Computers & Industrial Engineering 88 (2015) 254–263

Contents lists available at ScienceDirect

Computers & Industrial Engineering

journal homepage: www.elsevier.com/locate/caie

Robust assembly line balancing with heterogeneous workers

Mayron César O. Moreira [a,*], Jean-François Cordeau [b], Alysson M. Costa [c], Gilbert Laporte [d]

[a] Departamento de Ciência da Computação, Universidade Federal de Lavras, Campus Universitário, Caixa Postal 3037, 37200-000, Lavras, Brazil
[b] CIRRELT and Canada Research Chair in Logistics and Transportation, HEC Montréal, Montréal H3T 2A7, Canada
[c] School of Mathematics and Statistics, University of Melbourne, Parkville, 3010 VIC, Melbourne, Australia
[d] CIRRELT and Canada Research Chair in Distribution Management, HEC Montréal, Montréal H3T 2A7, Canada

A R T I C L E I N F O

Article history:
Received 19 January 2015
Received in revised form 8 July 2015
Accepted 9 July 2015
Available online 15 July 2015

Keywords:
Assembly line balancing
Heterogeneous workers
Robust optimization
Integer programming
Constructive heuristic

A B S T R A C T

Assembly lines are manufacturing systems in which a product is assembled progressively in workstations by different workers or machines, each executing a subset of the needed assembly operations (or tasks). We consider the case in which task execution times are worker-dependent and uncertain, being expressed as intervals of possible values. Our goal is to find an assignment of tasks and workers to a minimal number of stations such that the resulting productivity level respects a desired robust measure. We propose two mixed-integer programming formulations for this problem and explain how these formulations can be adapted to handle the special case in which one must integrate a particular set of workers in the assembly line. We also present a fast construction heuristic that yields high quality solutions in just a fraction of the time needed to solve the problem to optimality. Computational results show the benefits of solving the robust optimization problem instead of its deterministic counterpart.

1. Introduction

Assembly lines are flow-oriented systems that rely on the division of work. The operations needed to assemble a given product are assigned to different workstations, and this assignment must respect technical constraints, such as precedence relations between tasks. In its basic form, the resulting optimization problem is known as the *simple assembly line balancing problem* (SALBP) and its two most common variants consist in minimizing the number of workstations needed while ensuring a given productivity level (problem of type I) or maximizing productivity with a fixed number of workstations (problem of type II). The reader interested in the SALBP is referred to Baybars (1986), Scholl (1999), Scholl and Becker (2006), Becker and Scholl (2006), Boysen, Fliedner, and Scholl (2007, 2008), Battaïa and Dolgui (2013), and Sivasankaran and Shahabudeen (2014).

One of the main assumptions of the SALBP is that task execution times are worker-independent. This assumption is relaxed in the *assembly line worker assignment and balancing problem* (ALWABP), where one must simultaneously assign both tasks and workers to stations. Our interest in the ALWABP is motivated by its application to the management of assembly lines in sheltered work centers for the disabled (SWDs) (Miralles, García-Sabater, Andrés, & Cardos, 2007). Since the original study of Miralles et al. (2007), the ALWABP has received a considerable amount of attention. The problem has been tackled by means of heuristics (Blum & Miralles, 2011; Chaves, Lorena, & Miralles, 2009; Moreira & Costa, 2009; Moreira, Ritt, Costa, & Chaves, 2012; Mutlu, Polat, & Supciller, 2013) and exact algorithms (Borba & Ritt, 2014; Miralles, García-Sabater, Andrés, & Cardos, 2008; Vilà & Pereira, 2014). In addition, several authors have studied variants of the problem with features such as job rotation schedules, mixed-model production, parallel stations or worker collaboration (Araújo, Costa, & Miralles, 2012, 2015; Cortez & Costa, 2015; Costa & Miralles, 2009; Moreira & Costa, 2013). It is worth noting that most of these studies have considered problems of type II, which are relevant in the context of SWDs which usually aim to provide work experience to as many workers with disabilities as possible.

In many types of decision problems, deterministic models are inadequate and uncertainty should be taken into account explicitly in the optimization model so as to properly represent real-life situations. In the case of assembly lines, uncertainty is often present in task execution times and arises from a series of factors such as the unpredictability and variability in work rates, as well as in skill and motivation levels (Becker & Scholl, 2006). In the management of SWDs, these variations can be very significant due to the high heterogeneity of the workers and to their lack of prior work experience. Learning effects or successive improvements to the line are sometimes modeled by means of dynamic task times, which use

* Corresponding author.
E-mail addresses: mayron.moreira@dcc.ufla.br (M.C.O. Moreira), jean-francois.cordeau@hec.ca (J.-F. Cordeau), alysson.costa@unimelb.edu.au (A.M. Costa), gilbert.laporte@cirrelt.ca (G. Laporte).

http://dx.doi.org/10.1016/j.cie.2015.07.004
0360-8352/

fuzzy numbers with known membership functions (e.g. Boucher, 1987; Toksarı, İşleyen, Güner, & Baykoç, 2010; Zacharia & Nearchou, 2012). Other studies consider stochastic task execution times under some probability distributions. For more details, see Suresh and Sahu (1994), Nkasu and Leung (1995), Sotskov, Dolgui, and Portmann (2006), Özcan (2010), Fazlollahtabar, Hajmohammadi, and Es'aghzadeh (2011), Özcan, Kellegöz, and Toklu (2011), Gurevsky, Battaïa, and Dolgui (2012, 2013).

Robust optimization (Gabrel, Murat, & Thiele, 2014) is a popular approach for the handling of uncertainty when the probability distribution of the uncertain parameters is unknown. Here we assume that only an interval of possible values for each task execution time is available. We adopt a budget-of-uncertainty robustness approach as proposed by Bertsimas and Sim (2003, 2004), which has been successfully applied to a large variety of problems (Alem & Morabito, 2011; Bertsimas & Thiele, 2006; Hazır, Erel, & Günalay, 2011; Lu, Ying, & Lin, 2014; Moon & Yao, 2011; Solyalı, Cordeau, & Laporte, 2012). According to this paradigm, the combined scaled increase of uncertain parameters from their nominal values is limited by a budget, as will be seen in the following section. Hazır and Dolgui (2013) have proposed a robust approach for the SALBP of type II by including an uncertainty budget on each station, and have solved the resulting problem by means of a Benders decomposition algorithm. Gurevsky, Hazir, Battaïa, and Dolgui (2013) have considered a robust SALBP of type I which was solved by branch-and-bound.

In this paper, we extend the approaches for the SALBP developed by Hazır and Dolgui (2013) and Gurevsky, Hazir et al. (2013) to the ALWABP. We focus on the problem of type I, following the framework proposed by Moreira, Miralles, and Costa (2015) in which the ALWABP is extended beyond the context of SWDs to that of conventional assembly lines. There, the goal is to integrate a set of workers with disabilities in a conventional assembly line while minimizing the number of extra stations needed, resulting in the *assembly line worker integration and balancing problem* (ALWIBP).

This paper makes four main scientific contributions. We first introduce the *robust assembly line worker assignment and balancing problem* with the objective of minimizing the number of workstations (RALWABP-1). We then describe two formulations for the general problem and we explain how these formulations can be adapted to handle the integration of a set of heterogeneous workers (RALWIBP-1). Thirdly, we propose a fast heuristic for the RALWIBP-1 which yields high quality solutions within short computing times. Finally, we show that solving the robust problem leads to much better solutions compared to solving its deterministic counterpart.

The remainder of the paper is organized as follows. In Section 2, we provide a formal definition of the problem and we introduce our two mathematical models. This is followed by a description of the heuristic in Section 3, and by the results of computational experiments in Section 4. Finally, Section 5 ends the paper with some conclusions and avenues for future research.

2. Problem description and formulations

Let $S = \{1,\dots,m\}$ be a an ordered set of workstations, $W = \{1,\dots,o\}$ a set of workers, with $|W| = |S|$, and $N = \{1,\dots,n\}$ a partially ordered set of tasks. The partial order on the tasks can be defined by an acyclic precedence graph $G = (N,E)$, where arc $(i,j) \in E$ indicates that task i is an immediate predecessor of task j. We also define the graph $G^* = (N,E^*)$ as the transitive closure of G, i.e., there exists an arc $(i,j) \in E^*$ whenever there is a path from i to j in G. In addition to the above definitions, we use the following notation:

$t_{wi} \in \mathbb{N}^* \cup \{\infty\}$	time of task $i \in N$ when executed by worker $w \in W$;
$W_i = \{w \in W : t_{wi} \neq \infty\}$	set of workers who are able to execute task $i \in N$;
$N_w = \{i \in N : w \in W_i\}$	set of tasks that worker $w \in W$ is able to execute;
$D_i = \{j \in N : (j,i) \in E\}$	set of immediate predecessors of task $i \in N$;
$D_i^* = \{j \in N : (j,i) \in E^*\}$	set of all predecessors of task $i \in N$;
$F_i = \{j \in N : (i,j) \in E\}$	set of immediate successors of task $i \in N$;
$F_i^* = \{j \in N : (i,j) \in E^*\}$	set of all successors of task $i \in N$.

