

International Journal of Production Research

ISSN: 0020-7543 (Print) 1366-588X (Online) Journal homepage: www.tandfonline.com/journals/tprs20

The assembly line worker assignment and balancing problem with stochastic worker availability

Marcus Ritt, Alysson M. Costa & Cristóbal Miralles

To cite this article: Marcus Ritt, Alysson M. Costa & Cristóbal Miralles (2016) The assembly line worker assignment and balancing problem with stochastic worker availability, International Journal of Production Research, 54:3, 907-922, DOI: 10.1080/00207543.2015.1108534

To link to this article: https://doi.org/10.1080/00207543.2015.1108534

View supplementary material

Published online: 26 Nov 2015.

Submit your article to this journal

Article views: 967

View related articles

View Crossmark data

Citing articles: 23 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tprs20

International Journal of Production Research, 2016
Vol. 54, No. 3, 907–922, http://dx.doi.org/10.1080/00207543.2015.1108534

Taylor & Francis
Taylor & Francis Group

The assembly line worker assignment and balancing problem with stochastic worker availability

Marcus Ritt[a]*, Alysson M. Costa[b] and Cristóbal Miralles[c]

[a]*Instituto de Informática, Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil;* [b]*School of Mathematics and Statistics, University of Melbourne, Melbourne, Australia;* [c]*ROGLE, Departamento de Organización de Empresas, Universitat Politècnica de València, València, Spain*

(*Received 27 May 2015; accepted 6 October 2015*)

Assembly lines can be employed successfully in sheltered work centres to better include persons with disabilities in the labour market as well as to improve production efficiency. The optimal assignment of a heterogeneous workforce is known as the assembly line worker assignment and balancing problem (ALWABP). These assembly lines are characterised not only by a heterogeneous workforce, but also by high levels of absenteeism, which makes it more difficult to obtain stable and efficient line balancing solutions. In this paper, an extension of the ALWABP to minimise the expected cycle time under uncertain worker availability is proposed. We model this problem as a two-stage mixed integer program, and propose local search heuristics for solving it. Computational experiments show that stochastic modelling can help to improve the line's efficiency and that the proposed heuristics produce good results for instances of practical size.

Keywords: assembly line balancing; mixed integer linear programming; stochastic programming; heterogeneous workforce; simulated annealing

1. Introduction

People with disabilities represent about 10% of the world population and a common phenomenon in most countries is that these persons suffer much more difficulties in finding and maintaining a job. This is significant even in those countries with more social progress. For example, a recent research in the UK has shown that employment rates for persons with disabilities are very low, with only around 48% of them employed compared to 78% of adults without disabilities. Even when they are employed, they are more likely to work only part-time (ODI 2011).

Independent of the cultural differences among countries, it can be said that work is the most effective means of escaping the vicious circle of marginalisation, poverty and social exclusion. Persons with disabilities are frequently trapped in this vicious circle, and positive action is needed to assist them in breaking out of it. Barriers which persons with disabilities face in getting jobs and taking their place in society should be overcome through a variety of regulations, programs and institutional policies. To this effect, one of the formulas applied in many countries has been the creation of sheltered work centres for disabled (SWD). These companies receive some institutional support for maintaining a high percentage of workers with disabilities and have the duty of respecting their limitations and aiming to achieve a positive evolution of their capabilities and capacities. Apart from this support, these companies compete in the real market and thus have to be efficient. This work formula, despite its intrinsic managerial complexity, has been successful in decreasing the former unemployment rates of many countries. Some case studies in Spain were the origin of the so-called *assembly line worker assignment and balancing problem* (ALWABP).

1.1 *The ALWABP and the need to consider stochasticity*

The ALWABP was initially defined by Miralles et al. (2007), highlighting the advantages of assembly lines in the presence of a heterogeneous workforce, and opening a trend within the OR/MS approaches that traditionally have focused on homogeneous standardised human resources. Indeed, most of the literature on assembly lines consider a homogeneous workforce in which all workers are able to execute all tasks at the same speed, configuring the *simple assembly line balancing problem* (SALBP). Classical and recent reviews on this problem and its extensions can be found in Baybars (1986); Scholl (1999); Scholl and Becker (2006); Becker and Scholl (2006); Boysen, Fliedner, and Scholl (2007); Boysen, Fliedner, and Scholl (2008); Battaïa and Dolgui (2013); Sivasankaran and Shahabudeen (2014) and mainly focus on two objective functions known as type 1 (where the number of stations must be minimised given a desired productivity) and type 2 (in which productivity is maximised given a fixed number of stations).

*Corresponding author. Email: marcus.ritt@inf.ufrgs.br

© 2015 Taylor & Francis

In this spirit, the subarea of assembly line balancing has been the pioneer in extending former models and approaches to situations where heterogeneity needs to be considered, thus contributing to narrow the gap between research and practice. In the case of the ALWABP, the focus is the heterogeneity of task times and the presence of incompatibilities which are mainly related to the inability of some workers to execute a subset of the tasks or to be assigned to a subset of the workstations. As mentioned above, the problem is strongly motivated by the diverse paces and capabilities of workers with disabilities in assembly lines in SWDs. These characteristics define a new set of realistic hypotheses originally inspired by assembly lines in SWDs. Thus, from the initial paper (Miralles et al. 2007), many other references have contributed to give this problem visibility throughout the academic area, proposing different extensions and methods to solve it (see for e.g. Miralles et al. 2008; Blum and Miralles 2011; Araújo, Costa, and Miralles 2012; Moreira et al. 2012; Mutlu, Polat, and Ayca 2013; Borba and Ritt 2014; Vila and Pereira 2014; Cortez and Costa 2015; Moreira, Costa, and Miralles 2015). Most of this research effort considers the ALWABP of type 2, due to the characteristics of the motivating context (SWDs), which aim to employ as many persons with disabilities as possible while maximising productivity.

Despite these recent efforts, many misleading stereotypes on the work capabilities of workers with disabilities persist and questions on how to handle heterogeneous scenarios of SWDs still remain. Indeed, most of the references in the OR/MS literature still model production situations where heterogeneity is not considered. Furthermore, the decisions are supposed to be taken based on a one-only scenario where the aim is to find a certain optimal solution. However, even though having this optimal solution as a reference is important, its average quality in face of uncertainty must also be considered. This can be achieved by considering all the possible scenarios that are likely to happen. If, as has been stated, consideration of uncertainty is an important issue in ordinary environments, it becomes mandatory in SWDs due to some characteristics that have been summarised in Miralles et al. (2007):

- The primary objective of an SWD is to promote a work environment that helps workers with disabilities to have a positive and constant evolution of their own capabilities, in order to integrate them as soon as possible in ordinary companies. So, it is usual that many workers leave the SWD when they reach their best yields. The SWD must then replace them with new workers, which make a reassignment according to the abilities necessary.
- Absenteeism is also very common in this environment. One of the reasons is that workers with disabilities generally have more health problems than usual.
- Also, periodic psychological support and control is mandatory in SWDs, and requires that the workers leave the workplace frequently.

These circumstances justify the design of agile resolution procedures for finding not only the best solution for a single situation, but the solution attending all the possible scenarios of the available workforce. In those cases where moving the tasks along the stations is difficult (due to physical or managerial constraints), it becomes crucial to know the most stable assignment of tasks to workstations in face of uncertainty. This 'best average solution' must take into account the most probable scenarios of available workers which will be later dynamically assigned to the workplaces depending on operative circumstances like the ones described above.

1.2 *Aim and structure of the paper*

We introduce the *assembly line worker assignment and balancing problem with stochastic worker availability* (SALWABP). The goal of this problem is to balance assembly lines in the presence of an unknown set of heterogeneous workers. We model this problem as a two-stage mixed integer stochastic problem in which task assignments to workstations are first-stage decisions and worker assignments to workstations are second-stage decisions. The motivation for our study is the situation faced by assembly line managers in SWDs which have high levels of absenteeism. The proposed models can be also applied in normal production environments that suffer from absenteeism, and where workers have different skills, e.g. due to different training levels.

The following section presents a literature review on stochastic assembly line balancing. Section 3 introduces the problem formally, and heuristic solution procedures are proposed in Section 4. Section 5 presents a set of computational experiments that have been made with the goal of evaluating the proposed models and heuristics. Section 6 ends this paper with some conclusions and paths for future investigations.

