i and F_i, respectively, as well as sets of indirect predecessors and successors, P_i^* and F_i^*, which can be derived from $G(V, A)$. Note that the use of a single precedence graph is not required, and a different graph can be used for each product, see Bukchin and Rabinowitch (2006). In such a case, the set of predecessors, P_i, and successors, F_i, required by the formulations are defined as the union of the respective sets for each product.

For each task, $i \in V$, and product, $p \in P$, the operation time, t_{ip}, required to perform operation i on a unit of product p is known and deterministic, and it is set to 0 when the product does not need the said operation. In order to perform the operations assigned to each station, stations have an allotted time equal to the cycle time of the product, c_p, $p \in P$.

Some operations require resources, $R = \{1, \dots, |R|\}$, which can be acquired at a cost $(c_r,\ r \in R)$. Let a_{kr}, $k \in K$, $r \in R$, represent whether resource r is available at workstation k, and let n_{ir} denote whether resources $r \in R$ are required to elaborate operation, $i \in V$. Then, if workstation k is to perform operation i, $n_{ir} \leq a_{kr}$ must hold for each resource, r. Note that resources usually represent machinery or workers with some specific skills, and therefore the requirement usually takes a binary value, but the model could be easily modified to account for cumulative requirements. Finally, let c^w represent the cost associated to a station with no special resources, a baseline cost.

Given the previous notation, we can define the C-R-MMALBP as the problem of finding an assignment of tasks and additional resources to workstations such that precedence, cycle time and resource requirements are satisfied and the weighted cost associated to the number of workstations and required resources is minimised.

Equations (1)–(8) provide a mixed integer linear programming formulation of the problem using a set of binary decision variables x_{ik} whose value is equal to 1 if operation i is assigned to station k, or 0 otherwise; a set of auxiliary binary variables y_k to represent whether an item has been assigned to station k; and a set of auxiliary variables a_{kr} to represent whether resource r is required by station k (variables a_{kr} are treated as continuous without affecting the validity or the solution provided by the model),

$$\min \sum_{k \in K} c^w y_k + \sum_{r \in R} c_r \sum_{k \in K} a_{kr} \tag{1}$$

subject to:

$$\sum_{k \in K} x_{ik} = 1 \qquad \forall i \in V \tag{2}$$

$$\sum_{i\in V} t_{ip}\; x_{ik} \le c_p y_k \qquad \forall p \in P,\ \forall k \in K \tag{3}$$

$$\sum_{k\in K} k\; x_{ik} \le \sum_{k\in K} k\; x_{i'k} \qquad \forall i \in V,\ i' \in F_i \tag{4}$$

$$n_{ir}\; x_{ik} \le a_{kr} \qquad \forall i \in V,\ \forall r \in R,\ \forall k \in K \tag{5}$$

$$0 \le a_{kr} \le 1 \qquad \forall k \in K,\ \forall r \in R \tag{6}$$

$$x_{ik} \in \{0, 1\} \qquad \forall i \in V,\ \forall k \in K \tag{7}$$

$$y_k \in \{0, 1\} \qquad \forall k \in K \tag{8}$$

Objective (1) minimises the weighted sum of costs associated to stations and resources. Constraint set (2) ensures that all tasks are processed in a single station, constraint set (3) enforces the cycle time constraint for each product, and constraint set (4) enforces precedence constraints. When the problem contains a single product, equations (2)–(4) correspond to the constraint set of the SALBP. Constraint set (5) certifies that stations contain the resources required to perform the assigned tasks. Finally, the domain of the variables is defined in constraint sets (6)–(8).

An alternative reformulation that makes use of the feasible assignment concept follows. A feasible assignment corresponds to a set of tasks and is represented by a vector $\mathbf{q}$ of length $|V|$ such that $q_i = 1$ if task i belongs to the assignment and 0 otherwise. A feasible assignment fulfils the two following conditions: (1) the cumulative constraints (cycle time) of each product are satisfied $\left(\sum_{i\in V} t_{ip} \le c_p\ , \forall p \in P\right)$, and (2) the precedence constraints among tasks in the assignment are satisfied (if $q_i = 1, q_{i'} = 1$ and $i' \in F_i^*$ then $i'' \in F_i^* \cap P_{i'}^*$ implies $q_{i''} = 1$).

For any given feasible assignment, the minimum resource requirements are easily computed. Since the cost function (1) can be divided into the separate contributions of each station, for any given feasible assignment $\mathbf{q}$, its cost $\bar{c}_{\mathbf{q}}$ corresponds to (9), in which the cost of opening a station and the cost of each of the resources used by assignment $\mathbf{q}$ are summed).

$$\bar{c}_{\mathbf{q}} = c^w + \sum_{r\in R} c_r \min\left\{1, \sum_{i\in V} q_i n_{ir}\right\} \tag{9}$$

Let Q be the set of all feasible assignments, let z_{qk} be a binary variable that takes value 1 if station k uses assignment $\mathbf{q} \in Q$ and 0 otherwise. Then, formulation (10)–(14) is also a valid model for the C-R-MMALBP.

$$\min \sum_{k\in K} \sum_{q\in Q} \bar{c}_q z_{qk} \tag{10}$$

subject to:

$$\sum_{k\in K} \sum_{q\in Q} q_i\; z_{qk} = 1 \qquad \forall i \in V \tag{11}$$

$$\sum_{q\in Q} z_{qk} \le 1 \qquad \forall k \in K \tag{12}$$

$$\sum_{k\in K} \sum_{q\in Q} k\; q_i\; z_{qk} \le \sum_{k\in K} \sum_{q\in Q} k\; q_{i'}\; z_{qk} \qquad \forall (i, i') \in A \tag{13}$$

$$z_{qk} \in \{0, 1\} \qquad \forall q \in Q, k \in K \tag{14}$$

Objective (10) minimises the total cost of the stations. Constraint set (11) ensures that each task belongs to one of the chosen feasible assignments; constraint set (12) states that each station can only perform one feasible assignment; constraint set (13) imposes the precedence constraints and constraint set (14) defines the domain of the variables.

Note that formulation (10)–(14) does not explicitly manage the products nor the resources of the line balancing problem. These issues are transferred to an auxiliary problem (known as the pricing problem) which identifies assignments that are feasible according to constraint set (3) and calculates the costs of the assignment according to expression (9). Moreover, while the number of variables and constraints of the first formulation is polynomial to the number of tasks, models and resources, the number of variables of the second formulation is not polynomially bounded by the number of tasks of the instance, further complicating its resolution. Consequently, the line balancing literature usually uses the second formulation as a basis to design lower bounding procedures, see Peeters and Degraeve (2006), Pereira (2015). These lower bound procedures are solved by iteratively solving a pricing problem, and including the required feasible station assignments in Q when needed. Appendix 1 provides a detailed description of the proposed lower bound.

2.4 *Critical review of related line balancing formulations*

The C-R-MMALB formulation shares similarities with some previously studied problems. For example, some assembly line design problems contain similar trade-offs and characteristics to the C-R-MMALBP. Wilhelm and Gadidov (2004) considers the trade-offs between alternative operation assignments and the fixed costs of the stations, the machines, and the tools. The formulation allows a station to be equipped with different machines, and the cycle time constraints are imposed in each machine separately and according to its production rate. Hence, the model is effectively allowing parallel processing. Similar models with only one machine per station have been proposed in Pinnoi and Wilhelm (1997), Pinnoi and Wilhelm (1998) and Gadidov and Wilhelm (2000). All these works consider machines with different task times, and therefore their resources display hierarchical attributes rather than categorical attributes as in the C-R-MMALBP.

Bukchin and Tzur (2000) considers an assembly line in which the objective is to minimise the equipment costs of the workstations as the C-R-MMALBP, yet the formulation is not multi-model and only one equipment can be selected for each workstation. An analytic hierarchy process (AHP) approach to evaluate different line designs with similar characteristics can be found in Özdemir and Ayağ (2011).

Bukchin and Rabinowitch (2006) also studies a problem similar to the C-R-MMALBP in which line balancing for multiple products is jointly considered, but a common task among products can be assigned to different stations if a cost is paid. Hence, the focus of the problem is to balance task duplication and additional station costs, whereas in the C-R-MMALBP the objective tries to assign tasks with similar resource requirements to the same station.

The model proposed in Pekin and Azizoglu (2008) also shares similarities with the C-R-MMALBP, but a single model is considered, the processing times depend on the equipment and the problem is formulated within a bicriteria optimisation framework that tries to balance equipment costs and the number of required workstations. A U-shaped version of this problem is studied in Kazemi et al. (2011).

Another model that also shares similarities with the C-R-MMALBP was proposed in Corominas, Ferrer, and Pastor (2011). A single-model assembly line is considered and the costs depend on the number of resources allotted to each workstation. Moreover, there are alternatives to perform each task from which only one needs to be fulfilled. The C-R-MMALBP formulation differs in three aspects: it considers a multi-model line with cycle time constraints for each product; there is a single possible combination of resources to perform a task; and the nature of the resources is categoric.

Transfer line balancing (Battaïa et al. 2014; Borisovsky, Delorme, and Dolgui 2014), and reconfigurable manufacturing systems (RMS), (see Dou, Dai, and Meng 2010; Goyal and Jain 2016; Renzi et al. 2014; Youssef and Elmaraghy 2006, among others), also bear similarities to the problem. In all these problems the equipment of the workstations is selected among alternatives. The main difference between the C-R-MMALBP and RMS or transfer line balancing is that in the C-R-MMALBP tasks cannot be performed in parallel, while in the latter two that possibility exists (for instance, in RMS tasks that can be performed in the same workstation constitute an operation cluster which is considered as an input for the RMS optimisation problem). Moreover, once RMS selects the operation clusters and the machinery of each workstation, the throughput of the line (associated to the cycle time) can be obtained by installing multiple parallel machines. Consequently, the mathematical models and solution procedures greatly vary between the RMS and the line balancing literature.