Given a fixed productivity rate, associated with a cycle time $\bar{c}$, the aim of the ALWABP-1 is to determine an assignment of tasks to workers minimizing the number of stations required while respecting precedence relationships. In this study, we assume that the task execution times are uncertain and have unknown probability distributions. We consider, however, that the execution times are independent of each other and that the execution time of task i by worker w belongs to the interval $[\bar{t}_{wi}, \bar{t}_{wi} + \hat{t}_{wi}]$, where $\bar{t}_{wi}$ is the nominal value and $\hat{t}_{wi}$ is the maximum deviation from $\bar{t}_{wi}$.

Sections 2.1 and 2.2 present two RALWABP-1 formulations adapted from Borba and Ritt (2014) and Moreira et al. (2015), respectively. Although the model of Miralles et al. (2007) can also be adapted to handle uncertainty, preliminary tests have shown that finding feasible solutions to its robust counterpart is extremely hard, even for moderate size instances. For this reason, we do not consider it in this study. Section 2.3 considers the special case of the RALWIBP-1.

2.1. A robust model based on the formulation of Borba and Ritt (2014)

Borba and Ritt (2014) introduced an ALWABP-2 formulation that considers the assignment of tasks to workers and the relative position of the workers in the assembly line. Let x_{wi} be a binary variable equal to one if and only if task $i \in N$ is assigned to worker $w \in W$, and d_{vw} be a binary variable equal to one if and only if worker v precedes worker w. In order to modify their model for the type-I problem, we introduce binary variables z_w equal to one if and only if worker w is assigned to the assembly line. We also define parameter $\bar{c}$ as the maximum allowed cycle time in the line. The modified model is the following:

$$M1: \text{ minimize } \sum_{w \in W} z_w \tag{1}$$

subject to

$$\sum_{w \in W_i} x_{wi} = 1 \qquad i \in N \tag{2}$$

$$\sum_{i \in N_w} \bar{t}_{wi} x_{wi} \leqslant \bar{c} \qquad w \in W \tag{3}$$

$$d_{vw} \geqslant x_{vi} + x_{wj} - 1 \qquad (i,j) \in E, \quad v \in W_i, \quad w \in W_j \setminus \{v\} \tag{4}$$

$$d_{uw} \geqslant d_{uv} + d_{vw} - 1 \qquad \{u,v,w\} \subseteq W; \quad |\{u,v,w\}| = 3 \tag{5}$$

$$d_{vw} + d_{wv} \leqslant 1 \qquad v \in W, \quad w \in W \setminus \{v\} \tag{6}$$

$$\sum_{i \in N_w} x_{wi} \leqslant |N_w| z_w \qquad w \in W \tag{7}$$

$$x_{wi} \in \{0,1\} \qquad w \in W, \quad i \in N_w \tag{8}$$

$$d_{vw} \in \{0,1\} \qquad v \in W, \quad w \in W \setminus \{v\} \tag{9}$$

$$z_w \in \{0,1\} \qquad w \in W. \tag{10}$$

The objective function (1) minimizes the number of stations by minimizing the number of workers assigned to the assembly line. Constraints (2) ensure that each task is executed by one worker.

Constraints (3) guarantee that the resulting workload at each station does not exceed the desired cycle time. Precedence relations are enforced by constraints (4): if task i is assigned to worker v, task j is assigned to worker w and $(i,j) \in E$, then worker v must precede worker w. Constraints (5) and (6) impose the transitivity and the anti-symmetry of the precedence relationships, respectively. Constraints (7) force a worker to be assigned to the assembly line if there are tasks assigned to him.

Borba and Ritt (2014) have shown that the above model can be strengthened through the following task continuity constraints:

$$x_{wj} \geqslant x_{wi} + x_{wq} - 1 \quad i \in N,\ j \in F_i^*,\ q \in F_j^*,\ w \in W_i \cap W_j \cap W_q \tag{11}$$

$$x_{wq} + x_{wi} \leqslant 1 \quad i \in N,\ j \in F_i^*,\ q \in F_j^*,\ w \in W_i \cap (W \setminus W_j) \cap W_q. \tag{12}$$

Constraints (11) force task j to be executed by worker w whenever both a preceding task i and a following task q are executed by the same worker. Constraints (12) state that if a task i is assigned to worker w and its successor (predecessor) j is infeasible for w ($\bar{t}_{wj} = \infty$), then no successor (predecessor) of j can be assigned to w. We denote by $M1^*$ the augmented model $M1$ containing these strengthening constraints.

To define the robust counterpart of model $M1^*$, we use continuous variables u_{wi} measuring the scaled deviation of the processing time of task $i \in N_w$ when executed by worker $w \in W$. We also define a budget of uncertainty Γ_w for each worker $w \in W$. Using this notation, we can write the following non-linear robust model $R1^*$:

$$R1^*: \ \text{minimize} \sum_{w \in W} z_w \tag{13}$$

subject to

(2), (4)–(12) and

$$\sum_{i \in N_w} \bar{t}_{wi} x_{wi} + \max\left\{ \sum_{i \in N_w} \hat{t}_{wi} x_{wi} u_{wi} : \sum_{i \in N_w} u_{wi} \leqslant \Gamma_w u_{wi} \in [0,1] \quad i \in N_w \right\} \leqslant \bar{c} \quad w \in W. \tag{14}$$

By associating dual variables θ_w and α_{wi} with the constraints in the inner maximization problems in the left-hand-side of constraints (14), the model can be linearized as follows (see Bertsimas & Sim (2003, 2004)):

$$RL1^*: \ \text{minimize} \sum_{w \in W} z_w \tag{15}$$

subject to

(2), (4)–(12) and

$$\sum_{i \in N_w} \bar{t}_{wi} x_{wi} + \Gamma_w \theta_w + \sum_{i \in N_w} \alpha_{wi} \leqslant \bar{c} \qquad w \in W \tag{16}$$

$$\theta_w + \alpha_{wi} \geqslant \hat{t}_{wi} x_{wi} \qquad w \in W, \quad i \in N_w \tag{17}$$

$$\theta_w \in \mathbb{R}_+ \qquad w \in W \tag{18}$$

$$\alpha_{wi} \in \mathbb{R}_+ \qquad w \in W, \quad i \in N_w. \tag{19}$$

2.2. A robust model based on the formulation of Moreira et al. (2015)

We now consider binary variables x_{si} taking value 1 if and only if task $i \in N$ is assigned to station $s \in S$, and binary variables y_{sw} taking value 1 if and only if worker $w \in W$ is assigned to station $s \in S$. Then, taking $l_w \in \mathbb{R}$ as a positive constant, the type I version of the model proposed by Moreira et al. (2015) can be written as

$$M2: \ \text{minimize} \sum_{s \in S} \sum_{w \in W} y_{sw} \tag{20}$$

subject to

$$\sum_{s \in S} x_{si} = 1 \qquad i \in N \tag{21}$$

$$\sum_{s \in S} y_{sw} \leqslant 1 \qquad w \in W \tag{22}$$

$$\sum_{w \in W} y_{sw} \leqslant 1 \qquad s \in S \tag{23}$$

$$\sum_{\substack{s \in S \\ s \geqslant k}} x_{si} \leqslant \sum_{\substack{s \in S \\ s \geqslant k}} x_{sj} \qquad (i,j) \in E, \quad k \in S \setminus \{1\} \tag{24}$$

$$\sum_{i \in N_w} \bar{t}_{wi} x_{si} \leqslant \bar{c} + l_w(1 - y_{sw}) \qquad s \in S, \quad w \in W \tag{25}$$

$$x_{si} + y_{sw} \leqslant 1 \qquad s \in S, \quad i \in N, \quad w \in W \setminus W_i \tag{26}$$

$$\sum_{i \in N_w} x_{si} \leqslant \sum_{w \in W} |N_w| y_{sw} \qquad s \in S \tag{27}$$