2. Literature review

Several authors have considered stochastic models of assembly lines. When task times vary, the assembly line has to foresee remedial actions to be taken when the cycle time is exceeded at some station. In such a case, the unfinished products can

be repaired later (for example, by a mobile repair team, at certain offline repair stations placed along the line, or at the end of the line), or the assembly line can be stopped. Such actions incur additional *incompletion costs* and, therefore, research questions related to the modelling, evaluation and minimisation of such costs arise (Kottas and Lau 1976; Silverman and Carter 1986; Lyu 1997). Since even most basic versions of this task assignment problem are NP-complete, the literature has mainly focused on heuristic methods, which try to obtain good solutions in an acceptable computation time. The main methods include constructive, station-oriented assignment procedures, sometimes followed by a local search (Moodie and Young 1965; Kottas and Lau 1973; Kottas and Lau 1981; Carter and Silverman 1984; Shtub 1984; Chakravarty and Shtub 1986; Shin and Min 1991; McMullen and Frazier 1997; Gamberini et al. 2004; Liu, Ong, and Huang 2005; Chiang and Urban 2006), ant colony optimization (Agarwal and Tiwari 2008), genetic algorithms (Baykasoğlu and Özbakir 2007; Gamberini et al. 2009; Özcan, Kellegz, and Toklu 2011), simulated annealing (Suresh and Sahu 1994), and exact methods like dynamic programming and branch-and-bound (Kao 1976; Henig 1986; Wilson 1986; Carraway 1989), or truncated versions thereof (e.g. beam search) (Sarin and Erel 1990; Sarin, Erel, and Dar-El 1999; Erel, Sabuncuoglu, and Sekerci 2005). A brief overview of some other approaches can be found in Becker and Scholl (2006).

A few authors are concerned with a non-standard workforce. Downey and Leonard (1992), for example, give a method that flexibly organises the work when there are less workers than stations. Such a method may be a natural extension of the problem studied here, since it would allow to man the line even if less than the planned number of workers is available. Inman, Jordan, and Blumenfeld (2004) deal with absenteeism in a different fashion, by training workers to be able to assume other tasks in case a colleague is absent. A chaining structure in which each worker is able to perform the tasks of a single colleague is used to reduce training costs. To the best of our knowledge, there is no study in the literature that directly addresses the problem of a varying workforce. The reason is probably that this problem only arises if the workforce is heterogeneous.

The study presented here differs from the literature in many aspects. On the one hand, we are not directly concerned with varying task times of a single worker. Instead the task time uncertainty we consider is an indirect result of an unknown availability of heterogeneous workers. This allows us to adjust the cycle time according to an optimal assignment of the present workers (once they are known) to the stations. Besides being motivated by the practical problem faced by SWDs, there seems to be an interest in such a study since there are few articles that deal with stochastic problems in which the scenarios contain binary parameters. On the other hand, due to technical constraints, the assignment of tasks must be (partially) determined on a tactical level, giving this problem a clear two stage with recourse structure. In the following section, this problem is formalised.

3. Formal definition and models

Let P be a set of stations, W be a set of workers, $|W| > |P|$, $(N, \leq)$ be a partially ordered set of tasks, and $F_i = \{j \in N \mid i \leq j, i \neq j\}$ be the set of tasks that succeed task i in $(N, \leq)$. We assume $P = \{1, 2, \ldots, n\}$ and that the stations are placed in increasing order along the assembly line. Associated to each pair $(w, i) \in W \times N$ there is an integral value t_{wi} that represents the time needed by worker w to execute task i. Some workers may be unable to execute a task. In these cases the corresponding task execution time is set to ∞. We define $W_i = \{w \in W \mid t_{wi} < \infty\}$ as the set of workers that are able to execute task i. Due to absenteeism, at each given day only $|P|$ of the $|W|$ workers are available. The goal of the SALWABP is to find an assignment of tasks to stations such that the expected cycle time is minimised for the available workers. We propose two mixed integer programs. In the first 'fixed-tasks' model, all tasks are fixed to stations. The second model is a generalisation of the first in which some tasks can be reassigned at the beginning of each work day.

3.1 *A fixed-tasks model*

In this problem, the tasks must be assigned to stations in a first-stage optimisation and when the set of random variables, i.e. the available workers, is observed, the workers are assigned to the stations. This can be modelled as a two-stage stochastic problem with recourse. Let $S \subseteq \mathcal{P}(W)$ be the set of possible scenarios and p^s be the occurrence probability of scenario $s \in S$. Each scenario corresponds to a set of available workers. Let $\overline{y}^s_w$ be a parameter that equals one if worker w is available in scenario s. Defining the variables

x_{pi}, binary first-stage variable, which equals one iff task $i \in N$ is assigned to station $p \in P$;
y^s_{pw}, binary recourse variable, which equals one iff worker $w \in W$ is assigned to station $p \in P$ in scenario $s \in S$;
C^s, cycle time in scenario $s \in S$,

the SALWABP with fixed tasks (SALWABP-Fi) can be modelled as

$$\text{min.} \quad \sum_{s \in S} p^s C^s, \tag{1}$$

$$\text{s.t.} \quad \sum_{p\in P} x_{pi} = 1, \quad \forall i \in N, \tag{2}$$

$$\sum_{p\in P|p\le k} x_{pi} \ge \sum_{p\in P|p\le k} x_{pj}, \quad \forall\, i \in N,\, j \in F_i, k \in P, \tag{3}$$

$$\sum_{i\in N} x_{pi} t_{wi} \le C^s + M_w(1 - y^s_{pw}), \quad \forall p \in P, w \in W, s \in S, \tag{4}$$

$$\sum_{w\in W} y^s_{pw} = 1, \quad \forall p \in P, s \in S, \tag{5}$$

$$\sum_{p\in P} y^s_{pw} \le \overline{y}^s_w, \quad \forall w \in W, s \in S, \tag{6}$$

$$y^s_{pw} + x_{pi} \le 1, \quad \forall i \in N, w \in W\backslash W_i, s \in S, p \in P. \tag{7}$$

The objective function (1) minimises the expected cycle time. Constraints (2) force all tasks to be assigned to some workstation, while constraints (3) force this assignment to respect the precedence constraints. Constraints (4) establish that all stations must respect the cycle time. In these constraints, M_w is a large enough constant to deactivate the constraint for a worker–station pair (w, p) such that worker w is not assigned to station p. It can be set to $M_w = \sum_{i\in N|w\in W_i} t_{wi}$, for example. Finally, constraints (5) and (6) assign an available worker to a single station in each scenario while constraints (7) prohibit to assign a worker to a station that has tasks they are unable to execute.

3.2 *A flexible-tasks model*

The SALWABP-Fi assumes that all tasks must be assigned in the first stage. This assumption might be too restrictive if there are tasks that are sufficiently flexible (requiring no large or fixed equipment, for instance) and may, therefore, be assigned after the available workers are known. Assuming that N_1 is the set of tasks that need to be assigned in the first stage (fixed tasks) and $N_2 = N\backslash N_1$ is the set of remaining tasks (flexible tasks) and with the redefinition of task assignment variables as

x_{pi}, binary first-stage variable, which equals one iff task $i \in N_1$ is assigned to station $p \in P$;
x^s_{pi}, recourse variable, which equals one iff task $i \in N_2$ is assigned to station $p \in P$ in scenario $s \in S$;

the SALWABP with flexible tasks (SALWABP-Fl) can be modelled as:

$$\text{min.} \quad \sum_{s\in S} p^s C^s \tag{8}$$

$$\text{s.t.} \quad \sum_{p\in P} x_{pi} = 1, \quad \forall i \in N_1, \tag{9}$$

$$\sum_{p\in P} x^s_{pi} = 1, \quad \forall i \in N_2, \forall s \in S \tag{10}$$

$$\sum_{p\in P|p\le k} x_{pi} \ge \sum_{p\in P|p\le k} x_{pj}, \quad \forall\, i \in N_1,\, j \in N_1 \cap F_i, k \in P, \tag{11}$$

$$\sum_{p\in P|p\le k} x^s_{pi} \ge \sum_{p\in P|p\le k} x^s_{pj}, \quad \forall\, i \in N_2,\, j \in N_2 \cap F_i, k \in P, s \in S \tag{12}$$

$$\sum_{p\in P|p\le k} x_{pi} \ge \sum_{p\in P|p\le k} x^s_{pj}, \quad \forall\, i \in N_1,\, j \in N_2 \cap F_i, k \in P, s \in S \tag{13}$$

$$\sum_{p\in P|p\le k} x^s_{pi} \ge \sum_{p\in P|p\le k} x_{pj}, \quad \forall\, i \in N_2,\, j \in N_1 \cap F_i, k \in P, s \in S \tag{14}$$

$$\sum_{i\in N_1} x_{pi} t_{wi} + \sum_{i\in N_2} x^s_{pi} t_{wi} \le C^s + M_w(1 - y^s_{pw}), \quad \forall p \in P, w \in W, s \in S, \tag{15}$$

$$\sum_{w\in W} y^s_{pw} = 1, \quad \forall p \in P, s \in S, \tag{16}$$

$$\sum_{p\in P} y^s_{pw} \le \overline{y}^s_w, \quad \forall w \in W, s \in S, \tag{17}$$

$$y^s_{pw} + x_{pi} \leq 1, \quad \forall i \in N_1, w \in W \backslash W_i, s \in S, p \in P \tag{18}$$

$$y^s_{pw} + x^s_{pi} \leq 1, \quad \forall i \in N_2, w \in W \backslash W_i, s \in S, p \in P. \tag{19}$$