Finally, the C-R-MMALBP generalises some problems previously considered in the literature, like the SALBP-1 and the cost-oriented assembly line balancing problem (Amen 2006; Roshani et al. 2012). An equivalent C-R-MMALBP instance of a cost-oriented assembly line balancing instance can be obtained as follows: (a) order the ability levels according to non-decreasing costs; (b) associate a resource to each ability level; (c) set the cost of each resource to the difference between the costs of the said ability level and its previous level; and (d) each task that requires a specific ability level now requires the resource of its ability level and all the previous ones.

3. A hybrid heuristic for the C-R-MMALBP

This section describes the enhanced multi-Hoffmann heuristic (Sternatz 2014) and how the heuristic is hybridised with an Estimation of Distributions Algorithm (Larrañaga and Lozano 2001) in order to solve the C-R-MMALBP.

The Hoffmann heuristic, first proposed in Hoffmann (1963), corresponds to a truncated enumeration scheme. The original method has been modified and enhanced in order to take the particularities of the problem into account (Fleszar and Hindi 2003; Morrison, Sewell, and Jacobson 2014; Sternatz 2014; Sternatz 2015).

This heuristic was chosen due to its results in the SALBP benchmark instances, see Pape (2015) for a recent comparative study, its low run time requirements, and its adaptability to other line balancing problems, see Sternatz (2014).

In order to detail the method, the description and this section are divided into three parts: Section 3.1 describes the original proposal as well as several modifications suggested in the literature, Section 3.2 describes the changes introduced to adapt the heuristic for its use in the resolution of the C-R-MMALBP, finally, Section 3.3 describes the hybridisation of the enhanced Hoffmann heuristic with an estimation of distribution algorithms (EDA), see Larrañaga and Lozano (2001). The proposed hybrid constitutes the major algorithmic contribution of the paper as it provides the enhanced Hoffmann heuristic with an adaptive self-adjusting method to ascertain its behaviour.

3.1 *Description of the Hoffmann heuristic*

The Hoffmann heuristic starts from an empty solution with no stations, generates all feasible station assignments for the first station, and selects the best grouping according to its idle time (smaller idle times are preferred). This process is repeated for each of the following stations, considering only unassigned tasks, until a complete solution is found. Hence, the method can be seen as a truncated branch-and-bound in which the best *greedy* station assignment is determined by enumeration.

As the problem is reversible, a simple improvement over the previous method consists in applying the same solution procedure to the reverse instance (an instance in which the direction of each precedence relation is reversed, and thus the role of the set of predecessors and successors is exchanged), since the solutions of the Hoffmann heuristic may differ. The idea was extended in Fleszar and Hindi (2003) in order to obtain a bidirectional multi-step procedure. The bidirectional procedure constructs a predefined number of stations for the direct/reverse instance, and then constructs the remaining stations using the opposite one.

This bidirectional rule offers multiple alternative solutions, and partially palliates the myopic selection of the original heuristic by increasing the number of considered alternatives. Nonetheless, candidate assignments are still chosen according to the idle time of the assignment, and thus it may lead to poor decisions in which other features of the problem are ignored.

This issue is considered in Morrison, Sewell, and Jacobson (2014) and Sternatz (2014). Based on the high quality of the heuristic, a multi-start version is devised. In order to select alternative assignments in each iteration, the assignment is chosen according to a weighted combination of four factors with weights that change in each execution. These factors are: (1) sum of operation times of the unassigned tasks; (2) positional weight of the unassigned tasks, that is, the sum of the operation time of the task and of its successors; (3) the sum of the immediate successors of the remaining unassigned tasks; and (4) the number of unassigned tasks. Furthermore, a limit on the number of generated assignments for any given station is imposed in order to avoid enumerating all of the assignments. Among the generated assignments, the candidate that maximises the weighted function is selected. While the method may still suffer from a myopic behaviour, the heuristic is run with different weights, thus increasing its ability to provide the optimal solution.

Finally, the Hoffmann heuristic can be further enhanced by including different weights and considering lower and upper bounds on the task assignments. The later speeds up the search by avoiding the exploration of solutions that cannot improve the incumbent. Furthermore, these bounds avoid the selection of suboptimal partial solutions that may locally optimise the weighted function but cannot lead to improved solutions.

3.2 *Modified Hoffmann heuristic for the C-R-MMALBP*

We proceed to describe the main features of the modified Hoffmann heuristic for the C-R-MMALBP. The features are divided into three topics: (1) how to use the reverse instance within the resolution of the problem; (2) how to avoid the enumeration of all possible assignments; and (3) how to select an assignment among the previously generated ones.

Use of the direct and the reverse instances within the resolution of the problem: The heuristic uses the bidirectional approach from Fleszar and Hindi (2003). Consequently, we construct a given number of stations in one direction, and then we reverse the precedence graph and find an assignment for the remaining tasks.

Limits on the enumeration of assignments: In order to avoid the enumeration of all of the candidate assignments for each workstation, the enumeration is stopped when each feasible assignment has been enumerated or a maximum $a^{\max}$ assignments has been constructed.

The parameter $a^{\max}$, see Section 4, not only controls the running time of the algorithm, but also introduces a measure of skewness in the search, since the order of the tasks in the instance influences the order in which the assignments are constructed. Consequently, before each execution of the Hoffmann heuristic, a random order of tasks is generated in order to avoid any skewness caused by a specific ordering of tasks.

Evaluation of alternative assignments: The last decision should be based on an understanding of the special characteristics of the specific line balancing problem, and what makes a feasible assignment likely to belong to a good solution.

The proposed method is based on the following three requirements: (i) the number of stations should be minimised; (ii) tasks that require the same resource should be assigned together; and (iii) tasks with multiple successors should be prioritised. Requirement (ii) tries to take into account the cost incurred due to the resources, while requirements (i) and (iii) are common to other line balancing problems. Notice that requirement (i) for the SALBP equates to avoiding idle time, while for the C-R-MMALBP, an alternative estimate based on multiple products is required.

In order to balance these three conflicting requirements, a function to compare different station assignments, see equation (19), is used. The function aggregates three different sources of information according to the subset of the tasks, $S \subseteq V$, that the current station assignment contains: (i) the task times, whose relative weight is denoted by w^{τ}, see equation (15); (ii) the use of resources, whose relative weight is denoted by w^{ρ}, see (17); and (iii) the precedence relations, whose relative weight is denoted by w^{ϕ}, see (18). These three weights are then aggregated using (19). Further details on each source of information are given below.

Equation (15) extends the weighting functions for the SALBP (in which only one product is taken into account) to multiple products. Consequently, the sum of operation times for each product is considered, and then the summation is weighted according to a parameter α_p associated to the importance of each product. Equation (18) follows the same approach (extending the SALBP) and considers the successors of the operations in the assignment. The formula considers the direction in which the stations are constructed, and thus it gives higher priority to operations with a large number of successors.

Finally, expression (17) adds the resource requirements to the evaluation method. The calculation first obtains an indicator for each resource using (16), and then produces a weighted aggregated value. Expression (16) accomplishes two purposes: first, it rewards high utilisation rates and penalises low utilisation rates for each resource; and second, if a resource is not used, its contribution is considered to be null. Therefore, the expression uses a parameter, R_r, to define a minimum desired utilisation rate if the resource is to be assigned, it compares the sum of operation times of the tasks that require the resource, and it returns either 0 if the resource is not used, a positive value if the utilisation rate surpasses R_r, or a negative value otherwise.

$$w^{\tau}(S) = \sum_{p \in P} \alpha_p \frac{\sum_{j \in S} t_{jp}}{K^{\tau}_p} \tag{15}$$

$$\omega(S,r) = \begin{cases} \frac{\sum_{j \in S: n_{jr} > 0} \sum_{p \in P} t_{jp}}{K^{\rho}_r} - R_r & \text{if } \sum_{j \in P} n_{jr} > 0 \\ 0 & \text{otherwise} \end{cases} \tag{16}$$

$$w^{\rho}(S) = \sum_{r \in R} \beta_r \cdot \omega(S,r) \tag{17}$$

$$w^{\phi}(S) = \begin{cases} \gamma \cdot \sum_{j \in S} \frac{|F^*_j|}{K^{\phi}} & \text{in a forward step} \\ \gamma \cdot \sum_{j \in S} \frac{|P^*_j|}{K^{\phi}} & \text{in a backward step} \end{cases} \tag{18}$$

$$w(S) = w^{\tau}(S) + w^{\rho}(S) + w^{\phi}(S) \tag{19}$$

A summation of the previous weights is obtained in function (19). This function uses multiple parameters to control its behavior. The set of weights α_p, β_r, together with weight γ control the influence of the operation times of each product, $p \in P$; each resource, $r \in R$; and precedence relations into the preference function, equation (19), respectively. An additional set of parameters, R_r, $r \in R$, is used to approximate the minimum desired usage of a resource that justifies the presence of the resource in a workstation.

The remaining parameters, that is, K^{τ}_p, $p \in P$, K^{ρ}_r, $r \in R$, and K^{ϕ}, are normalisation factors. While these normalisation factors are non-essential (they could be included within the weights), they are explicitly declared in order to highlight the importance of normalising to a 0–1 scale each source of information. This allows to limit the search for the α_p, β_r and γ weights to a predefined and similar range. These normalisation parameters are set to:

$$K^{\tau}_p = c_p \qquad \forall p \in P \tag{20}$$

$$K^{\rho}_r = \sum_{p \in P} c_p \qquad \forall r \in R \tag{21}$$

$$K^{\phi} = \max_{j \in V} \{|P^*_j|, |F^*_j|\} \tag{22}$$

Even when the normalisation parameters are fixed, the number of parameters required is still larger than in any of the previous implementations; hence, it is advisable to consider a more elaborate method to ascertain their value. This topic is covered in Section 3.3.

3.3 *Estimation of Distribution Algorithm for online parameter configuration*

Two alternative methods to ascertain weights in Hoffmann-based heuristics have been previously proposed. In Morrison, Sewell, and Jacobson (2014), a systematic method uses 100 different pre-selected combinations of weights, while in Sternatz (2014) the weights are randomly drawn from a uniform distribution within [0;1] in each application of the Hoffmann heuristic.