$$\sum_{w \in W} y_{s+1,w} \leqslant \sum_{w \in W} y_{sw}, \qquad s \in S \tag{28}$$

$$x_{si} \in \{0,1\}, \qquad s \in S, \quad i \in N \tag{29}$$

$$y_{sw} \in \{0,1\}, \qquad s \in S, \quad w \in W. \tag{30}$$

The objective function (20) minimizes the number of stations, while constraints (22) and (23) require that each worker be assigned to a single station, and each station have at most one worker, respectively. Constraints (21) force the assignment of all tasks. Precedence relationships are imposed by constraints (24). Note that we are using an adaptation of the inequalities proposed by Ritt and Costa (2011) because these dominate the other forms of precedence constraints. Constraints (25) ensure that the cycle time is respected. Constant l_w must be sufficiently large to make the associated constraint redundant if $y_{sw} = 0$. Therefore, we take $l_w = \sum_{i \in N_w} \bar{t}_{wi} - \bar{t}_i$, where $\bar{t}_i = \min_{w \in W_i}\{\bar{t}_{wi}\}$. Constraints (26) take care of incompatibilities between tasks and workers. Constraints (27) imply that a worker must be assigned to a station if at least one task is assigned to that station. Finally, the symmetry-breaking constraints (28) establish an order in the use of the y_{sw} variables. The purpose of these constraints is to avoid the presence of multiple equivalent solutions that would differ only by the numbering of stations.

Analogously to the case of $M1^*$, continuity constraints can be written as:

$$x_{sj} \geqslant x_{si} + x_{sq} - 1 \qquad s \in S, \quad i \in N, \quad j \in F_i^*, \quad q \in F_j^* \tag{31}$$

$$y_{sw} + x_{si} + x_{sq} \leqslant 2 \qquad s \in S, \quad i \in N, \quad j \in F_i^*, \quad q \in F_j^*, \quad w \in W_i \cap (W \setminus W_j) \cap W_q. \tag{32}$$

We denote by $M2^*$ the model obtained by adding constraints (31) and (32) to $M2$. As before, we define u_{swi} as a continuous variable that measures the scaled deviation of the processing time of task $i \in N$ when it is executed by worker $w \in W$ in station s. Take l'_w as a positive constant such that $l'_w = \sum_{i \in N_w}[(\bar{t}_{wi} + \hat{t}_{wi}) - (\bar{t}_i + \hat{t}_i)]$, where $\hat{t}_i = \min_{w \in W_i}\{\hat{t}_{wi}\}$. Model $R2^*$ can be defined as follows:

$$R2^*: \ \text{minimize} \sum_{s \in S} \sum_{w \in W} y_{sw} \tag{33}$$

subject to

(21)–(24), (26)–(32) and

$$\sum_{i \in N_w} \bar{t}_{wi} x_{si} + \max\left\{ \sum_{i \in N_w} \hat{t}_{wi} x_{si} u_{swi} : \sum_{i \in N_w} u_{swi} \leqslant \Gamma_w u_{swi} \in [0,1] \quad i \in N_w \right\} \leqslant \bar{c} + l'_w(1 - y_{sw}) \qquad s \in S, \quad w \in W. \tag{34}$$

Let θ_{sw} and α_{swi} be the dual variables associated with the constraints in the inner maximization problem on the left-hand side of constraints (34). As before, using strong duality, a linear version of model $R2^*$ can be written as

$$RL2^*: \text{ minimize} \sum_{s\in S}\sum_{w\in W} y_{sw} \tag{35}$$

subject to

(21)–(24), (26)–(32) and

$$\sum_{i\in N_w}\bar{t}_{wi}x_{si}+\Gamma_w\theta_{sw}+\sum_{i\in N_w}\alpha_{swi}\leqslant\bar{c}+l'_w(1-y_{sw})\quad s\in S,\ w\in W \tag{36}$$

$$\theta_{sw}+\alpha_{swi}\geqslant\hat{t}_{wi}x_{si}\quad s\in S,\ w\in W,\ i\in N_w \tag{37}$$

$$\theta_{sw}\in\mathbb{R}_+\quad s\in S,\ w\in W \tag{38}$$

$$\alpha_{swi}\in\mathbb{R}_+\quad s\in S,\ w\in W,\ i\in N_w. \tag{39}$$

2.3. ALWIBP-1: a particular case

The *assembly line worker integration and balancing problem of type I* (ALWIBP-1) is a particular case of the ALWABP-1 requiring that a given subset of the workers be inserted in the assembly line. Let $H\subseteq W$ denote this set of workers. We propose two formulations for the RALWIBP-1:

- *BR*: obtained from the $RL1^*$ model by adding constraints (40):

$$\sum_{i\in N_w} x_{wi}\geqslant 1\qquad w\in H. \tag{40}$$

- *MMC*: composed by model $RL2^*$ with constraints (41):

$$\sum_{s\in S} y_{sw}=1\qquad w\in H. \tag{41}$$

This problem arises, for example, in conventional assembly lines when one must integrate workers with disabilities in the context of corporate social responsibility policies. As in Moreira et al. (2015), we assume that workers belonging to the set $W\setminus H$ have homogeneous task execution times (denoted by $\bar{t}_i$) while those in set H correspond to the workers with disabilities and have different execution times for the same task. Moreover, these workers may be unable to execute some tasks. Both models can be strengthened by taking advantage of these characteristics. By defining $H=\{1,\ldots,\omega\}$ and $W=\{1,\ldots,\omega,\ldots,o\}$, we can add the following constraints to BR:

$$z_{w+1}\leqslant z_w\qquad w\in W. \tag{42}$$

Constraints (42) are symmetry-breaking constraints and follow from the fact that workers with disabilities must be assigned to the line; these are defined as the first workers in the set W. We can also eliminate redundant constraints in model *MMC* and reduce the number of variables since it is not necessary to determine to which workstations the homogeneous workers are assigned. To this end, we follow the ideas proposed by Patterson and Albracht (1975). Let ϵ be an artificial task with $\bar{t}_\epsilon=\bar{t}_{w\epsilon}=0$, and $D_\epsilon=\{i\in N: F_i=\varnothing\}$ be a set of tasks that are the predecessors of ϵ. Define the extended sets $N'=N\cup\{\epsilon\}$, $W_\epsilon=H$, $N_w=\{i\in N': w\in W_i\cap H\}$ and $E'=E\cup\{(i,\epsilon): i\in D_\epsilon\}$. We first present an alternative $M2^*$ formulation:

$$M2a^*: \text{ minimize}\sum_{s\in S} s x_{s\epsilon} \tag{43}$$

subject to

(41) and

$$\sum_{w\in H} y_{sw}\leqslant 1\qquad s\in S \tag{44}$$

$$\sum_{i\in N_w}\bar{t}_{wi}x_{si}\leqslant\bar{c}+l_w(1-y_{sw})\qquad s\in S,\quad w\in H \tag{45}$$

$$\sum_{s\in S} x_{si}=1\qquad i\in N' \tag{46}$$

$$\sum_{\substack{s\in S\\ s\geqslant k}} x_{si}\leqslant\sum_{\substack{s\in S\\ s\geqslant k}} x_{sj}\qquad (i,j)\in E',\quad k\in S\setminus\{1\} \tag{47}$$

$$\sum_{i\in N'}\bar{t}_i x_{si}\leqslant\bar{c}\qquad s\in S \tag{48}$$

$$x_{si}+y_{sw}\leqslant 1\qquad s\in S,\quad i\in N',\quad w\in H\setminus W_i \tag{49}$$

$$y_{sw}\leqslant\sum_{i\in N} x_{si}\qquad s\in S,\quad w\in H \tag{50}$$

$$x_{si}\in\{0,1\}\qquad s\in S,\quad i\in N'. \tag{51}$$

The objective function (43) minimizes the index related to the last workstation, i.e., the one that executes the dummy task ϵ. Constraints (48) are the cycle time inequalities when a homogeneous worker is assigned to a station. Constraints (49) guarantee that a worker from H is not assigned to tasks which he is not able to execute. Constraints (50) guarantee that no worker from H is assigned to a station with no tasks.