The objective function (8) again minimises the expected cycle time. Constraints (9) force all fixed tasks to be assigned in the first stage while constraints (11) force this assignment to respect the precedence constraints. Constraints (10) and (12) are the equivalent of these constraints for flexible tasks, while constraints (13) and (14) guarantee that the assignment of flexible tasks in each scenario respects the precedence constraints with respect to the fixed tasks. Constraints (15)–(17) are equivalent to (4)–(6) and guarantee that the cycle time is respected and that an available worker is assigned to each station. Finally, (18) and (19) forbid a worker to be assigned to tasks they are not able to execute.

4. Heuristic assignment procedures

In this section, we propose fast and simple heuristics for finding a task assignment with a good expected cycle time. The heuristics consist of three main elements: a constructive procedure for finding an initial task assignment, a procedure for rapidly finding an optimal worker assignment for a given task assignment, and a local search procedure based on simulated annealing for improving the first-stage task assignment (for both models) as well as for finding a good recourse action (in the case of the flexible-tasks model). We describe these three elements next.

4.1 *Finding an initial task assignment*

For finding an initial task assignment, we simplify the problem to an instance of the simple assembly line balancing problem of type 2 (SALBP-2), by attributing to each task the minimal task time $t_i^- = \min\{t_{wi} \mid w \in W\}$ over all workers. The SALBP-2 is to find an assignment of tasks to the workstations, that satisfies the precedence constraints, and that minimises the cycle time. We heuristically solve the SALBP-2 instance by a standard lower bound search, as shown in Algorithm 1. It applies, for each cycle time, a station-oriented assignment procedure for finding a feasible task assignment (Scholl and Voß 1996). In every step, we choose among the available tasks a task $i \in N$ with the largest number of followers $|F_i|$ to be assigned to the current station. Ties are broken by maximum task time, and then by the maximum task number.

Algorithm 1. Station-oriented assignment procedure for finding an initial task assignment

```
function InitialTaskAssignment(N,P)
    for C = c_, c_ + 1, ..., c¯ do                          ▷ lower bound search
        U := N; X := ∅
        k := 1; S_k := 0
        while U ≠ ∅ do
            select an available task i ∈ U such that |F_i| is maximum
            if S_k + t_i^- > C then                          ▷ open a new station
                k := k + 1; S_k := 0
                if k > |P| then                              ▷ all stations have been used
                    continue with the next cycle time C
            X := X ∪ {i ↦ k}                                 ▷ assign task i station k
            S_k := S_k + t_i^-
            U := U \ {i}
        return X                                             ▷ return feasible assignment
```

4.2 *Computing an optimal worker assignment*

To evaluate a given task assignment we have to determine its expected cycle time. In both models this requires to compute the minimal cycle time in all scenarios. In the fixed-tasks model, the cycle time of a scenario is the cycle time of the optimal assignment of a given set of workers W to the stations P. In the case of the flexible-tasks model, the cycle time also depends on an optimal assignment of the flexible tasks, which will be addressed below.

The worker assignment problem can be solved as follows. We extend the set of stations by $|W| - |P|$ *dummy stations*, which will receive the workers who will not be assigned to a real station. Next, we determine for each pair of station $p \in P$

and worker $w \in W$ the station time S_{wp} that results when assigning w to p. The station time is defined as the total execution time of all tasks assigned to a station for a given worker w. For stations that have been assigned with tasks that are infeasible for w, the station time is defined as ∞. For an assignment of a worker to a dummy station the station time is defined as 0.

We can now define a bipartite graph on the extended set of stations and the set of workers, with an edge of weight S_{wp} between w and p. The minimum cycle time for the given set of workers and stations with tasks assigned to them corresponds to a minimum bottleneck perfect matching (Fulkerson, Glicksberg, and Gross 1953) in that bipartite graph. Such a matching can be found by binary search of an interval of admissible cycle times $[\underline{c}, \overline{c}]$. Here, $\underline{c}$ is a lower bound and $\overline{c}$ an upper bound on the cycle time. They can be set, for example, to 0 and $\sum_{i \in N} \max_{w \in W_i} t_{iw}$, respectively. For each candidate cycle time c we exclude all edges of weight more than c and find a maximum cardinality matching on the remaining graph, as shown in Algorithm 2. A maximum cardinality matching can be found in time $O(\sqrt{n}m)$ for a graph with n vertices and m edges using the algorithm of Hopcroft and Karp (1973). Therefore, in our case, the minimum cycle time can be found in time $O(\log \overline{c}\ |W|^{5/2})$, since the graph has $n = |W|$ workers, at most $m = |W|^2$ edges, and a binary search for the optimal cycle time needs at most $\log \overline{c}$ steps. Since the number of workers usually is small, this approach is very efficient in practice, as will be shown below.

Algorithm 2. Finding the best worker assignment for a given task assignment $X : N \to P$.

function BESTWORKERASSIGNMENT(N, P, W, X)
  if $|W| > |P|$ **then**
    add $|W| - |P|$ dummy stations to P
  compute station times $S_{wp} = \sum_{i \in N | X(i) = p} t_{wi}$ for all $w \in W$, $p \in P$
  while $\underline{c} \leq \overline{c}$ **do** ▷ binary search for best cycle time
    $c := \lfloor (\underline{c} + \overline{c})/2 \rfloor$
    let $G_c = (W \dot\cup P, A_c)$ with $A_c = \{(w, p) \mid w \in W, p \in P, S_{tw} \leq c\}$
    find a maximum cardinality matching M_c in G_c
    if $|M_c| = |W|$ **then** ▷ candidate cycle time feasible?
      $\overline{c} := c$
    else
      $\underline{c} := c + 1$
  return (M_c, c)

4.3 *Improving task assignments*

Simulated annealing is a local search heuristic which maintains a current solution and repeatedly applies a random local modification (called a *move*) to it. The new solution is accepted if it improves the objective function value, or, if the solution worsens by Δ, with probability $\exp(-\Delta / T)$. The parameter T is called the *temperature*. In the limit of very high temperatures simulated annealing corresponds to a random walk in the solution space; in the limit of $T \to 0$ it reduces to a local search, which randomly chooses an improving move. For this reason, the temperature usually follows a *cooling schedule*, which starts from an initially high temperature, holds it constant for a number L of moves, and successively reduces it, until some stopping criterion is satisfied.

The local search we propose considers two types of moves: *shifting* a task from its current station to another station, and *swapping* two tasks assigned to two different stations. The proposed simulated annealing heuristic has three phases. In the first phase, the initial task assignment (see Section 4.1) is improved by a best improvement local search considering shift and swap moves. In the second phase, it determines an adequate initial temperature. This is accomplished by a trial run, that starts from a very low initial temperature T_0, and doubles the temperature until the percentage of accepted moves at the current temperature exceeds a desired percentage p_i. The third phase uses a geometric cooling schedule, that reduces the temperature each L steps from the current value T to $T' = T \times r$, for a fixed *cooling rate* $r < 1$. The third phase stops, if in the last five temperature levels the incumbent did not improve, and the number of accepted moves was less than a minimal percentage p_m. Algorithm 3 summarises the whole procedure.