As the weighting function (19) proposed in Section 3.2 requires setting a larger number of parameters than the previous proposals, it is likely that the use of a more refined method to explore their space of values would provide better solutions. Consequently, the proposed method to solve the C-R-MMALBP applies a two-level resolution method: a high-level strategy to search for the best set of parameters used by a low-level method corresponding to the Hoffmann heuristic.

The resulting two-level method can be seen as: (1) a hyper-heuristic (Burke et al. 2013), in which the high-level strategy defines the heuristic used in the low-level (or more precisely configures the low-level heuristic); (2) a hybrid metaheuristic (Blum et al. 2011), in which the approaches of two different (meta)heuristics are combined within a single solution method; (3) an online automatic algorithm configuration (Birattari 2009), in which the high-level method configures the low-level algorithm; (4) a reactive search method in which the parameters of the modified Hoffmann heuristic are adapted during the optimisation process (Queiroz Dos Santos et al. 2014); or even as (5) a biased random-key genetic algorithm (Damm, Resende, and Ronconi 2016), in which an individual is decoded using the modified Hoffmann heuristic.

As the variety of interpretations hints, there are several alternatives to define a valid high-level strategy. Among them, an Estimation of Distribution Algorithm, EDA, Larrañaga and Lozano (2001) was selected, as EDA embodies the idea of an intelligent search over the parameters space of the lower level heuristic, which is the rationale behind the proposed method.

EDA is an evolutionary algorithm, see Larrañaga and Lozano (2001) or Wang et al. (2015) for an example of a hybrid application, in which the population is implicitly described using a probabilistic model. In each iteration, the model is sampled to generate a set of solutions. These solutions are evaluated, and the best performing ones are used to update the probabilistic model. This process is repeated until a stopping criterion is met.

For the case under study, a solution corresponds to the set of parameters that control the behaviour of the Hoffmann heuristic. Consequently, the probabilistic model should define the following components: (1) the weights in the evaluation function, that is, α_p, $p \in P$, β_r, $r \in R$, and γ; (2) the threshold level for each resource, R_r, $r \in R$; (3) the primary direction of the search, be it forward or reverse; and (4) the number of stations constructed using the primary direction before reverting to the secondary direction that is used to construct the remaining workstations.

Note that we do not consider the number of candidate assignments as a parameter to be optimised by EDA. Increasing the number of candidate assignments yields an improvement on the average solution quality at the expense of increasing the running times. Section 4 provides details on how this parameter was defined. Furthermore, the reordering of tasks is also left out of the EDA, since it would lead to a fundamentally different algorithm (the operation ordering rather than the configuration of parameters would constitute the main focus of the search method). Algorithm 1 depicts the overall structure of the method.

Algorithm 1. Structure of the proposed EDA algorithm

```
Input: popsize: population size; nsols: maximum number of evaluated solutions.
Output: best: best found solution.
 1: $n \leftarrow 0$
 2: $prob.model \leftarrow initialize()$
 3: while $n < nsols$ do
 4:     for $(i \leftarrow 1; i \leq popsize; i \leftarrow i+1, n \leftarrow n+1)$ do
 5:         $s[i] \leftarrow sample(prob.model)$
 6:         $eval[i] \leftarrow Hoffmann(s[i])$
 7:         $best \leftarrow updateBest(eval[i], s[i], best)$
 8:     end for
 9:     $prob.model \leftarrow update(prob.model,\ selectbest(s,\ eval, card^{\text{best}}))$
10: end while
```

Lines 1 and 2 correspond to the initialisation of the algorithm. Afterwards, the **while** loop is repeated until the desired number of solutions has been generated. In each iteration of the loop, *nsols* individuals are sampled from the probabilistic model, line 5; evaluated by solving the Hoffman heuristic with the parameters defined by the individual, line 6; and the best known solution is updated if needed, line 7. Subsequently, the probabilistic model is updated, line 9, using a subset of the best solutions, selected according to function $selectbest(s,\ eval)$.

While Algorithm 1 is agnostic to the probabilistic model, one is needed to define the inner workings of the EDA. In this work we use a simple probabilistic model without dependences. The probabilistic model stores a single real-valued parameter for each element required by the Hoffmann heuristic, which in turn takes a different role depending on the underlying distribution used to model the characteristics within the population. Parameters are classified into three categories:

- Evaluation weights and the threshold levels. A normal distribution truncated between 0 and 1 is used as the underlying probabilistic model for each of these elements. The parameter of the EDA is associated to the point estimation (the mean) of the distribution. Note that the distribution requires an additional parameter σ, which is related to the standard deviation of a normal distribution. σ controls the diversity among individuals and it is automatically updated as the search advances, see below.
- Primary direction of the Hoffmann heuristic. The primary direction may take two values (forward and reverse). Consequently, a Bernoulli distribution and its mean value correspond to the probability that the forward direction is the primary search direction.
- Number of workstations constructed using the primary direction. For any given individual of the population, the parameter is limited to take values between 0 and $|V|$. To represent the distribution, a normal distribution truncated between 0 and $|V|$ is used. As with the weights and threshold levels, σ' is required to fully describe the distribution, see generation of new solutions below.

During the initialisation step, see Algorithm 1 line 2, the point estimation of each distribution is set to half its range of values, that is 0.5 or $|V|/2$. Moreover, the additional parameters σ or σ' are also set to $\sigma = 0.5$, and $\sigma' = |V|/2$

Both parameters, σ and σ', control the variation among different individuals, given that they correspond to the standard deviation of their respective non-truncated normal distribution. The algorithm uses σ and σ' to control the focus of the search between diversification and intensification. Initially, the implementation fosters diversification, and thus σ and σ' ensure that all of the values within the range of feasible values are likely to be sampled.

When the probability model is updated, Line 9 of Algorithm 1, the values of σ, σ' are reduced, and the parameter is updated according to the best generated solutions in the previous step. The updating of σ and σ' is designed so as to decrease its value until a targeted minimum is reached in the last iteration. Let σ^+ and σ^- be the desired maximum (initial iteration) and minimum (final iteration) values for σ, and let $it^+ = \lceil \frac{nsols}{popsize} \rceil$ be the number of iterations of Algorithm 1. Then, σ and σ' are updated as in (23).

$$\sigma \leftarrow \sigma\ e^{\frac{(log(\sigma^-)-log(\sigma^+))}{it^+}} \tag{23}$$

After some preliminary tests, the constants are set to $\sigma^- = 0.025$ and $\sigma'^- = 1$ (note that $\sigma^+ = 0.5$ and $\sigma'^- = |V|$). These values were chosen in order to reflect minimal variability (similar to reaching convergence within an evolutionary algorithm).

The point estimation, mean, of the parameters is updated as follows. First, a subset of the best individuals (smallest cost) in the current iteration is selected, function $selectbest(s, eval, \text{card}^{\text{best}})$ in line 9 of Algorithm 1. The cardinality, $\text{card}^{\text{best}}$, of the subset is a parameter of the EDA, see Section 4. Second, for each parameter μ, the average value within the individuals of the subset, $\hat{\mu}$, is calculated. Third, the parameter is updated according to a learning rate, λ, which is another parameter of the algorithm, see Section 4. Consequently, the point estimation μ becomes $\mu + \lambda(\mu - \hat{\mu})$. After updating, if the point estimate falls below or rises over the limits for the distribution, the estimate is rectified so as to stay within the limits.

Finally, new individuals, see Algorithm 1 line 5, are constructed by generating a random number that follows the given distribution. The Hoffmann heuristic, Algorithm 1 line 6, uses the parameters that define the individuals. Note that the number of workstations is rounded to the nearest integer, and the primary direction of the search is set to forward if the Bernoulli sample takes value 1, and to backward if it takes value 0.

4. Results

4.1 *Implementation details and description of the instances*

The Hoffmann heuristic described in Section 3 was coded in C, compiled using gcc 4.4.7 and run on a 32-core 2.0 GHz Intel Xeon with 128-gigabytes RAM, running the Linux operating system. The 32 cores are not used in the code and were only used to perform the resolution of 32 different instances simultaneously. Therefore, the code can be considered to have been executed in a single processor machine. Note that the application is not memory intensive (the memory consumption in each execution stood below 100 Mb).

In order to investigate the performance of the proposed algorithm when compared with previous approaches, a computational experiment with instances derived from the literature was conducted. The description of these instances is provided below and the results of the experiments are reported in Section 4.2. A separate section is devoted to the study of the real case found in the apparel industry, see Section 4.3. Additionally, and in order to compare the solutions found with a bound on the optimal solution, the lower bound described in Appendix 1 was also implemented. This bound was selected as it is the state-of-the-art lower bound for the SALBP (see Pereira 2015). The implementation uses the IBM ILOG CPLEX library version 12.6.2 to solve both the mathematical formulations of the relaxation and the pricing subproblems.

An instance set derived from instances found in the literature was constructed in order to consider the effect of the additional features of the C-R-MMALBP. The set proposed in Otto, Otto, and Scholl (2013) was chosen because it contains harder instances than the classical instance set, see further discussion in Otto, Otto, and Scholl (2013). Moreover, the set follows a structured approach that enables the analysis of the effect of the different characteristics for the solution approach. The SALBP set is divided according to four characteristics, see Otto, Otto, and Scholl (2013) for further details:

Number of tasks in the instance. Following the suggestions given in Otto, Otto, and Scholl (2013), as well as the results reported in Pereira (2015), the *large* instance set was selected. Consequently, those instances with 100 tasks are considered, $|V| = 100$.

Order strength, OS, of the instance (the ratio between the number and the total possible number of predecessors). Instances with *low*, 0.2, and *high*, 0.6, OS are selected. Instances with *very high*, $OS = 0.9$, were left out since they were over constrained and, thus, easier to solve.

Structure of the precedence graph. Instances with *block* (tasks with multiple predecessors), *chain* (featuring long series of tasks with one predecessor and successor) and *mixed* (combining both structures) are considered.

The operation time of the tasks. Three types of operation times are considered: *bottom* (task time follows a normal distribution with mean equal to 0.1 times the cycle time), *middle* (normal distribution with mean equal to 0.5 times the cycle time) and *bimodal* (bimodal combining the previous distributions).