Following the steps described to obtain a linear robust formulation described in Section 2.2, we redefine the robust *MMC* formulation, where $\hat{t}_\epsilon=\hat{t}_{w\epsilon}=0$, and ρ_s and λ_{si} are dual variables analogous to variables θ_{sw} and α_{swi} defined earlier. We also consider Γ as the budget of uncertainty for all homogeneous workers. We obtain the following model:

$$MMC: \text{ minimize}\sum_{s\in S} s x_{s\epsilon} \tag{52}$$

subject to

(41), (44), (46), (47), (49), (50), (51) and

$$\sum_{i\in N'}\bar{t}_i x_{si}+\Gamma\rho_s+\sum_{i\in N'}\lambda_{si}\leqslant\bar{c}\qquad s\in S \tag{53}$$

$$\rho_s+\lambda_{si}\geqslant\hat{t}_i x_{si}\qquad s\in S,\quad i\in N' \tag{54}$$

$$\sum_{i\in N_w}\bar{t}_{wi}x_{si}+\Gamma_w\theta_{sw}+\sum_{i\in N_w}\alpha_{swi}\leqslant\bar{c}+l'_w(1-y_{sw}),\quad s\in S,\quad w\in H \tag{55}$$

$$\theta_{sw}+\alpha_{swi}\geqslant\hat{t}_{wi}x_{si},\qquad s\in S,\quad w\in H,\quad i\in N_w \tag{56}$$

$$\rho_s\in\mathbb{R}_+\qquad s\in S \tag{57}$$

$$\lambda_{si}\in\mathbb{R}_+\qquad s\in S,\quad i\in N' \tag{58}$$

$$\theta_{sw}\in\mathbb{R}_+,\qquad s\in S,\quad w\in H \tag{59}$$

$$\alpha_{swi}\in\mathbb{R}_+,\qquad s\in S,\quad w\in H,\quad i\in N_w. \tag{60}$$

3. Constructive heuristic

Models BR and MMC can be solved by means of a branch-and-cut algorithm implemented in a general-purpose MIP solver such as CPLEX. In this section, we introduce a constructive heuristic that is much faster than branch-and-cut and yields high quality solutions.

The *Robust Insertion Constructive Heuristic* (RICH) procedure relies on the constructive heuristic proposed by Moreira et al. (2015) for theALWIBP-1. Given a starting solution, this heuristic attempts to incorporate a set of workers by minimizing the number of additional stations needed. The RICH is made up of four mains phases: (i) construction of an initial solution; (ii) segmentation of the assembly line; (iii) task assignment; and (iv) update of the current solution. We describe each step in Sections 3.1–3.4, respectively, and present a pseudo-code in Section 3.4.1.

Table 1
Elementary priority rules.

T	Maximum task time over all workers and tasks
PW	Maximum sum of t_{wi} and task times of all its available successors
TdL	Maximum ratio of task time t_{wi} and identifier L_i of the latest station available for the assignment of task i
TdS	Maximum ratio of task time t_{wi} and slack $L_i - E_i - \epsilon$, where E_i is the identifier of the earliest station available for the assignment of task i and $\epsilon \approx 0$
F	Maximum number of successors over all tasks

Table 2
Number of instances, out of 1600 instances tested, for which the solution obtained by the ALWIBP-1 model was feasible for the RALWIBP-1.

Γ	$\hat{t}_{wi}$ (%)			
	1	5	10	50
1	606	97	16	0
2.5	406	11	0	0
5	260	2	0	0

3.1. Step 1: initial solution

We consider as a starting point the solution obtained by an adaptation of the heuristic of Scholl and Voss (1996). This is a station-based method that sorts available tasks according to some priority rules, and assigns them to the current station as long as the cycle time constraints are respected. The modification of the method to deal with the robust case requires changing the definition of the availability of a task. Let w' be a candidate worker and U_s be the set of tasks that have already been assigned to station s. A task i whose predecessors have already been assigned is available to be inserted in station s when operated by worker w' if:

$$\sum_{j \in U_s \cup \{i\}} \bar{t}_{w'j} + \max\left\{ \sum_{j \in U_s \cup \{i\}} \hat{t}_{w'j} u_j : \sum_{j \in U_s \cup \{i\}} u_j \leqslant \Gamma_{w'} u_j \in [0,1]\ j \in U_s \cup \{i\} \right\} \leqslant \bar{c}. \tag{61}$$

Among the available tasks, that with best priority criterion is selected. Table 1 presents the elementary priority rules used in this phase. When no additional task can be added to the current station without violating the cycle time constraint, the method moves to the new empty station and the process is repeated until all tasks have been assigned.

3.2. Step 2: segmentation of the assembly line

Let $W_a \subseteq H$ be a set of workers that have not yet been assigned. Consider the set S_c of candidate stations for the assignment of worker $w \in W_a$. Define s_b as the last processed station (in terms of tasks and workers) and m_c as the total number of stations in the current solution. Then we define S_c as

$$S_c = \left\{ s_b + 1, \ldots, s_b + 1 + \left\lfloor \frac{m_c - s_b}{|W_a|} \right\rfloor \right\}. \tag{62}$$

In words, we use a sequential procedure in which only a fraction of the available stations are considered for the assignment of a candidate worker, at each step. The size of this fragment is given by the ratio between the remaining stations in the line and the number of workers not yet assigned. This conservative approach enhances the probability of being able to assign all workers since a considerable part of the line is always "protected" for future assignments. It also has the beneficial side-effect of distributing the heterogeneous workers along the line, which is associated with a better integration of these workers.

3.3. Step 3: task assignment

In this step, we obtain a solution by considering all assignments of workers in W_a to stations in S_c, one at a time. When a station $s \in S_c$ is considered for the assignment of worker $w \in W_a$, we fix the allocation of tasks and workers in stations prior to s. For station s and those that follow, the solution is reconstructed with a modification of the Scholl and Voss (1996) heuristic to take robustness into account (see Section 3.1).

We adopt the elementary priority rules shown in Table 1, based on their good performance on similar problems, as verified in Otto and Otto (2014). In order to obtain diversified solutions, we apply the principle of aggregation, prioritizing the maximum value of the following rules: $T^{1\%\bar{c}}$, $T^{3.3\%\bar{c}}$, $PW^{1\%\bar{c}}$, $PW^{3.3\%\bar{c}}$, $TdL^{1\%\bar{c}}$, $TdL^{3.3\%\bar{c}}$, $TdS^{1\%\bar{c}}$, $TdS^{3.3\%\bar{c}}$, $F^{5\%n}$ and $F^{10\%n}$. For those rules that consider the task times (resp. number of successors), this strategy rounds the priority criterion to a multiple of a percentage of cycle time (resp. number of tasks), with the goal of preserving or even reducing some portion of "noise" in its value. For example, consider rule $T^{1\%\bar{c}}$, which is evaluated as $\lfloor \frac{t_i}{0.01\bar{c}} \rfloor 0.01\bar{c}$. Taking $\bar{c} = 1000$, $t_1 = 185$ and $t_2 = 512$, task aggregated at $1\%\bar{c}$-level will be $t_1^{1\%\bar{c}} = 180$ and $t_1^{1\%\bar{c}} = 510$. In both elementary and aggregated rules, we consider the task index as a tie break. Motivated by the insights reported in Otto and Otto (2014) concerning composite rules, we combine the five elementary rules in pairs such that the first rule can receive weights 1, 2 and 0.5 (the weight of the second rule is fixed to 1). Changing the roles of the primary and secondary rules, we arrive at a total of 3 × 5 × 4 = 60 combinations. Overall, 75 different pairs of rules were considered.

3.4. Step 4: selection of the best assignment

In each step, the best solution x^* obtained among all combinations of $w \in W_a$ and $s \in S_c$ is chosen. Suppose that x^1 and x^2 are solutions obtained by different assignments of w and s. We state that x^1 is better than x^2 ($x^1 < x^2$) if the number of stations in x^1 is smaller than in x^2. If a tie occurs, we select the solution with the smaller value of the smoothness index, which measures the imbalance level of the assembly line (Scholl, 1999).

3.4.1. Formal pseudo-code

Let x be a solution to the original SALBP problem, possibly already modified with the inclusion of workers in H in stations prior to a given station s. We define genSol(x, s) as a function that takes a solution x as an input, assigns tasks to workers in stations prior to s, and uses the method described in Section 3.3 for the assignment of the remaining stations. We also define $sts(x)$ as a function returning the number of stations in solution x. Algorithm 1 provides the pseudo-code of the RICH procedure.

Line 1 creates an initial solution as explained in Section 3.1. The division of the assembly line into segments of candidate stations is performed in lines 4 and 17. Lines 6–19 correspond to the main loop of the procedure, where lines 7–14 are related to Step 3 and lines 15–18 are related to Step 4.