4.4 *Solving the flexible-tasks model*

To solve the flexible-tasks model, we use the same general approach as described above. The optimisation for the flexible-tasks model starts from the best solution obtained for the fixed-tasks model, and applies a temperature calibration phase

Algorithm 3. Simulated annealing heuristic to find improved task assignments

function SIMULATEDANNEALING(N, P, W)
  X := INITIALTASKASSIGNMENT(N, P)
  (M, c) := BESTWORKERASSIGNMENT(W, P, X)
  X := localsearch(X, M) ▷ Phase 1: local search
  $T := T_0/2$
  repeat ▷ Phase 2: calibration
    $T := 2T$
    (X, M, c) := METROPOLIS(X, M, L)
  until percentage of accepted moves exceeds p_i
  repeat ▷ Phase 3: annealing
    (X, M, c) := METROPOLIS(X, M, L)
    $T := T \times r$
  until not improved in last 5 iterations and percentage of accepted moves less than p_m
  return best solution (X, M, c) found during the search

function METROPOLIS(N, P, W, X, M, L)
  for L iterations **do**
    apply a random feasible move to X to obtain X'
    (M', c') := BESTWORKERASSIGNMENT(N, P, W, X')
    $(X, M, c) := (X', M', c')$ with probability $\min\{e^{-(c'-c))/T}, 1\}$ ▷ accept move probabilistically
  return (X, M, c)

followed by an optimisation phase, as for the fixed-tasks model. To evaluate a solution, we have to find a good recourse action, that is able to reduce the cycle time. Therefore, before evaluating a scenario, we apply a best improvement local search to find a better assignment of the tasks in N_2. Observe that this best improvement is embedded in an outer simulated annealing heuristic, and is applied during the evaluation of a solution once for each scenario. The best improvement local search uses the same shift and swap moves as the simulated annealing heuristic.

5. Computational experiments

5.1 *Instances*

We have evaluated the performance of the mixed-integer models and the heuristics on a set of instances derived from SALBP instances proposed by Otto, Otto, and Scholl (2013). From these SALBP instances, we first selected the 525 instances with 20 tasks. For each instance we defined the number of workers as the upper bound on the number of stations as given by Otto, Otto, and Scholl (2013). Then, we selected from the 525 instances 184 instances with four to eight workers, which best represent the typical workforce size in SWDs. Finally, we generated four ALWABP instances from each instance, as follows. The task times for each worker were selected from the uniform distribution with intervals $[t, 2t]$ or $[t, 5t]$, where t is the original task time. Additionally, each worker has 10 or 20% randomly selected, infeasible tasks. The instances do not include any worker–stations incompatibilities since they are more rare in the context of SWDs even though they could be easily treated by the model. We verified that all resulting 736 instances have at least one feasible solution for the ALWABP. The instances are available at Costa (2013). For the stochastic problem, we consider all scenarios where a single worker is absent, which is the most common case. Note that we consider the absence of more than one worker too erratic to justify the application of our proposal. If the size of the available workforce varies considerably, an a priori assignment of tasks to stations is less useful in practice. Moreover, it would also be difficult to compare results and to even estimate by the manager all the double or triple combinations of absenteeism and their probabilities of occurrence as input. Each scenario has the same probability. For the flexible-tasks model, we have chosen to make 50% of the tasks flexible, to be able to compare a static to a highly flexible scenario. The flexible tasks have been defined arbitrarily to be the even-numbered tasks, in order of appearance in the description of the instance.

5.2 *Theoretical and practical evaluation of solutions*

In order to evaluate the effect of incorporating stochastic considerations in the model, two metrics of uncertainty effects are used, namely, *the expected value of perfect information* (EVPI) and *the value of the stochastic solution* (VSS). Both

measures are compared to the optimal objective function value of the stochastic model, namely the RP value. The EVPI indicates the maximum amount the decision-maker is willing to pay in order to obtain the value of a random variable before making his decision (Avriel and Williams 1970; Birge and Louveaux 1997). It is computed by evaluating the recourse model and the 'wait-and-see' solutions, as follows: (1) solve one problem for each scenario s and obtain the so-called wait-and-see solutions, where the first-stage decisions are taken to optimise a single scenario; (2) calculate the expectation over the set of scenarios of the wait-and-see solutions, WS; (3) compute the estimated value of perfect information as $\mathrm{EVPI} = \mathrm{RP} - \mathrm{WS}$ (in the case of minimization).

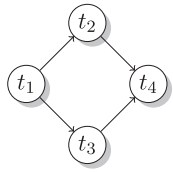
The measure VSS represents the added value of using a stochastic model (Birge 1995). It compares the solution of the recourse model to the value of a deterministic solution, which in real-valued scenarios is usually obtained by replacing all random variables by their expectations. This approach is not possible in our problem, since the scenarios are defined by binary decisions. We therefore compute the VSS as follows: (1) obtain the first-stage variables for all individual scenarios and compute the expected value for each first-stage solution; (2) use the best solution (in terms of expected value) as the estimated value BEV; (3) determine $\mathrm{VSS} = \mathrm{BEV} - \mathrm{RP}$. If $\mathrm{VSS} \leq \epsilon$ (where ϵ is a given tolerance), then the stochastic model can be approximated by the used scenario. Otherwise, there is a profit in using the stochastic model instead of considering a single scenario. This is, somehow, a best-case perspective for the deterministic solution. The same measure can be computed using the solution of the scenario that generates the minimum total revenue, WEV – in a worst case perspective. For both measures EVPI and VSS, we can also define relative statistics by $\mathrm{VSSr} = (\mathrm{BEV} - \mathrm{RP})/\mathrm{RP}$ and $\mathrm{EVPIr} = (\mathrm{RP} - \mathrm{WS})/\mathrm{RP}$.

The practical insights of the two proposed indicators, EVPI and VSS, can be further explained through the following example: imagine a factory section where the manager has to cope with the line balancing in a high absenteeism environment (or simply a dynamic environment due to other sections' eventual demands of resources) that makes it quite normal that only five out of the six heterogeneous workers of the section are usually available. In that case, every morning the manager would need to rebalance the line in order to get the best assignment with the actual set of five workers (or with some five selected workers, in case all six workers attended the factory).

We first discuss the EVPI. With the absence of other approaches, every day the line would be balanced with the best local solution with the corresponding five workers of that day; finally arriving to an average cycle time equal to the 'Wait and see' solution WS. With this scheme, the manager would clearly need some set-up time to rearrange the line every morning but, in any case, if we discard these set-ups, the resulting cycle time would be WS. Let us say this mean cycle time 'WS' for our manager is 74 s. On the contrary, if the manager calculates the best static assignment of tasks to the five stations using the fixed-tasks model, then the expected cycle time (initially assuming equal probability of absence for the six workers, although this could be fine-tuned in case the historical data recommends it) would be 'RP'. Let us say that with this 'less efficient but static' balance, we get an average cycle time 'RP' of 83 s. Thus, the estimated value of perfect information $\mathrm{EVPI} = \mathrm{RP} - \mathrm{WS} = 83 - 74 = 9$ s of cycle time; that can be easily translated into a $\mathrm{EVPIr} = (\mathrm{RP} - \mathrm{WS})/\mathrm{RP} = 10.84\%$ which, in a labour day of eight standard hours, would mean a 'daily time loss' of 52 min. Therefore, in a strict quantitative sense we can say that: when the average daily set-up of the line needs more than 52 min, the fixed-tasks model proposed is preferable; additionally offering several qualitative advantages due to its stability that may suggest its application even for lower set-up times. In any case, in a general sense, EVPI is a perfect indicator to analyse the productivity loss when no rebalancing is done.

In the case of the flexible-tasks model a kind of partial rebalancing is permitted, having some other managerial implications. Within this particular approach, a first intuition of the manager could be, for example, to fix at the five stations only those tasks that imply longer set-ups, and leaving the assignment of the rest free to reconfigure every morning once the absenteeism scenario is known. Nevertheless, intuitions like this can clearly lead to suboptimal areas and models like the flexible-tasks model proposed would be necessary (further analysis of these insights will follow in the discussion of the results below).

We now turn to the VSS. In the same scenario described, a manager wishing to gain stability through less sophisticated methods could easily opt for the intuitive approach of exploring all the ALWABP local solution spaces. E.g. take workers 1–5, obtain the ALWABP optimal solution, and keep this task assignment along the rest of the absenteeism scenarios that will successively arise. Then do the same with the workers 2–6, for example, and so on. Once all the initial possible scenarios are reviewed, the manager finally adopts the assignment of tasks to stations with yields the best average cycle time. Let us say this fixed assignment provides a best expected value (BEV) of the cycle time of 90 s. Following our previous example where the average cycle time RP using the fixed-tasks model was 83, then we would have a $\mathrm{VSS} = \mathrm{BEV} - \mathrm{RP} = 90 - 83 = 7$ s, that would represent the added value of properly using the fixed-tasks model instead. It deserves to be highlighted here that, apart from the possible gains in cycle time, the manager would have real problems to simply obtain feasible solutions with his intuitive approach, because many incompatibilities would certainly arise when extending the initial assignments along the different scenarios. As can be appreciated in Table 2, approaches like the fixed- or flexible-tasks model become therefore necessary in order to simply get feasible solutions for all the scenarios (while maximising the overall productivity).