The instance set described in Otto, Otto, and Scholl (2013) contains 525 instances per instance size, which are divided in subgroups of 25 instances with common characteristics, OS, structure, and operation times. For each combination of characteristics considered in this study, five instances were selected from each subgroup, summing up a total of 90 instances.

In addition to these characteristics, the instances include four factors relevant to the C-R-MMALBP:

Number of products. The cardinality of the set of products P introduces cumulative constraints on the problem and has a strong effect on the number of feasible station assignments. Instances with two levels are considered: *low*, $|P| = 2$, and *high*, $|P| = 5$.

Number of resources. The characteristic controls the cardinality of the set of resources, R, that any task may require. For simplicity and due to their more frequent use, resources are deemed binary, and thus if a station is equipped with a given resource, it can perform any task that requires the resource. Two different levels are considered: *low*, $|R| = 1$, and *high*, $|R| = 5$, number of resources.

Resource use. The parameter considers the average number of tasks that require a resource. Three levels are considered: *low*, *medium* and *high*, indicating that approximately 10%, 25% or 50% of the tasks require the resource.

Cost ratio between resources and stations. This parameter controls the relative cost of a resource when compared to the cost of a workstation. Three levels are considered: (1) *low* cost ratio instances, in which resources are less expensive than workstations (equipping a workstation with any given resource costs 25% of the cost of a new workstation); (2) *medium* cost ratio instances, in which the costs are identical, and (3) *high* cost ratio instances, in which the cost of the equipment doubles the cost of a workstation.

Each of the selected SALBP instances is then used to construct an instance for each combination of the four characteristics relevant to the C-R-MMALBP. Therefore, 36 instances are derived from each SALBP instance for a total of 3, 240 instances. In order to generate the operation times of the additional products, the first product inherits the operation times from the SALBP instance. The remaining operation times are constructed as follows: for each task and product, randomly draw an operation time among the operation times present in the original instance. Consequently, the cycle time of each product is identical. Note that, as discussed in Otto, Otto, and Scholl (2013), considering different cycle times is not relevant, since the cycle time and the operation times can be scaled to any given scale.

4.2 *Experimental results*

In order to verify the quality of the modified Hoffmann heuristic, Section 3.2, as well as the quality of the hybrid method that combines the Hoffmann heuristic with the EDA, Section 3.3, three different computational experiments were performed.

The first experiment tries to: (1) ascertain the best set of parameters for the EDA algorithm; (2) select the maximum number of alternative assignments considered by the Hoffmann heuristic; and (3) investigate the usefulness of using different task orderings within the Hoffmann heuristic.

For this experiment, a subset of 10 (out of the 3, 240) instances with varying characteristics (each belonging to a different set of characteristics) was randomly selected and solved using the EDA with different combinations of parameter settings: number of alternative assignments, $a^{\max} = \{500; 1000; 2500; 5000\}$; population size per iteration, $popsize = \{50; 100; 250; 500\}$; cardinality of the subset used to update the probabilistic model, $\text{card}^{\text{best}} = \{1; 3; 5\}$; and learning rate, $\lambda = \{0.01; 0.025; 0.05\}$. Additionally, three different alternatives to determine how to order the tasks were considered: (1) the Hoffmann heuristic uses a single task ordering (the order given in the instance file); (2) each time the Hoffmann heuristic is executed, a task ordering is randomly generated; and (3) each time the Hoffmann heuristic is executed, three different task orderings are generated, then each ordering is used by the Hoffmann heuristic to obtain a solution and the best solution among the three is reported back. For each combination of parameters and instances, the termination condition of the EDA was set to 50, 000 individual Hoffmann executions.

The effect of the parameters was tested in a fully factorial experimental design in which the instances are considered as fixed effects, and the best solution found per combination of parameters was considered as the response variable. An ANOVA analysis showed that all of the parameters had a significant impact on the results of the algorithm, and a post-hoc analysis between groups led us to select the following combination of parameters and options: generate one random task ordering each time the Hoffmann heuristic is called, set $popsize = 50$, $\text{card}^{\text{best}} = 5$, $\lambda = 0.025$ and $a^{\max} = 2500$ (the difference between 2500 and 5000 assignments was not statistically significant but increased the running time).

In light of the results of the first experiment, only the selected EDA configuration was considered for the third experiment.

The second experiment compares the proposed modified Hoffmann heuristic, Section 3.2, and the state-of-the-art implementation reported in Sternatz (2014). Both modified Hoffmann heuristics are embedded within a multi-start, M-S, method proposed in Sternatz (2014), which executes the modified Hoffmann heuristic repeatedly with different weights. Both implementations differ in the following details:

- The enumeration of alternative assignments per station. In Sternatz (2014) the maximum allotted time to generate assignments per station is limited, whereas in the proposed method a limit on the maximum number of considered assignments per station is imposed.
- For a given set of parameters of the weighting function, in Sternatz (2014) the method evaluates both the forward and the reverse directions as well as each possible number of workstations for the primary direction. In the proposed method, the direction and the number of forward or backwards steps are randomly chosen.

All of the 3240 instances were solved using both alternative M-S methods imposing a limit on the number of solutions provided by the Hoffmann heuristic, which was set to 500,000. For the proposed implementation, as described in Section 3.2, the maximum number of alternative assignments was set to $a^{\max} = 2500$, since it was the best performing value in the first experiment. For the implementation described in Sternatz (2014), the time limit was set to 5 s, as originally proposed. The results are compared in two dimensions: solution quality and running time requirements.

Regarding the solution quality, the implementation described in this paper provides an strictly better solution in 886 of the 3240 instances (27.3%), while in 583 instances (18%) the implementation proposed in Sternatz (2014) outperforms our implementation. In the remaining instances, 54.7% of the instance set, both implementations provide the same best solution. Consequently, both implementation are comparable in terms of solution quality, even if our implementation slightly outperforms the implementation described in Sternatz (2014).

As for the total running time, the results of the two implementations differ significantly. It should be expected an increase in the running time if the method is able to evaluate more than 2500 alternative assignments within the 5 s time limit. Conversely, the results show that while the median running time per instance of both implementations is similar (around 3250 seconds), the average time is 20 times larger when the 5 s time limit is used. Notice that evaluating station assignments is a combinatorial operation and thus this result is caused by extreme instances in which the 5 s time limit imposes a weak limit on the enumeration of alternative station assignments. Specifically, the maximum running time for the time-based limit method is much larger than for its counterpart. For the said instance, over 225,000 s are required to reach the 500,000 independent solutions limit. While this quantity is large, it equates to slightly less than half a second per Hoffmann execution, hence the time limit would still be applicable if a smaller number of executions was imposed, as in Sternatz (2014).

Given the above results, we conclude that the proposed implementation is to be preferred, mainly due to smaller running time requirements.

The third experiment compares the performance of EDA with the multi-start Hoffmann heuristic, M-S, described in Section 3.2 and investigates the robustness of EDA to converge to similar figures both in terms of solution quality and parameter values of the probabilistic model. All of the 3240 instances were solved using the EDA with a limit on the number

Table 1. Results from the different algorithms when instances are grouped according to graph characteristics. Results for the multi-start with 50,000 and 500,000 evaluations and EDA with 50,000 solutions are reported. For each grouping of instances according to precedence relations, order strength and distribution of operation times and each of the algorithms, the number of best found solutions among the 180 instances of each group, columns # best, and the average gap to the best known solution, columns gap, are provided. For EDA, the gap for the best and the average solution among all 10 independent executions, columns av.sol.gap and best sol.gap, as well as the optimality gap with respect to the lower bound, column Opt.gap, are also reported.

	Graph		M-S(50, 000)		M-S(500, 000)		EDA			
Structure	OS	Distribution	# best	gap	# best	gap	# best	av.sol.gap	best sol.gap	Opt.gap
block	*low*	*bottom*	57	3.27	81	2.36	176	0.99	0.07	8.91
		middle	54	0.92	100	0.47	156	0.40	0.11	3.25
		bimodal	45	2.98	68	1.97	180	0.59	0.00	7.42
	high	*bottom*	71	2.28	86	1.46	178	0.62	0.02	5.60
		middle	68	0.64	124	0.25	163	0.31	0.08	2.40
		bimodal	56	2.03	90	1.07	171	0.65	0.08	6.05
chain	*low*	*bottom*	60	3.20	77	1.97	179	0.71	0.02	9.10
		middle	64	0.81	118	0.36	166	0.30	0.06	3.02
		bimodal	37	3.06	57	1.98	177	0.63	0.03	8.48
	high	*bottom*	74	1.68	120	0.85	169	0.66	0.11	7.91
		middle	90	0.49	137	0.17	166	0.26	0.06	2.08
		bimodal	43	2.07	78	1.16	168	0.63	0.10	8.07
mixed	*low*	*bottom*	60	3.35	80	2.11	177	0.81	0.04	8.59
		middle	85	0.65	128	0.30	159	0.27	0.07	1.85
		bimodal	40	3.25	63	2.02	178	0.69	0.02	7.35
	high	*bottom*	62	2.35	85	1.41	177	0.71	0.05	6.46
		middle	90	0.59	130	0.25	161	0.32	0.09	1.98
		bimodal	49	2.22	81	1.26	169	0.62	0.08	7.88

of Hoffmann solutions set to 50,000, whereas the M-S methods were allowed 50,000 and 500,000 Hoffmann solutions. We will denote the M-S heuristics by M-S(50,000) and by M-S(500,000) according to the number of evaluated solutions. The EDA algorithm was run with ten random seeds in order to investigate the robustness of the method with different random seeds, as well as its convergence in different initial conditions.

Table 1 provides results of instances grouped according to their SALBP characteristics. Each group contains 180 instances, and the results of the M-S(50,000), the M-S(500,000) and the EDA are provided. For each M-S iteration limit, the number of instances (out of 180 instances) in which the algorithm reports the best known solution, columns # best, and the average gap, columns gap, are provided. The gap is measured as $\frac{UB-UB^b}{UB}$, in which UB corresponds to the best solution found by the algorithm under comparison and UB^b corresponds to the best solution found among all of the procedures. For the EDA, the number of best known solutions, column # best, as well as the gap for the best and for the average solution found among the 10 independent executions, columns best sol.gap and av.sol.gap, respectively, are reported. In addition to the above metrics, the average optimality gap between the best solution, UB, provided by the EDA and the lower bound, LB, provided by a Dantzig–Wolfe based lower bound, see Appendix 1, is also reported, columns Opt.gap. The average optimality gap is measured as $\frac{UB-LB}{UB}$.