Note that one can use the same strategy in a backward fashion, by first assigning workers at the end of the line and then moving to the beginning. In this case, the updates in lines 4 and 17 become $S_c \leftarrow \left\{ \left\lceil \frac{m_c}{|W_a|} \right\rceil, \ldots, m_c \right\}$ and $S_c \leftarrow \left\{ \left\lfloor \frac{s_b - 1}{|W_a|} \right\rfloor, \ldots, s_b - 1 \right\}$, respectively. Since the tasks and workers assigned to the end of the line are kept fixed, the algorithm may need to insert intermediate stations. The tie breaker is the same as the one described in Section 3.4.

In addition to assigning workers in a forward or backward manner, we can reverse the direction of the precedence arcs. This way, an assignment of workers and tasks will still be valid but, due to the heuristic nature of the approach, different solutions can be obtained. Combining the two possible precedence directions (forward – normal or backward – reversed) with the direction of assignment of workers, we have four possible variants of the algorithm.

4. Computational results

The models and algorithms were coded in C++. We used the branch-and-cut framework of IBM CPLEX 12.6 as MIP solver. The tests were carried out on an Intel Xeon X550 2.67 GHz processor with 12 GB of memory and running under Linux. Four threads were used for each example. Finally, we imposed limits of time and search space of 1 h and 6 GB, respectively.

Algorithm 1. Robust Insertion Constructive Heuristic (RICH).

Let x^i be an initial SALBP solution;
$m_c \leftarrow sts(x^i)$;
$W_a \leftarrow H$;
$S_c \leftarrow \left\{1, \ldots, \left\lceil \frac{m_c}{|W_a|} \right\rceil \right\}$;
Let x^* be the incumbent solution;
while $W_a \neq \emptyset$ **do**
  for all $w \in W_a$ **do**
    for all $s \in S_c$ **do**
      $x^c \leftarrow$ genSol(x^i, s);
      if $x^c < x^*$ **then**
        $w_b \leftarrow w; s_b \leftarrow s; x^* \leftarrow x^c$;
      end if
    end for
  end for
  $W_a \leftarrow W_a \setminus \{w_b\}$;
  $m_c \leftarrow sts(x^*)$;
  $S_c \leftarrow \left\{s_b + 1, \ldots, s_b + 1 + \left\lfloor \frac{m_c - s_b}{|W_a|} \right\rfloor \right\}$;
  $x^i \leftarrow x^*$;
end while
return x^* (best solution found).

We have performed computational experiments using the benchmark of Moreira et al. (2015) for the ALWIBP-1. These authors have selected 100 SALBP instances from Otto, Otto, and Scholl (2013) from which the base task precedence graph and the task execution times (t_i) were extracted. ALWIBP-1 instances were then generated by introducing task time variations for one of the workers (in the ranges $U[t_i, 2t_i]$ and $U[t_i, 5t_i]$) and task/worker incompatibilities (set at 10% or 20% of the tasks). They used the same scheme to generate examples with two, three and four heterogeneous workers. In this study, we consider the group of 1600 mid-size instances with 50 tasks, since preliminary tests showed that the models BR and MMC had difficulty in identifying feasible solutions for larger instances. We determined the additional task times $\hat{t}_{wi}, w \in W$ and $i \in N$, as 1%, 5%, 10% and 50% of their nominal values $\bar{t}_{wi}$. Also, we set the budget of uncertainty parameter Γ_w equal to Γ for each $w \in H$, where Γ can take values of 1, 2.5 and 5. These values are chosen to represent situations with low total uncertainty, in which the combined scaled change in the execution times is limited by one, to situations where this parameter can reach 5, which was the average number of tasks executed by a worker in the tested instances. In Section 4.1, we give the results for the robust formulations, while the heuristic approaches are analyzed in Section 4.2.

4.1. Experiment 1: formulations

For the execution of both models, an upper bound on the number of stations is computed as the number of stations obtained by the adaptation of the Scholl and Voss (1996) heuristic for the robust case increased by $|H|$. Concerning the priority rule, we consider the combination (TdL, Id, 1), the best elementary one according to Otto and Otto (2014). The first set of experiments, presented in Table 2, evaluates the number of best solutions obtained by the ALWIBP-1 model (Moreira et al., 2015) that are acceptable for the RALWIBP-1. In this table, we show the percentage of acceptable ALWIBP-1 solutions for each combination of the parameter Γ and the percentage of additional time $\hat{t}_{wi}$ (recall that for each combination of budget and percentage of additional time, we have tested 1600 instances). One can see that in most cases, a solution to the deterministic problem is in fact infeasible for the robust counterpart, even when considering low levels of task time variability and budget of uncertainty. This justified the need to explicitly solve the robust problem.

Table 3 summarizes the overall performance of the RALWIBP-1 models concerning the number of instances solved. Columns 1 and 2 indicate the values of "Budget" and the "Status" of the solutions, respectively. Columns "1", "5", "10" and "50" represent the number of instances whose best solutions obtained within the time limit of 1 h were "Optimal", "Feasible" or "Unavailable" during the time limit for additional task times $\hat{t}_{wi}$ of 1%, 5%, 10% and 50% of the nominal task times $\bar{t}_{wi}$. The MMC formulation outperforms the BR model taking into account the number of optimal solutions and the unsolved problems in all scenarios. This superiority is confirmed by the following tests, in which we consider solutions that are at least feasible for both models.

We now look at the price of robustness, defined in this context as the percentage of additional stations needed in the robust solution with respect to its deterministic counterpart. Table 3 presents

Table 3
Number of solutions per status.

Γ	Status	BR $\hat{t}_{wi}$ (%)				MMC $\hat{t}_{wi}$ (%)			
		1	5	10	50	1	5	10	50
1	Optimal	918	794	677	273	1311	1278	1192	915
	Feasible	678	798	905	1248	288	322	408	684
	Unavailable	4	8	18	79	1	0	0	1
2.5	Optimal	873	649	476	13	1277	1177	1024	746
	Feasible	720	935	1091	1361	323	423	576	852
	Unavailable	7	16	33	226	0	0	0	2
5	Optimal	518	486	450	383	1175	1086	953	780
	Feasible	1015	1022	1060	1129	425	514	647	813
	Unavailable	67	92	90	88	0	0	0	7

Table 4
Average price of robustness (%) considering the best solution of BR and MMC.

Γ	$\hat{t}_{wi}$ (%)			
	1	5	10	50
1	0.48	1.81	3.76	19.74
2.5	0.72	3.49	6.73	36.69
5	0.97	4.56	9.06	46.91

Table 5
Average gap (%) of BR and MMC models obtained by the MIP solver.

Γ	BR $\hat{t}_{wi}$ (%)				MMC $\hat{t}_{wi}$ (%)			
	1	5	10	50	1	5	10	50
1	10.49	12.01	12.38	22.62	1.77	1.99	2.64	7.75
2.5	12.05	13.95	16.23	33.67	2.03	2.71	4.13	13.09
5	11.79	15.13	18.73	38.06	2.12	3.27	5.00	14.73

Table 6
Average computational time (s) of BR and MMC models obtained by the MIP solver.

Γ	BR $\hat{t}_{wi}$(%)				MMC $\hat{t}_{wi}$(%)			
	1	5	10	50	1	5	10	50
1	1929	2147	2340	3066	868	948	1119	1632
2.5	2024	2410	2712	3538	970	1181	1458	1947
5	2073	2496	2948	3545	1001	1327	1677	2190

these figures for different values of Γ and $\hat{t}_{wi}$. The higher the additional task times or the safety parameter Γ, the higher the price to pay (in terms of additional stations) in order to guarantee robust solutions (these results are more detailed in Table A.8 of Appendix A). In a practical context, the manager can use this information to decide the appropriate level of robustness desired (expressed as the parameter Γ) as a trade-off between the increase in the line cost in the robust solution versus the cost of stopping the line and reprocessing products in the case the deterministic solution is chosen (see Table 4).

In Tables 5 and 6, with respect to the average gap obtained by CPLEX and the computational effort to solve the models, we observe that large values of Γ and $\hat{t}_{wi}$ increase the computing times. The MMC model is faster than BR, requiring approximately half of the time, on average, to solve the instances. The good performance of the MMC is due to the way it considers the presence of homogeneous workers, avoiding the redundancy of constraints. Complete results can be seen in Tables A.9 and A.10 of Appendix A.

Table A.8
Price of robustness (%) considering the best solution of BR and MMC.