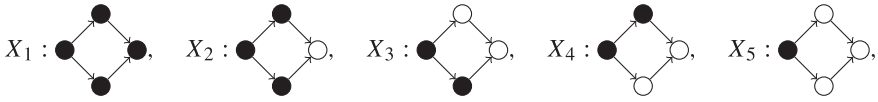
t_{wi}	t_1	t_2	t_3	t_4
w_1	1	1	∞	2
w_2	2	2	1	1
w_3	2	∞	2	∞

Figure 1. Example of an SALWABP instance. Left: precedence constraints among the tasks. Right: task execution times.

5.3 *A numerical example*

Consider the instance shown in Figure 1 with three workers and four tasks. We will assume in this example, as we do in the computational experiments below, that all scenarios where one worker is missing will occur with the same probability. This is a very common case, but note that the models and algorithms given can deal with any selection of scenarios, i.e. the choice of a useful set of scenarios remains a decision of the production manager. In this case, we have a set of three feasible scenarios $S = \{S_{12}, S_{13}, S_{23}\}$ where $S_{12} = \{w_1, w_2\}$, $S_{13} = \{w_1, w_3\}$, and $S_{23} = \{w_2, w_3\}$, and $p_{12} = p_{13} = p_{23} = 1/3$. The optimal wait-and-see solutions for the scenarios are given in Figure 2. We can see that the optimal assignment of tasks to stations, as well as the assignment of workers to stations changes as a response to the available workers. If we can afford to 'wait and see', the expected cycle time is $E[C] = \text{WS} = 1/3(2 + 4 + 4) \approx 3.3$.

We now turn to the solutions of the recourse problem. Here we must assign the tasks to the stations beforehand, and the recourse action will assign the workers. Assuming, without loss of generality, that task t_1 is assigned to station 1, the possible task assignments are



where a black dot represents an assignment to the first station, and a white dot an assignment to the second. It can be easily seen, that assignments X_1, X_2, X_4, and X_5 all have at least one infeasible scenario. For example, for task assignment X_4 there is no feasible recourse action in scenario S_{13}, since worker w_3 can execute neither tasks t_1 and t_2 nor tasks t_3 and t_4. Figure 3 shows the optimal recourse actions for task assignment X_3. The recourse actions for scenarios S_{13} and S_{23} are equal to the wait-and-see solutions, but in scenario S_{12} the best worker assignment leads to a cycle time of 3. Thus, the expected cycle time of the optimal solution to the recourse problem is $E[C] = \text{RP} = 1/3(3 + 4 + 4) \approx 3.7$. Therefore, we would have a $\text{EVPI} = \text{RP} - \text{WS} = 3.7 - 3.3 = 0.4$, what means an EVPIr = 10, 8%.

For this simple example, it can be easily checked that the incompatibilities of w_3 make very few solutions feasible when calculating the BEV. As it is also equal to 3.7, we have that $\text{VSS} = \text{BEV} - \text{RP} = 0$ in this case.

5.4 *Behaviour of the stochastic models in general*

In this section, we analyse the behaviour of the stochastic models. We first summarise the number of instances for which an optimal or feasible solution could be found, and then proceed to evaluate the quality of these solutions in terms of the stochastic measures EVPI and VSS.

All experiments have been run on a PC with an AMD FX-8150 processor running at 1.4 GHz and having 32 GB of main memory. The mathematical models have been solved with the commercial solver CPLEX 12.5. The solver was run with eight threads and a time limit of 15 min, i.e. an equivalent of 2 h of solving time. We have solved each individual scenario optimally, to determine the wait-and-see solution and the best and worst scenarios, as well as the recourse problems.

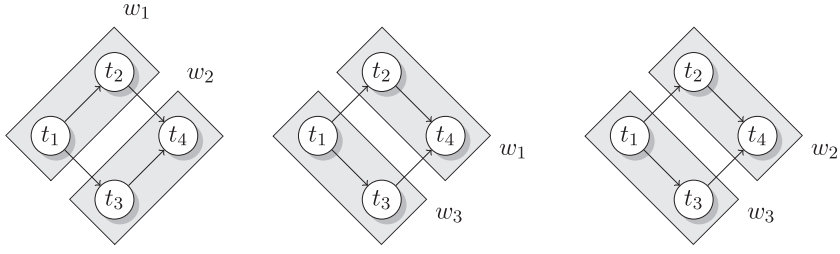


Figure 2. Wait-and-see solutions for scenarios S_{12}, S_{13}, and S_{23}. The assignment task to stations is shown in grey, the assignment of workers to station is indicated next to the stations. The corresponding cycle times are $C_{12} = 2$, $C_{13} = 4$, and $C_{23} = 4$.

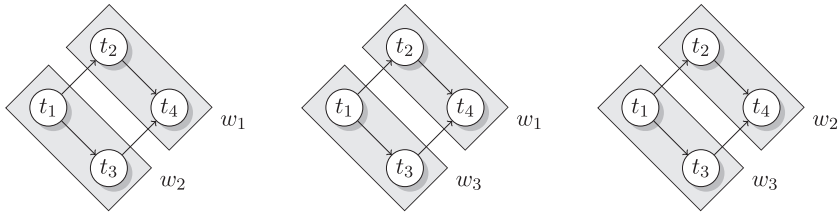


Figure 3. Optimal recourse action for task assignment X_4. The assignment of tasks to stations is shown in grey, the assignment of workers to station is indicated next to the stations. The corresponding cycle times are $C_{12} = 3$, $C_{13} = 4$ and $C_{23} = 4$.

Table 1. Total number and number of feasible instances, percentage of optimally solved recourse problems, and average solution time for the fixed- and flexible-tasks models.

					Fixed tasks		Flexible tasks	
$\|W\|$	Var.	Inc.	N	Feas.	Solved	t(s)	Solved	t(s)
4	$[t, 2t]$	10	39	37	100.0	0.1	100.0	0.5
		20	39	10	100.0	0.1	100.0	0.3
	$[t, 5t]$	10	39	36	100.0	0.1	100.0	0.3
		20	39	9	100.0	0.1	100.0	0.2
5	$[t, 2t]$	10	89	88	100.0	8.8	33.7	94.6
		20	89	81	100.0	1.3	97.8	13.5
	$[t, 5t]$	10	89	89	100.0	5.9	61.8	73.6
		20	89	74	100.0	1.7	98.9	9.7
6	$[t, 2t]$	10	49	49	8.2	870.3	0.0	900.0
		20	49	44	87.8	247.3	10.2	529.4
	$[t, 5t]$	10	49	49	22.4	765.6	0.0	868.2
		20	49	47	98.0	132.1	14.3	315.4
7	$[t, 2t]$	10	6	6	0.0	900.0	0.0	900.0
		20	6	6	0.0	900.0	0.0	900.0
	$[t, 5t]$	10	6	6	0.0	900.0	0.0	900.0
		20	6	6	0.0	900.0	0.0	900.0
8	$[t, 2t]$	10	1	1	0.0	900.0	0.0	900.0
		20	1	1	0.0	900.0	0.0	900.0
	$[t, 5t]$	10	1	1	0.0	900.0	0.0	900.0
		20	1	1	0.0	900.0	0.0	900.0
Sum/weighted avg.			736	641	84.0	170.6	58.2	231.4

5.4.1 *Solvability of the models*

We first look at the number of feasible solutions, and the number of recourse problems that could be solved. Table 1 groups the instances by the number of workers ('$|W|$'), the task time variability ('Var.') and the percentage of incompatibilities ('Inc.'). For each group, it reports the total number of instances ('N'), the number of instances that always have a feasible wait-and-see solution ('Feas.'), and two statistics for the fixed- and flexible-tasks models: the percentage of the total number of instances for which the recourse problem could be solved ('Solved'), and the average solution time ('t'), in seconds. All instances with a proven optimal or infeasible solution of the recourse problem are considered as solved. Problems which could not be solved contribute with the time limit of 900 s to the average.

The solver was able to find the optimal wait-and-see solution of all 736 instances in less than 2 min solving time per instance. We can see that in 641 of the 736 instances all scenarios have a feasible task assignment and therefore the instance has a feasible wait-and-see solution. In the remaining 95 instances, there is at least one scenario which has no feasible task assignment. This happens mainly in instances with few workers and a high number of incompatibilities. Since these instances have no feasible solution, they are excluded from the remaining analysis.

Looking at columns 'Solved', we can see that the models become more difficult to solve with an increasing number of workers, a low task time variability, and a low number of incompatibilities. The recourse problem could be solved optimally

Table 2. Percentage of feasible recourse problems, and percentages of feasible best and worst scenarios for the fixed- and flexible-tasks models.