Table 2 provides the same results for the instances grouped according to their C-R-MMALBP characteristics. In this case, each group contains 90 instances.

The results show that the EDA outperforms the M-S approaches, as the average solution gaps are smaller for the EDA, column av.sol.gap, than for the multi-start heuristic, column gap M-S(50,000)). Consequently, we can conclude that the EDA improves upon the random exploration of the parameter space for the modified Hoffmann heuristic. Moreover, if the multi-start approach is allowed a larger number of evaluations, the EDA is still the preferred method, since the gap of the average EDA solution is consistently smaller than the gap of M-S(500,000), even if EDA is outperformed for some specific instance groupings. When the difference between the average and the best solution gap of EDA is analysed, we can conclude that independent runs of the algorithm provide solutions of similar quality, that is, small difference between both gaps, even when some differences among replications exist. Consequently, the algorithm does not always converge to a unique local optima and may benefit from multiple independent runs showing that it is able to explore different areas of the search space, reaching a good quality solution in each case.

Note that while there are some instances in which the solution found by the M-S heuristic outperforms the EDA method (EDA fails to provide the best known solution for every instance), EDA provides better solutions for most instance groupings.

Table 2. Results from the different algorithms when instances are grouped according to the specific C-R-MMALBP characteristics. Results for the multi-start with 50, 000 and 500, 000 evaluations and EDA with 50, 000 solutions are reported. For each grouping of instances according to number of products, number of resources, level of resource use and resource cost, the number of best found solutions among the 90 instances of each group, columns # best, and the average gap to the best known solution, columns gap, are provided. For EDA, the gap for the best and the average solution among all 10 independent executions, columns av.sol.gap and best sol.gap, as well as the optimality gap with respect to the lower bound, column Opt.gap, are also reported.

#	#	resource	resource	M-S(50, 000)		M-S(500,000)		EDA			
products	resources	Use	Cost	# best	gap	# best	gap	# best	av.sol.gap	best sol.gap	Opt.gap
low	*low*	*low*	*low*	64	0.34	83	0.01	82	0.34	0.14	3.12
			medium	76	0.26	87	0.09	86	0.25	0.06	2.65
			high	74	0.27	85	0.07	84	0.21	0.10	3.26
		medium	*low*	51	0.51	84	0.08	74	0.48	0.24	3.74
			medium	63	0.73	83	0.18	84	0.47	0.10	4.11
			high	56	0.80	81	0.16	82	0.42	0.16	4.92
		high	low	44	0.55	82	0.08	73	0.41	0.14	4.09
			medium	54	0.92	72	0.51	83	0.45	0.10	4.34
			high	38	1.34	67	0.61	83	0.54	0.08	5.12
	high	*low*	*low*	5	1.83	24	1.01	85	0.61	0.03	5.23
			medium	8	3.41	31	1.78	88	0.94	0.05	6.72
			high	7	3.91	21	2.41	88	0.98	0.03	7.77
		medium	*low*	2	2.29	8	1.40	88	0.63	0.03	6.59
			medium	2	3.66	5	2.45	90	0.75	0.00	8.79
			high	2	4.02	8	2.47	87	0.93	0.03	10.29
		high	*low*	4	1.93	16	1.06	90	0.46	0.00	5.95
			medium	0	2.75	10	1.79	87	0.65	0.03	7.82
			high	2	3.03	3	2.04	89	0.66	0.01	8.49
high	*low*	*low*	*low*	65	0.49	76	0.20	86	0.34	0.10	4.19
			medium	66	0.99	81	0.44	87	0.58	0.10	3.58
			high	62	0.88	80	0.41	88	0.32	0.04	4.00
		medium	*low*	46	0.68	67	0.33	84	0.38	0.05	4.53
			medium	49	1.40	64	0.87	87	0.53	0.08	4.86
			high	34	2.11	61	1.08	83	0.58	0.10	5.43
		high	*low*	36	0.78	60	0.34	82	0.43	0.06	4.60
			medium	39	1.72	68	0.72	87	0.45	0.03	5.47
			high	21	2.48	47	1.33	84	0.52	0.07	5.97
	high	*low*	*low*	21	1.72	36	1.12	88	0.52	0.02	5.65
			medium	23	3.64	37	2.43	84	0.81	0.06	7.54
			high	19	4.84	30	3.31	89	0.78	0.01	7.84
		medium	*low*	12	1.97	30	1.16	88	0.55	0.01	6.52
			medium	12	3.60	20	2.51	88	0.80	0.01	8.41
			high	12	5.02	20	3.44	87	0.84	0.01	9.11
		high	*low*	14	1.50	29	0.87	83	0.49	0.03	5.84
			medium	12	2.35	27	1.49	86	0.50	0.02	7.40
			high	10	2.96	20	1.98	86	0.72	0.03	7.99

An analysis of the characteristics of the instances in which M-S(500, 000) outperforms the average results of EDA provides further insight into the advantages of the EDA over randomly sampling the value of the parameters.

Table 1 shows that M-S(500,000) only outperforms the average EDA results in terms of average solution gap for three task groupings with *middle* distribution of operation times. This distribution generates instances in which few tasks may share workstations. Consequently, the additional randomness and the larger number of alternatives considered by the MS(500,000) over EDA, which runs fewer executions, improves the chances that different tasks are packed together, thus improving the chances of finding a very good solution. Note that this result does not apply when the number of Hoffmann executions allotted to the EDA method is increased. Also note that the best solution of EDA after 10 independent runs can be considered as an EDA method with a limit of 500,000 Hoffmann executions that uses 10 restarts as a diversification policy, and thus it is comparable to M-S(500,000) in terms of allotted computing resources, see running time comparisons below. In such a case, EDA outperforms M-S for each grouping of instances.

Table 3. Average and maximum running times for the instances grouped according to each of the characteristics of the instance.

Characteristic	type	Av. t(s)	Max. t(s)
Structure	*block*	378	1648
	chain	257	1560
	mixed	350	1745
OS	*low*	469	1745
	high	187	1099
Distribution	*bottom*	618	1745
	middle	39	104
	bimodal	328	1540
# products	*low*	268	999
	high	388	1745
# resources	*low*	241	1059
	high	415	1745
resource usage	*low*	308	1533
	medium	336	1591
	high	341	1745
resource cost	*low*	324	1745
	medium	331	1736
	high	330	1648

Table 2 reports 11 groups in which M-S(500,000) outperforms the EDA. Seven of these groups feature *low* number of products and *low* number of resources, while the remaining groups correspond to *low* number of resources. Notice that the number of products and resources determines the number of parameters that the modified Hoffmann heuristic uses. Hence orth, a low number of products and resources means a smaller search space that the M-S(500,000) needs to explore through random examination. Additionally, there are four groups in which M-S(500,000) outperforms, by a small margin, the aggregated results of 10 independent EDA executions, and all of them correspond to cases in which the number of products and resources are low, further supporting the previous conclusion.

Considering that the multi-start method was shown to be very effective for the SALBP, see Sternatz (2014), we can conclude that when few parameters define the behavior of the modified Hoffmann heuristic, there is no need for an elaborate method to obtain their values. Note that the results still show that if the number of evaluations is limited, or a small running time limit is imposed, the use of the EDA to explore the parameter setting is still advantageous.

Also note that there is a strong correlation between the optimality gap obtained by the EDA and the relative difference between the EDA and the M-S(500,000). For those instance groupings in which the EDA obtains results similar to those of the M-S(500,000), the optimality gap is small, while it increases for those instances in which the EDA outperforms the M-S heuristics. We conjecture that the proposed lower bounding method does not fully reflect the particularities of the C-R-MMALBP and, thus, the quality of the lower bound decreases as the differences between the SALBP and the C-R-MMALBP increase. Note that both the M-S heuristics and the lower bound provide very good solutions for the SALBP (Pereira 2015; Sternatz 2014), and both fail to capture the additional complexities of the C-R-MMALBP.

These results highlight the differences between the SALBP and the C-R-MMALBP, and clearly support the use of a solution method that takes into account the particularities of the C-R-MMALBP during the resolution procedure.

Table 3 reports the running times of the EDA implementation grouped according to the different characteristics of the instances. We do not report the running time of the multi-start algorithms as the time required by the sampling and the updating steps of the EDA, lines 5, 7 and 9 of Algorithm 1, is less than 2 s in every execution. Consequently, the running time of EDA is, for all practical purposes, identical to the time required to perform the 50,000 independent executions of the Hoffmann heuristic and, thus, equal to the running time of the M-S(50,000) heuristic and ten times smaller than the time required for the M-S(500,000) heuristic.

An analysis of the results show that the maximum running time of the algorithm stands below half an hour for each of the tested instances and slightly over 5 min in most cases. It is important to note that the distribution of operation times constitutes the characteristic with the most significant effect on the running times, since it provides an indication of the number of different partial assignments that can be constructed for any given workstation.

This experiment was also used to investigate the convergence of EDA by considering the differences among final point estimates provided by independent executions. For each of the 3240 instances and each parameter estimated by EDA, the coefficient of variation, $\hat{c_v} = \frac{s}{\bar{x}}$, and the alternative quartile coefficient of dispersion, $\frac{Q_3 - Q_1}{Q_3 + Q_1}$, were obtained considering the ten independent runs as a sample of its distribution.

Both statistics provide an indication of the dispersion of the underlaying distribution. Consequently, if EDA converges to similar estimates, low coefficients are expected, whereas the opposite holds if the algorithm converges to very different point estimates.

As the coefficient of variation is limited to values in ratio scale, an average among all point estimates that are known to lie between 0 and 1, the weights, threshold levels and search direction, were considered. The resulting average among coefficients of variation was $\hat{c_v} = 0.141$, while the average quartile coefficient of dispersion was 0.083. Therefore, the variability among independent runs is small, reinforcing the previous conclusion that EDA converges to similar solutions but does not stagnate the search to a limited area of the search space, since differences exist among independent runs.