Γ	$\|W\|$	Var	Inc	$\hat{t}_{wi}$ (%)			
				1	5	10	50
1	1	2	10	0.28	1.38	3.69	20.10
			20	0.56	1.41	3.65	19.89
		5	10	0.57	2.14	3.43	19.13
			20	0.24	1.71	3.27	18.90
	2	2	10	0.64	1.82	3.28	19.29
			20	0.78	2.20	3.73	20.25
		5	10	0.53	2.19	4.02	19.61
			20	0.11	2.04	4.44	20.41
	3	2	10	0.71	1.58	3.82	20.71
			20	0.37	1.41	3.69	20.00
		5	10	0.56	1.85	3.69	19.38
			20	0.65	1.56	3.40	18.70
	4	2	10	0.38	2.15	4.45	20.89
			20	0.33	1.49	3.82	19.42
		5	10	0.50	2.24	3.47	19.54
			20	0.39	1.86	4.32	19.55
	Average			0.48	1.81	3.76	19.74
2.5	1	2	10	0.63	3.72	6.52	37.28
			20	0.65	3.70	6.47	37.14
		5	10	0.69	3.21	6.26	37.88
			20	0.45	2.96	6.37	36.97
	2	2	10	1.06	3.18	6.19	37.61
			20	0.78	3.45	7.15	37.42
		5	10	0.67	3.77	7.15	36.51
			20	0.34	3.43	6.98	37.13
	3	2	10	0.79	3.64	7.51	37.90
			20	0.55	3.34	7.37	37.48
		5	10	0.41	3.73	6.30	35.60
			20	0.87	3.05	5.47	34.22
	4	2	10	0.96	4.13	7.49	37.34
			20	0.84	3.56	6.91	36.93
		5	10	1.06	3.06	6.49	34.90
			20	0.74	3.98	6.97	34.76
	Average			0.72	3.49	6.73	36.69
5	1	2	10	0.64	4.47	9.80	49.71
			20	0.65	4.30	9.38	48.66
		5	10	1.49	4.04	8.86	48.45
			20	0.80	4.21	9.53	47.67
	2	2	10	1.06	4.18	8.88	48.35
			20	1.20	4.68	9.62	48.20
		5	10	0.74	4.94	9.13	46.28
			20	0.86	5.26	8.64	46.87
	3	2	10	1.04	4.48	9.20	48.09
			20	0.63	5.35	9.59	47.03
		5	10	0.61	4.18	8.49	44.84
			20	1.12	3.77	7.96	43.33
	4	2	10	1.10	5.60	9.48	48.24
			20	1.19	4.39	8.70	46.45
		5	10	1.48	4.55	9.02	44.87
			20	0.86	4.56	8.60	43.48
	Average			0.97	4.56	9.06	46.91

Table 7
Results obtained by the RICH.

Γ	$\hat{t}_{wi}$ (%)	Gap^- (%) (Pr. rule)	Gap^+ (%) (Pr. rule)	Gap (%)	Ties (%)	Impr (%) (#inst)
1	1	3.97 (F, TdS, 1)	4.69 ($F^{10\%n}$, Id, 1)	4.37	58.20	−13.33 (1)
	5	4.14 (F, PW, 2)	4.85 ($F^{10\%n}$, Id, 1)	4.47	56.56	−6.67 (1)
	10	3.97 (PW, TdS, 1)	4.69 ($F^{10\%n}$, Id, 1)	4.40	56.65	−11.76 (1)
	50	3.21 ($PW^{3.3\%\bar{c}}$, Id, 1)	4.21 ($F^{10\%n}$, Id, 1)	3.52	60.85	−5.69 (34)
2.5	1	4.06 (F, PW, 2)	4.73 (T, TdL, 1)	4.44	57.26	– (0)
	5	3.99 (F, PW, 1)	4.74 ($F^{10\%n}$, Id, 1)	4.37	57.02	– (0)
	10	3.98 (F, PW, 2)	4.74 (F, Id, 1)	4.28	56.73	−5.56 (1)
	50	2.54 (PW, T, 2)	3.85 ($F^{10\%n}$, Id, 1)	2.76	62.45	−5.89 (98)
5	1	4.00 (F, PW, 2)	4.76 (T, F, 1)	4.45	57.11	– (0)
	5	6.45 (F, PW, 1)	7.19 ($F^{10\%n}$, Id, 1)	6.78	36.20	−6.67 (1)
	10	3.56 (F, PW, 2)	4.41 ($F^{10\%n}$, ID, 1)	3.89	59.13	−6.11 (2)
	50	2.03 ($PW^{3.3\%\bar{c}}$, Id, 1)	3.58 ($F^{10\%n}$, Id, 1)	2.33	62.59	−6.51 (128)

Table A.9
Gap (%) of BR and MMC models obtained by the MIP solver.

Γ	$\lvert W\rvert$	Var	Inc	BR $\hat{t}_{wi}$ (%)				MMC $\hat{t}_{wi}$ (%)			
				1	5	10	50	1	5	10	50
1	1	2	10	8.52	10.71	13.93	22.30	0.33	0.67	0.93	5.43
			20	10.16	12.48	13.54	23.86	0.23	0.39	1.00	5.02
		5	10	10.91	7.66	9.26	19.25	0.74	0.46	0.79	5.23
			20	8.37	9.70	9.65	21.20	0.47	0.49	1.26	4.86
	2	2	10	12.12	14.77	14.34	21.68	0.97	1.02	1.24	5.14
			20	10.31	13.45	13.61	24.10	0.86	1.25	1.31	6.48
		5	10	7.90	11.42	8.98	19.73	1.07	1.45	2.38	6.68
			20	7.68	11.09	10.42	20.44	1.07	1.98	2.66	7.55
	3	2	10	11.22	14.61	14.75	25.97	1.16	1.17	2.11	8.23
			20	11.92	13.80	13.26	25.15	1.43	1.28	2.35	7.99
		5	10	11.43	11.50	10.75	22.23	2.23	2.76	3.72	9.52
			20	7.99	9.46	10.84	20.88	2.54	2.60	3.16	8.49
	4	2	10	14.99	15.46	17.06	26.98	2.44	2.83	3.38	9.21
			20	12.40	13.48	13.63	24.63	2.26	2.48	3.25	9.00
		5	10	11.14	12.61	12.84	22.56	5.46	4.92	5.93	12.42
			20	10.82	9.89	11.28	20.95	5.06	6.10	6.82	12.80
	Average			10.49	12.01	12.38	22.62	1.77	1.99	2.64	7.75
2.5	1	2	10	11.35	13.32	16.52	36.79	0.58	1.05	1.79	10.42
			20	11.28	13.18	15.86	33.96	0.52	1.03	1.81	10.44
		5	10	12.70	12.90	15.00	34.01	0.72	1.04	2.19	11.20
			20	10.58	11.98	12.67	31.41	0.48	1.04	2.95	11.73
	2	2	10	14.62	12.85	18.83	36.59	1.11	1.38	2.98	11.51
			20	12.26	13.25	16.16	34.17	0.88	1.23	3.33	12.28
		5	10	10.36	12.70	13.68	31.99	1.21	1.93	3.46	11.97
			20	8.58	12.05	13.71	30.72	1.45	2.03	3.42	13.10
	3	2	10	12.71	17.17	19.30	37.55	1.62	2.80	4.61	12.29
			20	11.87	14.37	18.76	37.22	1.46	2.52	3.92	12.84
		5	10	12.23	12.72	15.28	31.95	2.86	3.75	5.43	14.23
			20	10.98	13.11	14.18	29.84	3.02	3.31	4.79	14.89
	4	2	10	14.00	17.37	17.87	36.78	2.40	3.62	4.89	13.53
			20	14.82	17.67	18.59	35.07	2.27	3.81	5.20	13.94
		5	10	13.43	14.21	17.01	30.37	6.08	6.03	7.81	17.96
			20	11.00	14.35	16.19	30.30	5.77	6.81	7.48	17.03
	Average			12.05	13.95	16.23	33.67	2.03	2.71	4.13	13.09
5	1	2	10	10.73	16.38	21.99	40.66	0.58	1.15	3.32	11.63
			20	10.00	14.83	22.93	41.76	0.56	1.07	3.07	10.88
		5	10	13.04	11.13	17.61	40.03	1.23	1.33	2.94	12.25
			20	11.23	12.30	16.85	37.77	0.59	1.67	3.30	11.76
	2	2	10	11.96	17.18	19.43	39.39	1.24	1.91	3.83	12.91
			20	11.98	15.79	19.33	40.24	0.99	1.67	3.62	13.00
		5	10	10.83	13.98	18.17	34.79	1.35	2.64	4.20	14.99
			20	9.03	11.84	15.44	36.14	1.54	2.87	3.84	13.72
	3	2	10	13.83	17.57	21.13	39.96	1.39	3.17	4.61	14.10
			20	11.17	16.17	18.91	39.49	1.60	3.54	4.72	13.44
		5	10	11.83	15.76	16.92	35.36	2.91	4.43	6.21	16.83
			20	9.26	13.15	15.93	35.32	3.07	4.32	6.31	16.63
	4	2	10	15.86	19.52	20.87	41.81	2.58	3.98	6.08	15.43
			20	15.35	16.54	19.72	39.93	3.03	3.58	6.10	15.93
		5	10	12.58	15.44	19.38	33.09	5.50	7.28	8.79	21.44
			20	9.92	14.49	15.00	33.25	5.82	7.71	9.02	20.70
	Average			11.79	15.13	18.73	38.06	2.12	3.27	5.00	14.73