				Fixed tasks			Flexible tasks		
$\|W\|$	Var.	Inc.	Feas.	RP	BEV	WEV	RP	BEV	WEV
4	$[t, 2t]$	10	37	89.2	29.7	2.7	100.0	86.5	27.0
		20	10	30.0	0.0	0.0	60.0	10.0	0.0
	$[t, 5t]$	10	36	94.4	41.7	5.6	94.4	86.1	27.8
		20	9	33.3	0.0	0.0	88.9	55.6	0.0
5	$[t, 2t]$	10	88	100.0	95.5	25.0	100.0	98.9	75.0
		20	81	76.5	19.8	0.0	95.1	75.3	7.4
	$[t, 5t]$	10	89	100.0	94.4	27.0	100.0	97.8	74.2
		20	74	79.7	23.0	0.0	94.6	81.1	12.2
6	$[t, 2t]$	10	49	100.0	100.0	65.3	100.0	100.0	87.8
		20	44	100.0	52.3	2.3	100.0	95.5	40.9
	$[t, 5t]$	10	49	100.0	100.0	73.5	100.0	100.0	87.8
		20	47	100.0	68.1	0.0	100.0	89.4	44.7
7	$[t, 2t]$	10	6	100.0	100.0	100.0	100.0	100.0	100.0
		20	6	100.0	66.7	33.3	100.0	100.0	83.3
	$[t, 5t]$	10	6	100.0	100.0	83.3	100.0	100.0	100.0
		20	6	100.0	100.0	33.3	100.0	100.0	100.0
8	$[t, 2t]$	10	1	100.0	100.0	100.0	100.0	100.0	100.0
		20	1	100.0	100.0	0.0	100.0	100.0	0.0
	$[t, 5t]$	10	1	100.0	100.0	100.0	100.0	100.0	100.0
		20	1	100.0	100.0	0.0	100.0	100.0	0.0
Sum/weighted avg.			641	91.7	63.3	21.1	97.7	89.5	49.5

for the instances with four and five workers and fixed tasks, and four workers with flexible tasks, but no instance with seven or more workers was solved optimally.

We now turn to the percentage of feasible solutions of the recourse problem, and the best and worst scenarios. Table 2 reports for each instance group the percentage of problems that have a feasible solution of the recourse problem (RP), a feasible best scenario (BEV) and a feasible worst scenario (WEV) for both models. As before the solver was able to find the optimal best and worst scenario in less than 2 min for all instances. Also, if the recourse problem had a feasible solution, then the solver always was able to find one, even if for some instances the solution is not optimal. Thus, the percentages of feasible solutions in Table 2 are exact values.

The percentages of feasible solutions increase from the worst scenario, over the best scenario, to the recourse problem, as expected. In the fixed-tasks model, only 21.1% of the worst scenarios are feasible, compared to 63.3% feasible best scenarios, and 91.7% feasible recourse problems. This shows that a task assignment based on a single scenario is not a good approach to find appropriate solutions in face of worker uncertainty, since it leads, in the best case, to an infeasible solution in about 30% of the instances, although a feasible solutions exists.

When tasks are flexible the differences are smaller, and the best scenario often leads to a feasible solution. This was expected, due to the greater room for manoeuvre with 50% of flexible tasks. Note that in the limit of 100% flexible tasks, all solutions would be equal to the wait-and-see solution. Still, the recourse problem leads to a feasible solution in 8% more of the instances. Also, the solution quality in the best scenario can be different, as discussed below. These findings confirm the efficacy of the recourse model in finding flexible first-stage solutions.

5.4.2 *Quality of the solutions*

The quality of the solutions found by solving the models is summarised in Table 3. As above, the instances are grouped by the number of workers ('$|W|$'), the time variability ('Var.') and the percentage of incompatibilities ('Inc.'). We report the EVPIr of the solution of the recourse problem and the VSSr for the best scenario for both the fixed- and the flexible-tasks model. The EVPIr is the average over all instances where a feasible solution to the recourse problem exists, and the VSSr is the average over all instances with a feasible best scenario. For the two groups with an entry of '–', the best scenario for the fixed-tasks models was infeasible for all instances in the group.

Table 3. Relative EVPI and relative VSS for the fixed- and flexible-tasks models.

				Fixed tasks		Flexible tasks	
$\|W\|$	Var.	Inc.	N	EVPIr	VSSr	EVPIr	VSSr
4	$[t, 2t]$	10	39	7.9	3.5	3.9	1.6
		20	39	16.7	–	7.0	0.0
	$[t, 5t]$	10	39	14.9	5.8	7.8	2.7
		20	39	25.7	–	12.4	5.0
5	$[t, 2t]$	10	89	5.6	2.5	3.5	0.4
		20	89	12.3	3.0	6.1	2.6
	$[t, 5t]$	10	89	13.1	6.7	6.4	0.9
		20	89	18.3	11.4	8.5	2.8
6	$[t, 2t]$	10	49	6.0	0.8	4.7	−1.5
		20	49	7.6	3.3	5.8	−0.4
	$[t, 5t]$	10	49	14.3	4.6	9.7	−1.8
		20	49	15.7	13.5	10.1	−0.0
7	$[t, 2t]$	10	6	6.0	0.4	6.8	−3.2
		20	6	7.8	1.6	6.1	−2.3
	$[t, 5t]$	10	6	15.7	2.8	12.5	−5.5
		20	6	14.5	6.9	9.3	−1.7
8	$[t, 2t]$	10	1	5.6	−1.7	3.5	0.0
		20	1	5.6	−4.3	5.7	−4.4
	$[t, 5t]$	10	1	8.8	0.0	9.9	−2.5
		20	1	6.7	12.8	10.8	−5.4
Sum/weighted avg.			736	12.7	5.5	6.9	0.9

First note that for the groups and models where the number of optimally solved instances is less than 100 (see Table 1) the reported EVPIr is an upper bound on the true EVPIr, and the reported VSSr is a lower bound on the true VSSr, since the optimal value of the recourse problem may be lower. In particular, negative entries show that the solution of the recourse problem found by the solver is worse than the solution obtained using the best scenario.

For the fixed-tasks model the average EVPIr was 12.7%. This indicates that there is a real gain in knowing in advance the available workers. This fact can be used, for example, to justify measures in order to reduce the absenteeism in such contexts. Moreover, the average VSSr was 5.5%, showing that using a stochastic model pays off, if stochasticity can not be reduced. Similar observations hold for the flexible-tasks model with an average EVPIr of 6.9% and a average VSSr of 0.9%. The reduction in EVPIr is roughly proportional to the number of flexible tasks, except for the larger instances. For these instances, the solver was not able to find good solutions.

5.5 *Evaluation of the heuristic assignment procedure*

We next evaluate the heuristic assignment procedure. Since no other method is available in the literature for this problem, we use the results from the solver for smaller instances as a benchmark. Further tests are also executed with larger instances in order to evaluate the scalability of the heuristic in terms of computational efficiency. We conducted some preliminary experiments to find a reasonable set of parameters for the simulated annealing heuristic. For these experiments, we chose the first instance from each class of four to eight workers. In the tests we fixed the number of iterations per temperature level for the calibration phase to 1/10 of the number of iterations of the optimisation phase, but not less than 10. We also decided to fix the initial and final acceptance percentage at 70, and 5%, respectively. For $k \in \{1/4, 1/2, 1, 2, 4\}$ and $j \in \{1, 2, 3, 4\}$, we ran experiments with a cooling rate of 0.95^k and $100kj$ iterations per temperature level. In this way, for a fixed j, the average overall work per temperature level is constant. Each run was replicated three times. We found that the best parameter combinations achieved an average relative deviation of about 1 and 0.2% from the best solutions found over all runs for the fixed- and flexible-tasks model, respectively. We chose the parameters shown in Table 4 which achieved these limits fastest. For the main experiments, we ran the heuristic on all 736 instances and replicated each run 10 times.

We first focus on the instances with up to five workers, where the model for the stochastic solution could be solved to optimality for most instances. Of these instances, 371 have a feasible solution of the recourse problem for fixed tasks, and 409 for flexible tasks. The results are summarised in Table 5, which, for each model, gives the average relative deviation of the value of recourse problem found by simulated annealing from that found by solving the mathematical model ('R.d.'), the

Table 4. Parameters used in the simulated annealing heuristic.

Parameter	Description	Phase 1	Phase 2
L	Number of moves per temperature	40	400
p_i	Initial acceptance percentage	70%	–
r	Cooling rate	–	0.975
p_m	Final acceptance percentage	–	5%

Table 5. Heuristic results on instances with up to five workers.