4.3 *Results for the industrial example*

The industrial example described in Section 2.2, see also Appendix 2, was solved using the integer linear programming formulation, (1)–(8), Section 2.3, and EDA, Section 3. Moreover, the lower bound provided by the Dantzig–Wolfe reformulation, Appendix 1, was calculated.

The results of our tests show that the instance can be regarded as ‘easy’ from a computational point of view. First, the integer linear programming formulation provides the optimal solution within seconds and the EDA implementation is able to yield multiple optimal solutions within similar computing times. Moreover, the industrial solution is also optimal and when the first optimal solution provided by EDA is compared to the current line balance, only slight differences among the station assignments are found.

Nonetheless, from a practical standpoint there is an important difference between the use of EDA and the current practice. While the current solution was obtained by a decision maker that took a long time to define an initial solution and then improved it through a trial-and-error process, our method provides multiple solutions within negligible times. The advantages of automatic, or computer-aided, planning should become more evident as more complex problems are considered. In a more complex environment, the planner may fail to tackle some particularities of the problem in hand and ignore them as it may be the case in this example. Consequently, the decision-maker would greatly benefit by the suggestion of fast-to-achieve, near-optimal solutions, which would also enable considering a more detailed formulation of the problem.

The experiments also support known results for the SALBP as well as our previous results, see Section 4.2. Specifically, (1) C-R-MMALBP instances with few products and few resources are not significantly different from SALBP instances, corroborating our previous results, and (2) *small* (few tasks) SALBP instances are easily solved to optimality. In fact, it is possible to optimally solve 1000 tasks instances with some exceptions, (see Morrison, Sewell, and Jacobson 2014).

When the C-R-MMALBP solution is compared to the SALBP solution for each product, some differences appear. While the number of optimal stations for the C-R-MMALBP was 11 and 8 of these stations were manned by skilled personnel, the optimal solution for the SALBP required 10 stations for each product. Moreover, all optimal assignments with 10 stations were generated in order to obtain the minimal number of skilled personnel required to equip the machines. For the male model, only 5 stations are manned by skilled workers, while for the female sweater, 8 of these 10 stations need to be manned by skilled workers. Consequently, the multi-model solution involves the use of an additional unskilled worker to avoid reconfiguring the line between batches of different products, but the need of skilled personnel is not increased. An optimal solution for both SALBP instances, minimal number of stations breaking ties by minimising resource requirements, are included in Appendix 2.

When the optimal solution is compared to the lower bound provided by the Dantzig–Wolfe reformulation, a 5% gap between these two values is found. The difference comes from the impossibility of assigning all of the operations that require the resource to specific workstations if multiple products and precedence constraints are to be fulfilled, further highlighting the differences between the SALBP and the C-R-MMALBP. Note that the optimality gap of the considered lower bound is usually below 1% for SALBP instances.

To conclude, we would like to point out that while the proposed methods were not able to outperform the current practice in terms of solution quality, the example highlights the differences between the SALBP and the C-R-MMALBP, and the trade-offs involved between manual and automatic line balancing and between single- and multi-product line balancing.

5. Conclusions

In this paper, we have considered a general assembly line balancing problem that models a real-life situation found in the apparel industry. This novel problem is denoted by C-R-MMALBP, the cost and resource multi-mode assembly line balancing problem, and it extends the classical formulations of the simple assembly line balancing problem, SALBP, by taking into account a cost objective function, the production of multiple products in a single line, and the use of special resources (machinery) that need to be assigned to workstations in order to perform the tasks.

Specifically, we provide two integer programming formulations for the problem, as well as adaptations of the SALBP state-of-the-art lower bound, based on the Dantzig–Wolfe reformulation of the problem, and the SALBP state-of-the-art upper bound, the Hoffmann heuristic, to the C-R-MMALBP. Additionally, the Hoffmann heuristic is combined with an Estimation of Distribution Algorithm, EDA, to better solve the problem in hand. Computational experiments with the proposed methods lead us to the following conclusions:

- The differences between the C-R-MMALBP and the SALBP formulations become apparent by the different behaviour of both the lower and the upper bounds on the various instance characteristics of the instances. Specifically, the quality of the SALBP-based methods deteriorates as the number of differences (additional products or resources) are included in the problem.
- The EDA-based method is able to outperform the previously proposed approaches to search for the best set of parameters in the Hoffmann heuristic, see Morrison, Sewell, and Jacobson (2014), Sternatz (2014, 2015). Specifically, the EDA-based method reaches higher quality solutions even if a smaller number of trial solutions is allowed.
- The advantage of the EDA method over the previous approaches is accentuated when the number of products and resources in the instance increases. This behaviour can be explained by the ability of the proposed method to better explore alternative parameter settings than the previously used random search.
- The behaviour of the EDA method is found to be robust among replications. Specifically, the variation among independent runs of the algorithm is small both in terms of best-found solution and in terms of the dispersion of the parameter values offered by EDA.

In addition to the computational experiment, the applicability of the problem in practical settings is considered through an example stemming from the apparel industry. In this case, the following conclusions are drawn:

- The formulation takes into account heterogeneity among workers and equipments. Heterogeneity among workstations resembles many assembly lines in which cost-oriented objectives need to be considered. Specifically, the industrial example shows that worker differences can be represented using different resources.
- The results show that the industrial instance can be regarded as ‘easy’ from a computational point of view, since the integer linear programming formulation, the EDA implementation and the industrial solution all provide optimal solutions.

  Consequently, the main advantage of the proposed method over the current practice lies in the ability of the automatic, or computer-aided, planning method to obtain multiple solutions within negligible times, whereas the decision maker needs a long time to define an initial solution and then improves it through a trial-and-error process.
- An additional advantage of the proposed framework is the ability to handle additional information within the decision process. Consequently, the application of a computer-aided system would encourage the planner to collect additional information for its use within more complex models, which resemble real-life situations more closely.
- To sum up, the example highlights the differences between the C-R-MMALBP and other line balancing formulations and provides some discussion on the trade-offs involved in alternative line balancing methods and formulations.

Based on these results, we are led to conclude that the need for an intelligent search of the parameter space is likely to hold for other line balancing problems, and specifically for line balancing problems with additional features like those found in the industry. For these problems, the proposed EDA-Hoffmann heuristic offers a starting point for the development of efficient algorithms, and the C-R-MMALBP provides a basic formulation to build additional features for different cost-oriented assembly lines. The method provides an explicit mechanism to handle heterogeneity and multiple resources in assembly line balancing problems and, after suitable modifications, should be applicable to similar problems with cost-oriented objectives and/or side constraints.

As a final remark, we would like to stress the relevance of integrating line balancing issues within other tactical and strategic decisions. The pertinence of such issues has already been raised by different authors (see Boysen, Fliedner, and Scholl 2009; Tonelli et al. 2013, among others). The resulting problems should lead to line balancing formulations that depart from efficiency oriented objectives to profit or cost-oriented objectives (as discussed in Hazir, Delorme, and Dolgui (2015)) that account for resource and personnel decisions. Additional features, like multi-objective considerations, uncertainty issues or robustness considerations are of special interest and worthy topics of future research.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by Fondo Nacional de Desarrollo Científico y Tecnológico of the Ministry of Education of Chile [grant number 1150306] titled ‘Heterogeneous assembly line balancing problems with process selection features’. This research was also partially supported by the supercomputing infrastructure of the NLHPC (ECM-02) in which some preliminary computational tests were conducted.

References

Amen, M. 2006. “Cost-oriented Assembly Line Balancing: Model Formulations, Solution Difficulty, Upper and Lower Bounds.” *European Journal of Operational Research* 168 (3): 747–770.

Battaïa, O., X. Delorme, A. Dolgui, J. Hagemann, A. Horlemann, S. Kovalev, and S. Malyutin. 2015. “Workforce Minimization for a Mixed-model Assembly Line in the Automotive Industry.” *International Journal of Production Economics* 170: 489–500.

Battaïa, O., A. Dolgui, N. Guschinsky, and G. Levin. 2014. “Combinatorial Techniques to Optimally Customize an Automated Production Line with Rotary Transfer and Turrets.” *IIE Transactions (Institute of Industrial Engineers)* 46 (9): 867–879.

Bautista, J., and J. Pereira. 2007. “Ant Algorithms for a Time and Space Constrained Assembly Line Balancing Problem.” *European Journal of Operational Research* 177 (3): 2016–2032.

Bautista, J., and J. Pereira. 2011. “Procedures for the Time and Space constrained Assembly Line Balancing Problem.” *European Journal of Operational Research* 212 (3): 473–481.

Baybards, I. 1986. “A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem.” *Management Science* 32 (8): 909–932.

Birattari, M. 2009. *Tuning Metaheuristics, A Machine Learning Perspective*. Berlin Heidelberg: Springer.

Blum, C., J. Puchinger, G. R. Raidl, and A. Roli. 2011. “Hybrid Metaheuristics in Combinatorial Optimization: A Survey.” *Applied Soft Computing Journal* 11 (6): 4135–4151.

Bock, S. 2008. “Using Distributed Search Methods for Balancing Mixed-model Assembly Lines in the Automotive Industry.” *OR Spectrum* 30 (3): 551–578.

Borba, L., and M. Ritt. 2014. “Exact and Heuristic Methods for the Assembly Line Worker Assignment and Balancing Problem.” *Computers & Operations Research* 45: 87–96.

Borisovsky, P. A., X. Delorme, and A. Dolgui. 2014. “Balancing Reconfigurable Machining Lines via a set Partitioning Model.” *International Journal of Production Research* 52 (13): 4026–4036.

Boysen, N., M. Fliedner, and A. Scholl. 2007. “A Classification of Assembly Line Balancing Problems.” *European Journal of Operational Research* 183 (2): 674–693.

Boysen, N., M. Fliedner, and A. Scholl. 2009. “Production Planning of Mixed-model Assembly Lines: Overview and Extensions.” *Production Planning and Control* 20 (5): 455–471.

Bukchin, Y., and I. Rabinowitch. 2006. “A Branch-and-bound Based Solution Approach for the Mixed-model Assembly Line-balancing Problem for Minimizing Stations and Task Duplication Costs.” *European Journal of Operational Research* 174 (1): 492–508.