4.2. Experiment 2: constructive heuristic

Table 7 presents the results of the RICH algorithm for the RALWIBP-1. Columns 1 and 2 provide averages for each combination of budget and additional task time. Columns "Gap$^-$ (%) (Pr. Rule)" and "Gap$^+$ (%) (Pr. Rule)" show the deviations obtained by the priority rule which had the best and worst performance, respectively. Column "Gap (%)" presents the average deviation considering the best result over all priority rules. Note that in these three columns, we have compared the heuristic solution with the best solutions obtained by the BR and MMC models. Column "Tie (%)" represents the percentage of instances for which we obtained the same value of objective function as in the best known solutions. Column "Impr (%) (#inst)" presents the percentage of improvement and the number of instances in which the RICH was the best, since in some cases none of the formulations could prove optimality.

On average, the heuristic runs within less than one second. By analyzing Tables 3 and 7, we find that the quality of the gap is inversely proportional to the number of optimal solutions. Composite rules involving priority rules F and PW yield the best results. On the other hand, aggregated rules did not work well, as one can see in Column "Gap$^+$ (%) (Pr. Rule)". Nevertheless, the algorithm appears to be robust, since the difference between the best and worst gaps is small. The percentage of ties (about 60%) demonstrates the accuracy of the RICH. Concerning the improvements found by the heuristic, we can note that they coincide with the scenarios whose number of optimal solutions are lower. This can be explained by the fact that the feasible solutions obtained by the models in these cases are mostly equal to the pre-defined upper bound.

Table A.10
Computational time (s) of BR and MMC models obtained by the MIP solver.

Γ	$\lvert W\rvert$	Var	Inc	BR $\hat{t}_{wi}$ (%)				MMC $\hat{t}_{wi}$ (%)			
				1	5	10	50	1	5	10	50
1	1	2	10	1652	1844	2203	2833	169	437	463	1165
			20	1581	1903	2186	3018	142	270	488	1252
		5	10	1797	1810	1957	2976	346	331	504	976
			20	1611	1910	1958	2853	246	283	566	932
	2	2	10	1941	2173	2111	2923	470	495	653	1419
			20	1810	2089	2298	2977	460	587	630	1440
		5	10	1813	2077	2223	3012	660	804	1020	1462
			20	1698	2246	2447	3219	629	958	1100	1356
	3	2	10	1969	2191	2534	3072	664	780	1086	1832
			20	1974	2112	2477	3114	745	735	1062	1726
		5	10	2041	2269	2358	3140	1285	1426	1583	1983
			20	2039	2009	2299	3136	1405	1347	1370	1815
	4	2	10	2253	2520	2722	3258	1227	1444	1596	1934
			20	2235	2295	2653	3249	1238	1307	1604	1976
		5	10	2270	2534	2511	3120	2147	2002	2150	2533
			20	2174	2373	2500	3150	2050	1970	2026	2305
	Average			1929	2147	2340	3066	868	948	1119	1632
2.5	1	2	10	1816	2223	2449	3504	249	496	828	1710
			20	1648	2360	2456	3483	242	569	834	1815
		5	10	1933	1985	2485	3501	409	554	980	1523
			20	1729	2055	2591	3482	268	557	1228	1558
	2	2	10	2056	2156	2757	3517	543	668	1132	1644
			20	1882	2171	2769	3518	450	654	1342	1742
		5	10	1879	2418	2691	3560	748	969	1367	1732
			20	1734	2448	2621	3567	795	1045	1258	1844
	3	2	10	1997	2513	2933	3568	830	1290	1654	1677
			20	2005	2546	2901	3535	816	1216	1534	1947
		5	10	2173	2509	2689	3566	1484	1689	1764	2327
			20	2157	2331	2702	3513	1470	1525	1528	1842
	4	2	10	2403	2749	2749	3588	1377	1725	1771	2137
			20	2325	2688	2792	3587	1321	1637	1890	2287
		5	10	2431	2667	2836	3571	2310	2168	2119	2773
			20	2213	2738	2975	3554	2213	2129	2099	2593
	Average			2024	2410	2712	3538	970	1181	1458	1947
5	1	2	10	1746	2297	2973	3546	263	524	1312	1634
			20	1685	2191	3005	3551	283	529	1267	1496
		5	10	1950	2182	2858	3483	527	630	1276	1460
			20	1778	2276	3017	3484	309	775	1235	1629
	2	2	10	2053	2453	2961	3542	599	844	1453	2050
			20	1988	2378	3043	3540	535	886	1430	1809
		5	10	1924	2617	2927	3503	786	1264	1522	2235
			20	1807	2375	2745	3519	855	1282	1505	1796
	3	2	10	2135	2572	2918	3578	779	1432	1723	2210
			20	1944	2657	2820	3550	828	1551	1802	2322
		5	10	2234	2629	2913	3576	1483	1879	1903	2410
			20	2181	2443	2981	3594	1525	1712	1696	2680
	4	2	10	2561	2809	2914	3600	1310	1726	1939	2521
			20	2373	2637	2918	3535	1409	1693	1984	2613
		5	10	2509	2731	3223	3576	2288	2259	2475	2997
			20	2300	2687	2957	3551	2243	2242	2316	3185
	Average			2073	2496	2948	3545	1001	1327	1677	2190

To measure the modification of the initial solution by the RICH, we use the Solution Similarity Index (SSI) proposed in Otto and Otto (2014). This metric computes the percentage of pairs of tasks assigned together in two solutions. We observe in our tests that approximately 23% of these keep their same relative position when considering 1%, 5% and 10% time deviations. For the higher value of deviation, this average increases to 47%. Finally, we observe that the forward allocation of workers with a backward graph yielded the best results, followed by the forward–forward scheme.

5. Conclusions

The consideration of uncertainty in assembly lines has recently gained attention in the scientific literature. It is also an important preoccupation in practice since variations in task execution times can significantly affect performance. This study introduces a robust assembly line balancing problem with heterogeneous workers under task time uncertainty, and focusing on the special case of the problem requiring the integration of a given set of workers. Two models and a constructive heuristic were designed. Computational experiments taking into account a range of different instances have shown the difficulty of solving the RALWIBP-1 to optimality, even though feasible solutions were obtained in most cases. We have developed an efficient heuristic procedure called *robust insertion constructive heuristic* (RICH). This heuristic yields high quality solutions (with an average gap of 4%) within computing times that do not exceed one second on the tested cases. It also identified the best known solution for many of the instances.

The results indicate that the proposed approach can provide solutions that are considerably more robust to task time variation at relatively low increase in the number of stations and workers needed. This provides an effective tool for line assembly managers when designing more stable lines. From this practical point of view, further research might investigate this topic from a cost perspective, computing the added value of robustness as a trade-off between the savings obtained with less line interruption and product reprocessing versus the added cost of a larger line.

Acknowledgements

This research was supported by São Paulo Research Foundation (FAPESP), by CAPES-Brazil and by the Canadian Natural Sciences and Engineering Research Council under Grants 227837-09 and 39682-10. We are also grateful to Calcul Québec for providing the computing facilities used to run the experiments.

Appendix A

The extended Tables A.8–A.10 show the average values of price of robustness, optimality gap and computational times obtained by the RALWIBP-1 models. The results are grouped according to the characteristics of the instances: $|W|$ (number of workers to be inserted in the assembly line), "Var" (variability of task execution times) and "Inc" (percentage of incompatible tasks). The remaining columns, namely "1", "5", "10" and "50", represent the results for each variation of percentage of additional nominal task time.