			Fixed tasks				Flexible tasks			
$\|W\|$	Var.	Inc.	R.d.	EVPIr	VSSr	t(s)	R.d.	EVPIr	VSSr	t(s)
4	$[t, 2t]$	10	0.2	8.0	3.2	1.7	0.1	4.0	1.4	45.0
		20	0.0	16.8		1.8	0.2	7.2	0.0	99.4
	$[t, 5t]$	10	0.2	15.1	5.8	1.6	0.0	7.8	2.7	48.8
		20	0.1	25.8		1.7	0.0	11.8	5.7	85.8
5	$[t, 2t]$	10	0.2	5.8	2.4	1.7	−0.5	3.0	0.8	94.5
		20	0.6	12.8	2.4	3.1	0.3	6.3	2.3	161.1
	$[t, 5t]$	10	0.1	13.2	6.6	1.7	−0.4	6.1	1.2	98.1
		20	1.6	19.3	10.4	2.8	0.3	8.8	2.6	157.5
Weighted avg.			0.5	12.2	5.1	2.1	−0.1	6.0	1.8	110.3

average VSSr and EVPIr, as well as the average solution time in seconds ('t'). As above, the values are averages for groups of instances of the same number of workers ('$|W|$'), task time variability ('Var.'), and percentage of incompatibilities ('Inc.'), the EVPIr is computed over all instances which have a feasible solution of the recourse problem, and the VSSr is computed over all instances where the best scenario is feasible.

For the fixed-tasks model, the heuristic always found a feasible solution. The solution values are in average about 0.5% higher than those obtained by the model, over all feasible instances. In 199 instances, the heuristic was able to find the optimal stochastic solution in all 10 replications, and only 4 instances have an average gap of more than 5%. The running time for the fixed-tasks model did not exceed 5 s or 500 iterations. We also verified that the simulated annealing heuristic is effective: only 94 initial solutions were feasible, with an average relative deviation of 18.5%. After the best improvement local search, still only 263 solutions were feasible, with an average relative deviation of 5.3%.

For the flexible-tasks model the heuristic obtained 408 feasible solutions while the exact solver was able to solve one instance more. The average relative deviation is −0.1% over all feasible instances. The heuristic found the optimal stochastic solution in all 10 replications for 288 instances, and only one instance has a relative deviation of more than 5%. The EVPIr for the flexible-tasks model was 6.0% and, as expected by the low relative deviation, reduces the EVPIr of 6.6% of the stochastic model, which could not solve all flexible-tasks instances with five workers. The heuristic ran in average for about 2 min.

For the 217 a priori feasible instances with six to eight workers, the stochastic model could not be solved optimally, but both the model and the simulated annealing heuristic obtained feasible solutions in all of the 217 cases. We again compare the relative deviation of the heuristic value from the value found by solving the mathematical model ('R.d.'), the average EVPIr and VSSr, and the solution time ('t'). In all except two groups with fixed tasks, the heuristic quickly found better solutions than the solver. They were in average 0.3% less than that of the solver for the fixed-tasks model, and 2.4% lower for the flexible-tasks model, in about 2% of the computation time. Since the solutions of the recourse problem improved, the heuristic has an smaller EVPIr and a larger VSSr than the solver on the same instances. The values found are similar to those for a smaller number of workers: the relative VSS is about 5% for fixed tasks, and about 2% for flexible tasks. In comparison to scenarios with a smaller number of workers the values seem to indicate that the heuristic can produce solutions of similar quality also for higher numbers of workers. Indeed, as the number of a priori feasible solutions in each class suggests, the models are easier to optimise for a higher number of workers, since we have more options to assign the remaining workers to the workstations.

In summary, for the smaller instances, the solution times are comparable to that of the exact solver and the solution quality is very close to or slightly better than that found by the solver. The performance of the solver does not scale to instances with six or more workers. For these larger instances, the heuristic obtains better values in a much shorter time.

Table 6. Heuristic results on instances with six to eight workers.

			Fixed tasks				Flexible tasks			
$\|W\|$	Var.	Inc.	R.d.	EVPIr	VSSr	t	R.d.	EVPIr	VSSr	t
6	$[t, 2t]$	10	−0.6	5.5	1.4	2.3	−1.9	2.9	0.4	176.1
		20	0.5	8.0	3.0	4.0	−1.5	4.3	1.1	205.5
	$[t, 5t]$	10	−1.0	13.3	5.6	2.2	−3.4	6.5	1.7	180.0
		20	0.6	16.1	12.9	3.8	−2.0	8.3	2.2	202.5
7	$[t, 2t]$	10	−0.7	5.3	1.1	3.3	−3.7	3.2	0.5	339.9
		20	−1.6	6.4	3.1	4.2	−2.4	3.7	0.2	352.9
	$[t, 5t]$	10	−1.6	14.3	4.5	3.1	−6.4	6.5	1.0	302.7
		20	−1.6	13.0	8.8	3.4	−3.3	6.1	1.7	324.7
8	$[t, 2t]$	10	−1.7	4.0	−0.0	6.0	0.0	3.5	0.0	184.4
		20	−4.3	1.3	0.0	4.7	−4.4	1.3	0.0	132.4
	$[t, 5t]$	10	0.0	8.8	−0.0	4.0	−2.5	7.6	0.0	137.3
		20	−0.0	6.7	12.8	5.5	−6.9	4.1	1.6	189.6
Weighted avg.			−0.3	10.5	5.5	3.1	−2.4	5.4	1.3	205.4

6. Conclusions

We have addressed the problem of finding the best assignment of tasks to workstation in assembly lines under an uncertain heterogeneous workforce. This problem is an extension of the ALWABP, which is motivated by optimally planning the production in sheltered work centres for persons with disabilities, but can be applied in any situation where the performance variation among the workers cannot be neglected. We have presented two mathematical models for this problem for the cases of fixed- and flexible-task assignment and proposed a heuristic solution procedure based on simulated annealing and evaluated them on a set of 768 instances.

We have found that the solutions of the models is limited to small instances, and that the proposed heuristic scales better and is able to find good solutions in a much shorter time for instances with up to eight workers. The heuristic is effective, and can match the stochastic solution within 1% for almost all instances.

What concerns the usefulness and quality of the stochastic solutions, we have three main findings. First, the EVPI is generally larger than the VSS. This indicates that measures to reduce absenteeism in the workforce in general seem to be more effective than planning with stochasticity. In particular, an investment in the well being of workers of up to about 10% (for fixed tasks) of the daily production time can be beneficial. Second, if the stochasticity in the workforce cannot be reduced, planning a proper task assignment using a stochastic model is clearly better than scenario-based solutions. A stochastic model leads to more feasible solutions, and can also improve productivity by about 5% (for fixed tasks). Third, the reduction of the EVPIr when some tasks are made flexible seems to be roughly proportional to the number of flexible tasks. This indicates that if the manager is able to somehow reconfigure the line, even if only partially, then the stochasticity implied by the absenteeism looses importance. From a practical point of view, this suggest that lines should be made as flexible as possible, particularly in contexts with an unstable workforce. However, a realistic assessment must also take the time to reconfigure the line into account: in the instances considered here, the reconfiguration is beneficial up to about 5% of the production time, and even for higher percentages if we consider the additional workforce with set-up expertise that is often required. An interesting future research avenue is a more formal integration of these managerial aspects in the proposed analytical approach. This includes a more detailed study on scenario generation and a more formal evaluation of rebalancing costs in typical SWD lines.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by the CNPq under [grant number 478847/2013-0]; University of Melbourne under [grant number ECR 2015].

Supplementary data

The underlying research materials for this article can be accessed at http://dx.doi.org/10.1080/00207543.2015.1108534.

References

Agarwal, S., and M. K. Tiwari. 2008. "A Collaborative Ant Colony Algorithm to Stochastic Mixed-model U-shaped Disassembly Line Balancing and Sequencing Problem." *International Journal of Production Research* 46 (6): 1405–1429.

Araújo, F. F. B., A. M. Costa, and C. Miralles. 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.

Avriel, M., and A. C. Williams. 1970. "The Value of Information and Stochastic Programming." *Operations Research* 18 (5): 947–954.

Battaïa, O., and A. Dolgui. 2013. "A Taxonomy of Line Balancing Problems and their Solution Approaches." *International Journal of Production Economics* 143 (2): 259–277.

Baybars, I. 1986. "A Survey of Exact Algorithms for the Simple Assembly Line Balancing Problem." *Management Science* 32: 909–932.

Baykasoğlu, A., and L. Özbakir. 2007. "Stochastic U-line Balancing using Genetic Algorithms." *International Journal of Advanced Manufacturing Technology* 32: 139–147.