Bukchin, J., and M. Tzur. 2000. “Design of Flexible Assembly Line to Minimize Equipment Cost.” *IIE Transactions (Institute of Industrial Engineers)* 32 (7): 585–598.

Burke, E. K., M. Gendreau, M. Hyde, G. Ochoa, G. Kendall, E. Özcan, and R. Qu. 2013. “Hyper-heuristics: A Survey of the State of the Art.” *Journal of the Operational Research Society* 64 (12): 1695–1724.

Chakravarty, A. K., and A. Shtub. 1985. “Balancing Mixed Model Lines with In-process Inventories.” *Management Science* 31 (9): 1161–1174.

Chen, J. C., C.-C. Chen, L.-H. Su, H.-B. Wu, and C.-J. Sun. 2012. “Assembly Line Balancing in Garment Industry.” *Expert Systems with Applications* 39 (11): 10073–10081.

Corominas, A., L. Ferrer, and R. Pastor. 2011. “Assembly Line Balancing: General Resource-constrained Case.” *International Journal of Production Research* 49 (12): 3527–3542.

Damm, R. B., M. G. C. Resende, and D. P. Ronconi. 2016. “A Biased Random Key Genetic Algorithm for the Field Technician Scheduling Problem.” *Computers & Operations Research* 75: 49–63.

De Bruecker, P., J. Van Den Bergh, J. Beliën, and E. Demeulemeester. 2015. “Workforce Planning Incorporating Skills: State of the Art.” *European Journal of Operational Research* 243 (1): 1–16.

Dolgui, A., and O. Battaïa. 2013. “A Taxonomy of Line Balancing Problems and their Solution Approaches.” *International Journal of Production Economics* 142 (2): 259–277.

Dou, J., X. Dai, and Z. Meng. 2010. “Optimisation for Multi-part Flow-line Configuration of Reconfigurable Manufacturing System Using GA.” *International Journal of Production Research* 48 (14): 4071–4100.

ElMaraghy, H., G. Schuh, W. Elmaraghy, F. Piller, P. Schnsleben, M. Tseng, and A. Bernard. 2013. “Product Variety Management.” *CIRP Annals - Manufacturing Technology* 62 (2): 629–652.

Emde, S., N. Boysen, and A. Scholl. 2010. “Balancing Mixed-model Assembly Lines: A Computational Evaluation of Objectives to Smoothen Workload.” *International Journal of Production Research* 48 (11): 3173–3191.

Fleszar, K., and K. S. Hindi. 2003. “An Enumerative Heuristic and Reduction Methods for the Assembly Line Balancing Problem.” *European Journal of Operational Research* 145 (3): 606–620.

Gadidov, R., and W. Wilhelm. 2000. "A Cutting Plane Approach for the Single-product Assembly System Design Problem." *International Journal of Production Research* 38 (8): 1731–1754.

Gao, J., L. Sun, L. Wang, and M. Gen. 2009. "An Efficient Approach for Type II Robotic Assembly Line Balancing Problems." *Computers and Industrial Engineering* 56 (3): 1065–1080.

Goyal, K. K., and P. K. Jain. 2016. "Design of Reconfigurable Flow Lines Using MOPSO and Maximum Deviation Theory." *International Journal of Advanced Manufacturing Technology* 84 (5–8): 1587–1600.

Hamta, N., S. M. T. Fatemi Ghomi, F. Jolai, and U. Bahalke. 2011. "Bi-criteria Assembly Line Balancing by Considering Flexible Operation Times." *Applied Mathematical Modelling* 35 (12): 5592–5608.

Hazir, Ö., X. Delorme, and A. Dolgui. 2015. "A Review of Cost and Profit Oriented Line Design and Balancing Problems and Solution Approaches." *Annual Reviews in Control* 40: 14–24.

Hoffmann, T. R. 1963. "Assembly Line Balancing with a Precedence Matrix." *Management Science* 9 (4): 551–562.

Kazemi, S. M., R. Ghodsi, M. Rabbani, and R. Tavakkoli-Moghaddam. 2011. "A Novel Two-stage Genetic Algorithm for a Mixed-model U-line Balancing Problem with Duplicated Tasks." *International Journal of Advanced Manufacturing Technology* 55 (9–12): 1111–1122.

Kimms, A. 2000. "Minimal Investment Budgets for Flow Line Configuration." *IIE Transactions (Institute of Industrial Engineers)* 32 (4): 287–298.

Larrañaga, P., and J. A. Lozano. 2001. *Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation (Genetic Algorithms and Evolutionary Computation)*. Norwell, MA: Kluwer Academic Publishers.

Moreira, M. C. O., J.-F. Cordeau, A. M. Costa, and G. Laporte. 2015. "Robust Assembly Line Balancing with Heterogeneous Workers." *Computers and Industrial Engineering* 88: 254–263.

Moreira, M. C. O., R. Pastor, A. M. Costa, and C. Miralles. 2017. "The Multi-objective Assembly Line Worker Integration and Balancing Problem of Type-2." *Computers & Operations Research* 82: 114–125.

Morrison, D. R., E. C. Sewell, and S. H. Jacobson. 2014. "An Application of the Branch, Bound, and Remember Algorithm to a New Simple Assembly Line Balancing Dataset." *European Journal of Operational Research* 236 (2): 403–409.

Nilakantan, J. M., I. Nielsen, S. G. Ponnambalam, and S. Venkataramanaiah. 2017. "Differential Evolution Algorithm for Solving RALB Problem Using Cost- and Time-based Models." *International Journal of Advanced Manufacturing Technology* 89 (1–4): 311–332.

Otto, A., C. Otto, and A. Scholl. 2013. "Systematic Data Generation and Test Design for Solution Algorithms on the Example of SALBPGen for Assembly Line Balancing." *European Journal of Operational Research* 228 (1): 33–45.

Özdemir, R. G., and Z. Ayağ. 2011. "An Integrated Approach to Evaluating Assembly-line Design Alternatives with Equipment Selection." *Production Planning and Control* 22 (2): 194–206.

Pape, T. 2015. "Heuristics and Lower Bounds for the Simple Assembly Line Balancing Problem Type 1: Overview, Computational Tests and Improvements." *European Journal of Operational Research* 240 (1): 32–42.

Peeters, M., and Z. Degraeve. 2006. "An Linear Programming based Lower Bound for the Simple Assembly Line Balancing Problem." *European Journal of Operational Research* 168: 716–731.

Pekin, N., and M. Azizoglu. 2008. "Bi Criteria Flexible Assembly Line Design Problem with Equipment Decisions." *International Journal of Production Research* 46 (22): 6323–6343.

Pereira, J. 2015. "Empirical Evaluation of Lower Bounding Methods for the Simple Assembly Line Balancing Problem." *International Journal of Production Research* 53 (11): 3327–3340.

Pinnoi, A., and W. E. Wilhelm. 1997. "A Family of Hierarchical Models for the Design of Deterministic Assembly Systems." *International Journal of Production Research* 35 (1): 253–280.

Pinnoi, A., and W. E. Wilhelm. 1998. "Assembly System Design: A Branch and Cut Approach." *Management Science* 44 (1): 103–118.

Queiroz Dos Santos, J. P., J. D. De Melo, A. D. Duarte Neto, and D. Aloise. 2014. "Reactive Search Strategies Using Reinforcement Learning, Local Search Algorithms and Variable Neighborhood Search." *Expert Systems with Applications* 41 (10): 4939–4949.

Renzi, C., F. Leali, M. Cavazzuti, and A. O. Andrisano. 2014. "A Review on Artificial Intelligence Applications to the Optimal Design of Dedicated and Reconfigurable Manufacturing Systems." *International Journal of Advanced Manufacturing Technology* 72 (1–4): 403–418.

Roshani, A., P. Fattahi, A. Roshani, M. Salehi, and A. Roshani. 2012. "Cost-oriented Two-sided Assembly Line Balancing Problem: A Simulated Annealing Approach." *International Journal of Computer Integrated Manufacturing* 25 (8): 689–715.

Scholl, A., and C. Becker. 2006. "State-of-the-art Exact and Heuristic Solution Procedures for Simple Assembly Line Balancing." *European Journal of Operational Research* 168 (3): 666–693.

Scholl, A., N. Boysen, and M. Fliedner. 2008. "The Sequence-dependent Assembly Line Balancing Problem." *OR Spectrum* 30 (3): 579–609.

Scholl, A., M. Fliedner, and N. Boysen. 2010. "Absalom: Balancing assembly lines with assignment restrictions." *European Journal of Operational Research* 200 (3): 688–701.

Sternatz, J. 2014. "Enhanced Multi-Hoffmann Heuristic for Efficiently Solving Real-world Assembly Line Balancing Problems in Automotive Industry." *European Journal of Operational Research* 235 (3): 740–754.

Sternatz, J. 2015. "The Joint Line Balancing and Material Supply Problem." *International Journal of Production Economics* 159: 304–318.

Sungur, B., and Y. Yavuz. 2015. "Assembly Line Balancing with Hierarchical Worker Assignment." *Journal of Manufacturing Systems* 37: 290–298.

Swift, K. G., J. D. Booker, and N. F. Edmondson. 2004. “Strategies and Case Studies in Assembly System Selection.” *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture* 218 (7): 675–688.

Thomopoulos, N. T. 1970. “Mixed Model Line Balancing with Smoothed Station Assignments.” *Management Science* 18 (9): 593–603.

Tonelli, F., M. Paolucci, D. Anghinolfi, and P. Taticchi. 2013. “Production Planning of Mixed-model Assembly Lines: A Heuristic Mixed Integer Programming based Approach.” *Production Planning and Control* 24 (1): 110–127.

Tuncel, G., and S. Topaloglu. 2013. “Assembly Line Balancing with Positional Constraints, Task Assignment Restrictions and Station Paralleling: A Case in an Electronics Company.” *Computers and Industrial Engineering* 64 (2): 602–609.

Vilà, M., and J. Pereira. 2014. “A Branch-and-bound Algorithm for Assembly Line Worker Assignment and Balancing Problems.” *Computers & Operations Research* 44: 105–114.