References

Alem, D., & Morabito, R. (2011). Production planning in furniture settings via robust optimization. *Computers & Operations Research, 39*, 139–150.
Araújo, F. F. B., Costa, A. M., & Miralles, C. (2012). Two extensions for the assembly line worker assignment and balancing problem: Parallel stations and collaborative approach. *International Journal of Production Economics, 140*, 483–495.
Araújo, F. F. B., Costa, A. M., & Miralles, C. (2015). Balancing parallel assembly lines with disabled workers. *European Journal of Industrial Engineering, 9*, 344–365.
Battaïa, O., & Dolgui, A. (2013). A taxonomy of line balancing problems and their solution approaches. *International Journal of Production Economics, 142*, 259–277.
Baybars, I. (1986). A survey of exact algorithms for the simple assembly line balancing problem. *Management Science, 32*, 909–932.
Becker, C., & Scholl, A. (2006). A survey on problems and methods in generalized assembly line balancing. *European Journal of Operational Research, 168*, 694–715.
Bertsimas, D., & Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming Series B, 98*, 49–71.
Bertsimas, D., & Sim, M. (2004). The price of robustness. *Operations Research, 52*, 35–53.
Bertsimas, D., & Thiele, A. (2006). A robust optimization approach to inventory theory. *Operations Research, 54*, 150–168.
Blum, C., & Miralles, C. (2011). On solving the assembly line worker assignment and balancing problem via beam search. *Computers & Operations Research, 38*, 328–339.
Borba, L., & Ritt, M. (2014). A heuristic and a branch-and-bound algorithm for the assembly line worker assignment and balancing problem. *Computers & Operations Research, 45*, 87–96.
Boucher, T. (1987). Choice of assembly line design under task learning. *International Journal of Production Research, 25*, 513–524.
Boysen, N., Fliedner, M., & Scholl, A. (2007). A classification of assembly line balancing problems. *European Journal of Operational Research, 183*, 674–693.
Boysen, N., Fliedner, M., & Scholl, A. (2008). Assembly line balancing: Which model to use when? *International Journal of Production Economics, 111*, 509–528.
Chaves, A. A., Lorena, L. A. N., & Miralles, C. (2009). Hybrid metaheuristic for the assembly line worker assignment and balancing problem. *Lecture Notes on Computer Science, 5818*, 1–14.
Cortez, P. M. C., & Costa, A. M. (2015). Sequencing mixed-model assembly lines operating with a heterogeneous workforce. *International Journal of Production Research, 53*, 3419–3432.
Costa, A. M., & Miralles, C. (2009). Job rotation in assembly lines employing disabled workers. *International Journal of Production Economics, 120*, 625–632.
Fazlollahtabar, H., Hajmohammadi, H., & Es'aghzadeh, A. (2011). A heuristic methodology for assembly line balancing considering stochastic time and validity testing. *The International Journal of Advanced Manufacturing Technology, 52*, 311–320.
Gabrel, V., Murat, C., & Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research, 235*, 471–483.
Gurevsky, E., Battaïa, O., & Dolgui, A. (2012). Balancing of simple assembly lines under variations of task processing times. *Annals of Operations Research, 201*, 265–286.
Gurevsky, E., Battaïa, O., & Dolgui, A. (2013). Stability measure for a generalized assembly line balancing problem. *Discrete Applied Mathematics, 161*, 377–394.
Gurevsky, E., Hazir, O., Battaïa, O., & Dolgui, A. (2013). Robust balancing of straight assembly lines. *Journal of the Operational Research Society, 64*, 1607–1613.
Hazır, O., & Dolgui, A. (2013). Assembly line balancing under uncertainty: Robust optimization models and exact solution method. *Computers & Industrial Engineering, 65*, 261–267.
Hazır, O., Erel, E., & Günalay, Y. (2011). Robust optimization models for the discrete time/cost trade-off problem. *International Journal of Production Economics, 130*, 87–95.
Lu, C.-C., Ying, K.-C., & Lin, S.-W. (2014). Robust single machine scheduling for minimizing total flow time in the presence of uncertain processing times. *Computers & Industrial Engineering, 74*, 102–110.
Miralles, C., García-Sabater, J. P., Andrés, C., & Cardos, M. (2007). Advantages of assembly lines in sheltered workcentres for disabled. A case study. *International Journal of Production Economics, 110*, 187–197.
Miralles, C., García-Sabater, J. P., Andrés, C., & Cardos, M. (2008). Branch and bound procedures for solving the assembly line worker assignment and balancing problem: Application to sheltered work centres for disabled. *Discrete Applied Mathematics, 156*, 352–367.
Moon, Y., & Yao, T. (2011). A robust mean absolute deviation model for portfolio optimization. *Computers & Operations Research, 38*, 1251–1258.
Moreira, M. C. O., & Costa, A. M. (2009). A minimalist yet efficient tabu search algorithm for balancing assembly lines with disabled workers. In: *Anais do XLI Simpósio Brasileiro de Pesquisa Operacional* (pp. 660–671) Porto Seguro, Brazil.
Moreira, M. C. O., & Costa, A. M. (2013). Hybrid heuristics for planning job rotation schedules in assembly lines with heterogeneous workers. *International Journal of Production Economics, 141*, 552–560.
Moreira, M. C. O., Miralles, C., & Costa, A. M. (2015). Model and heuristics for the assembly line worker integration and balancing problem. *Computers & Operations Research, 54*, 64–73.
Moreira, M. C. O., Ritt, M., Costa, A. M., & Chaves, A. A. (2012). Simple heuristics for the assembly line worker assignment and balancing problem. *Journal of Heuristics, 18*, 505–524.
Mutlu, O., Polat, O., & Supciller, A. A. (2013). An iterative genetic algorithm for the assembly line worker assignment and balancing problem of type-II. *Computers & Operations Research, 40*, 418–426.
Nkasu, M., & Leung, K. (1995). A stochastic approach to assembly line balancing. *International Journal of Production Research, 33*, 975–991.
Otto, A., & Otto, C. (2014). How to design effective priority rules: Example of simple assembly line balancing. *Computers & Industrial Engineering, 69*, 43–52.
Otto, A., Otto, C., & Scholl, A. (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*, 33–45.
Özcan, U. (2010). Balancing stochastic two-sided assembly lines: A chance-constrained, piecewise-linear, mixed integer program and a simulated annealing algorithm. *European Journal of Operational Research, 205*, 81–97.
Özcan, U., Kellegöz, T., & Toklu, B. (2011). A genetic algorithm for the stochastic mixed-model u-line balancing and sequencing problem. *International Journal of Production Research, 49*, 1605–1626.
Patterson, J. H., & Albracht, J. J. (1975). Assembly-line balancing: Zero-one programming with Fibonacci search. *Operations Research, 23*, 166–172.
Ritt, M., & Costa, A. M. (2011). A comparison of formulations for the simple assembly line balancing problem. *Computing Research Repository*, Available from abs/1111.0934 <http://arxiv.org/abs/1111.0934>.
Scholl, A. (1999). *Balancing and sequencing of assembly lines* (2nd ed.). Heidelberg: Physica.
Scholl, A., & Becker, C. (2006). State-of-the-art exact and heuristic solution procedures for simple assembly line balancing. *European Journal of Operational Research, 168*, 666–693.
Scholl, A., & Voss, S. (1996). Simple assembly line balancing-heuristic approaches. *Journal of Heuristics, 2*, 217–244.
Sivasankaran, P., & Shahabudeen, P. (2014). Literature review of assembly line balancing problems. *International Journal of Advanced Manufacturing Technology, 73*, 1665–1694.
Solyalı, O., Cordeau, J.-F., & Laporte, G. (2012). Robust inventory routing under demand uncertainty. *Transportation Science, 46*, 327–340.
Sotskov, Y., Dolgui, A., & Portmann, M. (2006). Stability analysis of an optimal balance for an assembly line with fixed cycle time. *European Journal of Operational Research, 168*, 783–797.
Suresh, G., & Sahu, S. (1994). Stochastic assembly line balancing using simulated annealing. *International Journal of Production Research, 32*, 1801–1810.
Toksarı, M., İşleyen, S., Güner, E., & Baykoç, O. (2010). Assembly line balancing problem with deterioration tasks and learning effect. *Expert Systems with Applications, 37*, 1223–1228.
Vilà, M., & Pereira, J. (2014). A branch-and-bound algorithm for assembly line worker assignment and balancing problems. *Computers & Operations Research, 44*, 105–114.
Zacharia, P., & Nearchou, A. (2012). Multi-objective fuzzy assembly line balancing using genetic algorithms. *Journal of Intelligent Manufacturing, 23*, 615–627.