Becker, C., and A. Scholl. 2006. "A Survey on Problems and Methods in Generalized Assembly Line Balancing." *European Journal of Operational Research* 168 (3): 694–715.

Birge, J. R. 1995. "Models and Model Value in Stochastic Programming." *Annals of Operations Research* 59 (1): 1–18.

Birge, J. R., and F. Louveaux. 1997. *Introduction to Stochastic Programming*. New York: Springer.

Blum, C., and C. Miralles. 2011. "On Solving the Assembly Line Worker Assignment and Balancing Problem Via Beam Search." *Computers and Operations Research* 38 (2): 328–339.

Borba, L. M., and M. Ritt. 2014. "A Parallel Branch-and-bound for the Assembly Line Worker Assignment and Balancing Problem." In *Anais do XLVI Simpósio Brasileiro de Pesquisa Operacional*, 2293–2303. Salvador.

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

Boysen, N., M. Fliedner, and A. Scholl. 2008. "Assembly Line Balancing: Which Model to Use When." *International Journal of Production Research* 111 (2): 509–528.

Carraway, R. L. 1989. "A Dynamic Programming Approach to Stochastic Assembly Line Balancing." *Management Science* 35 (4): 459–471.

Carter, J. C., and F. N. Silverman. 1984. "A Cost-effective Approach to Stochastic Line Balancing with Off-line Repairs." *Journal of Operations Management* 4: 145–157.

Chakravarty, A. K., and A. Shtub. 1986. "A Cost Minimization Procedure for Mixed Model Production Lines with Normally Distributed Task Times." *European Journal of Operational Research* 23: 25–36.

Chiang, W.-C., and T. L. Urban. 2006. "The Stochastic U-line Balancing Problem: A Heuristic Procedure." *European Journal of Operational Research* 175: 1767–1781.

Cortez, P. M. C., and A. M. Costa. 2015. "Sequencing Mixed-model Assembly Lines Operating with a Heterogeneous Workforce." *International Journal of Production Research* 53 (11): 3419–3432.

Costa, A. M. 2013. "Instances for the ALWABP." http://www.inf.ufrgs.br/~mrpritt/alwabp-instances.rar.

Downey, B. S., and M. S. Leonard. 1992. "Assembly Line with Flexible Work-force." *International Journal of Production Research* 30 (3): 469–483.

Erel, E., I. Sabuncuoglu, and H. Sekerci. 2005. "Stochastic Assembly Line Balancing using Beam Search." *International Journal of Production Research* 43 (7): 1411–1426.

Fulkerson, R., I. Glicksberg, and O. Gross. 1953. "A Production Line Assignment Problem." Tech. Rep. Rm-1102. RAND Research memo..

Gamberini, R., E. Gebennini, A. Grassi, and A. Regattieri. 2009. "A Multiple Single-pass Heuristic Algorithm Solving the Stochastic Assembly Line Balancing Problem." *International Journal of Production Research* 47 (8): 2141–2164.

Gamberini, R., A. Grassi, M. Gamberi, R. Manzini, and A. Regattieri . 2004. "U-shaped Assembly Lines with Stochastic Tasks Execution Times: Heuristic Procedures for Balancing and Rebalancing Problems. In *Proceedings of the Business and Industry Symposium, Advanced Simulation Technologies Conference*, Arlington, VA, USA.

Henig, M. I. 1986. "Extensions of the Dynamic Programming Method in the Deterministic and Stochastic Assembly-line Balancing Problems." *Computers and Operations Research* 13: 443–449.

Hopcroft, J. E., and R. Karp. 1973. "An $n^{5/2}$ Algorithm for Maximum Matching in Bipartite Graphs." *SIAM J. Computing* 2: 225–231.

Inman, R. R., W. C. Jordan , and D. E. Blumenfeld. 2004. "Chained Cross-training of Assembly Line Workers." *International Journal of Production Research* 42 (10): 1899–1910.

Kao, E. P. C. 1976. "A Preference Order Dynamic Program for Stochastic Assembly Line Balancing." *Management Science* 22 (10): 1097–1104.

Kottas, J. F., and H.-S. Lau. 1973. "A Cost-oriented Approach to Stochastic Line Balancing." *IIE Transactions* 5: 164–171.

Kottas, J. F., and H.-S. Lau. 1976. "A Total Operating Cost Model for Paced Lines with Stochastic Task Times." *IIE Transactions* 8: 234–240.

Kottas, J. F., and H.-S. Lau. 1981. "A Stochastic Line Balancing Procedure." *International Journal of Production Research* 19 (2): 177–193.

Liu, S. B., H. L. Ong, and H. C. Huang. 2005. "A Bidirectional Heuristic for Stochastic Assembly Line Balancing Type II Problem." *International Journal of Advanced Manufacturing Technology*. 25 : 71–77.

Lyu, J. 1997. "A Single-Run Optimization Algorithm for Stochastic Assembly Line Balancing Problems." *Journal of. Manufacturing Systems* 16 (3): 204–210.

McMullen, P. R., and G. V. Frazier. 1997. "A Heuristic for Solving Mixed-model Line Balancing Problems with Stochastic Task Durations and Parallel Stations." *International Journal of Production Economics* 51: 177–190.

Miralles, C., J. P. Garcia-Sabater, C. Andrés, and M. Cardos. 2007. "Advantages of Assembly Lines in Sheltered Work Centres for Disabled. A Case Study." *International Journal of Production Research* 110 (2): 187–197.

Miralles, C., J. P. Garcia-Sabater, C. Andrés, and M. Cardos. 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 (2): 352–367.

Moodie, C. L., and H. H. Young. 1965. "A Heuristic Method of Assembly Line Balancing for Assumptions of Constant or Variable Work Element Times." *Journal of Industrial Engineering* 16 (1): 23–29.

Moreira, M. C. O., A. M. Costa, and C. Miralles. 2015. "Model and Heuristics for the Assembly Line Worker Integration and Balancing Problem." *Computers and Operations Research* 54: 64–73.

Moreira, M. C. O., M. Ritt, A. M. Costa, and A. A. Chaves. 2012. "Simple Heuristics for the Assembly Line Worker Assignment and Balancing Problem." *Journal of Heuristics* 18 (3): 505–524.

Mutlu, O., O. Polat, and A. Ayca. 2013. "An Iterative Genetic Algorithm for the Assembly Line Worker Assignment and Balancing Problem Of Type-II." *Computers and Operations Research* 40 (1): 418–426.

Office for Disability Issues (ODI) 2011. *Roadmap 2025: Disability Equality Indicators*. *Indicator B1: Employment rates*. https://www.gov.uk/government/organisations/office-for-disabi.

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.

Özcan, U., T. Kellegz, and B. Toklu. 2011. "A Genetic Algorithm for the Stochastic Mixed-model U-Line Balancing and Sequencing Problem." *International Journal of Production Research* 49 (6): 1605–1626.

Sarin, S. C., and E. Erel. 1990. "Development of Cost Model for the Single-model Stochastic Assembly Line Balancing Problem." *International Journal of Production Research* 28 (7): 1305–1316.

Sarin, S. C., E. Erel, and E. M. Dar-El. 1999. "A Methodology for Solving Single-model, Stochastic Assembly Line Balancing Problem." *Omega* 27: 525–535.

Scholl, A. 1999. *Balancing and Sequencing of Assembly Lines*. 2nd ed. Heidelberg: Physica.

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., and S. Voß. 1996. "Simple Assembly Line Balancing - Heuristic Approaches." *Journal of Heuristics* 2: 217–244.

Shin, D., and H. Min. 1991. "Uniform Assembly Line Balancing with Stochastic Task Times in Just-in-time Manufacturing." *International Journal of Operations and Production Management* 11 (8): 23–34.

Shtub, A. 1984. "The Effect of Incompletion Cost on Line Balancing with Multiple Manning of Work Stations." *International Journal of Production Research* 22 (2): 235–245.

Silverman, F. N., and J. C. Carter. 1986. "A Cost-based Methodology for Stochastic Line Balancing with Intermittent Line Stoppages." *Management Science* 32: 455–463.

Sivasankaran, P., and P. Shahabudeen. 2014. "Literature Review of Assembly Line Balancing Problems." *International Journal of Advanced Manufacturing Technology* 73: 1665–1694.

Suresh, G., and S. Sahu. 1994. "Stochastic Assembly Line Balancing using Simulated Annealing." *International Journal of Production Research* 32 (8): 1801–1810.

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

Wilson, J. 1986. "Formulation of a Problem Involving Assembly Lines with Multiple Manning of Work Stations." *International Journal of Production Research* 24 (1): 59–63.