Wang, S., L. Wang, M. Liu, and Y. Xu. 2015. “A Hybrid Estimation of Distribution Algorithm for the Semiconductor Final Testing Scheduling Problem.” *Journal of Intelligent Manufacturing* 26 (5): 861–871.

Wilhelm, W. E., and R. Gadidov. 2004. “A Branch-and-cut Approach for a Generic Multiple-product, Assembly-system Design Problem.” *INFORMS Journal on Computing* 16 (1): 39–55.

Yagmahan, B. 2011. “Mixed-model Assembly Line Balancing Using a Multi-objective Ant Colony Optimization Approach.” *Expert Systems with Applications* 38 (10): 12453–12461.

Youssef, A. M. A., and H. A. Elmaraghy. 2006. “Modelling and Optimization of Multiple-aspect RMS Configurations.” *International Journal of Production Research* 44 (22): 4929–4958.

Appendix 1. Lower bounds derived from the Dantzig–Wolfe reformulation

The linear relaxation of formulation (10)–(14) has been shown to provide tight lower bounds for the SALBP (Pereira 2015). Following the precedence constraints relaxation proposed in Peeters and Degraeve (2006), we focus our attention on optimally solving the linear relaxation of formulation (A1)–(A3), in which precedence constraints among tasks in different workstations are not enforced. The later makes possible to reduce the master problem to a set partitioning formulation, which still provides a tight bound (Pereira 2015).

$$\min \sum_{q\in Q} \bar{c}_q \cdot z_q \tag{A1}$$

subject to:

$$\sum_{q\in Q} q_i \cdot z_q = 1 \qquad \forall i \in V \tag{A2}$$

$$0 \le z_q \le 1 \qquad \forall q \in Q \tag{A3}$$

Pricing is used to obtain a subset of Q that suffices to check the optimality of the current solution. Initially, the subset is composed of solitary assignments (that is, feasible assignments in which only one task belongs to the assignment), and then additional variables are added by solving a pricing problem according to the shadow prices of (A2).

Let π_i be the shadow price of constraint (A2) associated to task i. Then, a valid formulation for the pricing problem corresponds to (A4)–(A9), in which variables $x_i, i \in V$ indicate whether task i belongs to the best feasible assignment according to the shadow prices, and variables $v_r, r \in R$ correspond to the resource requirements of the assignment. If the optimal solution to (A4) has a negative value, then the corresponding assignment needs to be added to the subset of Q and model (A1)–(A3) is to be solved again. The process is repeated until the optimal solution to (A4)–(A9) has a non-negative cost.

$$\min \sum_{r\in R} c_r\, v_r - \sum_{i\in V} \pi_i\, x_i \tag{A4}$$

subject to:

$$\sum_{i\in V} t_{ir}\, x_j \le c_r \qquad \forall p \in P \tag{A5}$$

$$x_{i'} + x_{i''} \le x_i + 1 \qquad (\forall i \in V) \wedge (\forall i' \in P_j^*) \wedge (\forall i'' \in F_j^*) \tag{A6}$$

$$v_r \ge n_{ir}\, x_i \qquad \forall r \in R,\ \forall i \in V \tag{A7}$$

$$v_r \ge 0 \qquad \forall r \in R \tag{A8}$$

$$x_i \in \{0, 1\} \qquad \forall i \in V \tag{A9}$$

Objective (A4) minimises the reduced cost of the feasible assignment. A feasible assignment fulfills the following constraints: the cycle time cumulative constraints for all of the products, $p \in P$, represented by constraint set (A5); the internal precedence constraints (A6); and the assignment of the required resources, as calculated by constraint set (A7). Constraint sets (A8) and (A9) correspond to the domain of the variables. In the proposed implementation, the formulation and the pricing problems are solved using an off-the-shelf commercial linear programming software, see Section 4, as they are only used as an indication of the quality of the proposed solution approach.

Appendix 2. Data from the industrial example

The following constitutes the instance description of the problem. The operation times of both products were slightly modified and then scaled up to a cycle time equal to 1000 time units. The cost was set to be two times higher when the workstation is equipped with the resource, that is $c^{w} = c_{r=1} = 1$, as per suggestions of the company. The precedence graph is depicted in Figure B1, while the remaining information and the solutions are described in Table B1.

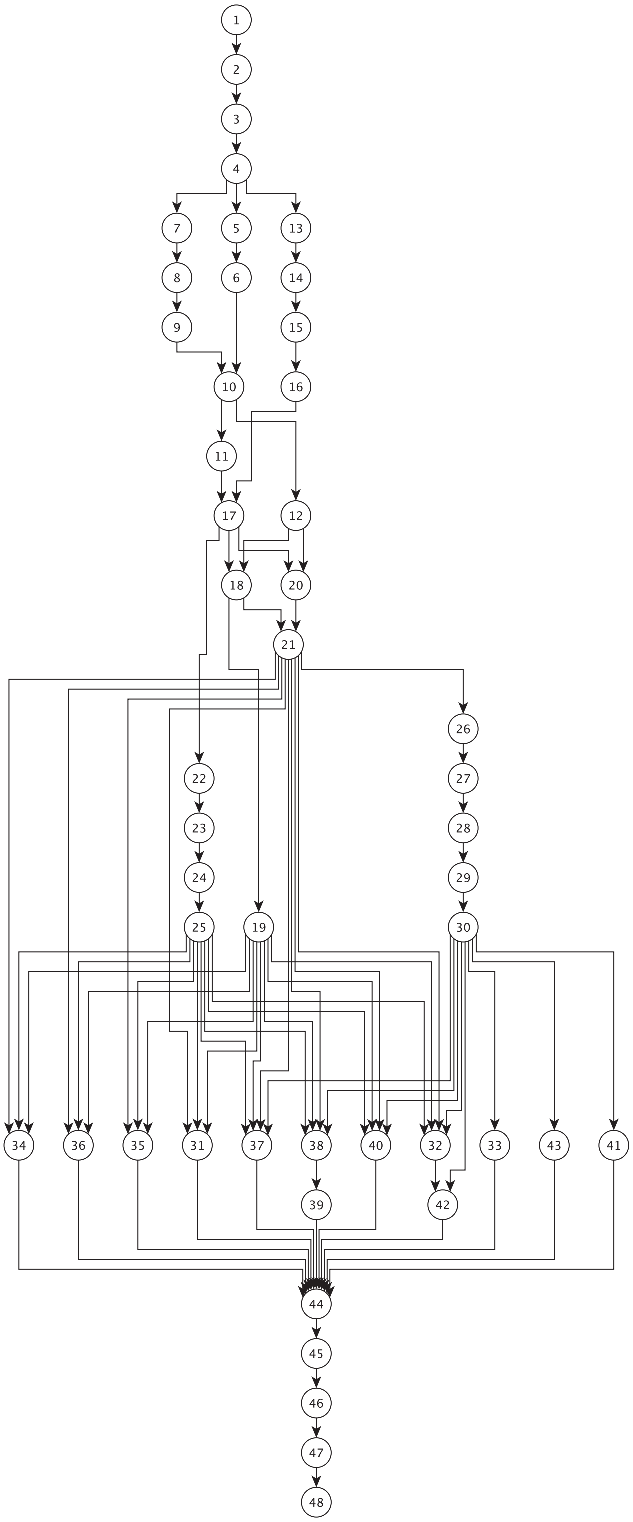

Figure B1. Precedence graph of the case study.

Table B1. Data from the industrial example. For each operation (columns id.), the operation times for each product (columns $t_{i,\text{male}}$ and $t_{i,\text{female}}$), resource requirements (column Resource) and current solution (column Corrent sol.) are provided. Moreover, one of the optimal solutions provided by EDA (column EDA sol.) and one of the optimal solutions for the SALBP instance considering only the male (column $\text{SALBP}_{\text{male}}$) pr the female (colum $\text{SALBP}_{\text{female}}$) model are also detailed. Whenever a task is not required for a specific garment, be it male or female, no station assignment is reported, as indicated by the '-' sign.

id.	$t_{i,\text{male}}$	$\text{SALBP}_{\text{male}}$	$t_{i,\text{female}}$	$\text{SALBP}_{\text{female}}$	Resource	EDA sol.	Current sol.
1	405	1	360	1	0	1	1
2	90	1	80	1	1	1	1
3	270	1	240	1	1	1	1
4	45	1	40	1	1	1	1
5	0	–	960	2	0	2	2
6	0	–	240	3	0	3	3
7	225	2	0	–	0	2	2
8	90	3	0	–	0	2	3
9	90	3	0	–	0	2	3
10	135	3	80	3	1	3	3
11	0	–	340	3	1	3	3
12	23	3	0	–	0	1	2
13	180	2	0	–	0	2	2
14	158	2	0	–	0	2	2
15	135	2	0	–	0	2	2
16	203	3	0	–	1	3	3
17	23	3	20	3	0	3	3
18	338	3	280	3	1	3	3
19	158	5	340	8	1	4	4
20	338	5	0	–	1	4	4
21	158	5	160	4	1	4	4
22	1000	4	0	–	0	5	5
23	135	5	0	–	0	6	6
24	990	6	0	–	0	7	7
25	135	7	0	–	0	8	8
26	0	–	360	4	0	4	4
27	0	–	280	4	0	5	5
28	0	–	600	5	0	7	5
29	0	–	380	5	1	7	6
30	0	–	1000	6	1	8	7
31	495	8	0	–	0	8	8
32	180	9	200	9	1	10	9
33	0	–	300	7	0	9	8
34	90	7	0	–	0	10	9
35	113	7	0	–	0	9	10
36	135	7	0	–	0	9	9
37	495	8	620	8	1	9	9
38	90	7	80	9	0	9	8
39	23	7	20	9	0	10	8
40	68	7	60	9	0	10	8
41	0	–	80	7	0	10	10
42	0	–	20	7	0	10	10
43	0	–	60	7	0	10	8
44	90	9	80	9	1	10	10
45	540	9	480	9	1	10	10
46	90	10	80	10	1	11	11
47	540	10	480	10	1	11	11
48	135	10	120	10	1	11	11