

# Markov decision process for multi-manned mixed-model assembly lines with walking workers

S. Ehsan Hasemi-Petroodi<sup>a,\*</sup>, Simon Thevenin<sup>a</sup>, Sergey Kovalev<sup>b</sup>, Alexandre Dolgui<sup>a</sup>

<sup>a</sup>*IMT Atlantique, LS2N, Nantes, France*

<sup>b</sup>*INSEEC Grande Ecole - INSEEC U. Research Center, 25 rue de l'Université, 69007 Lyon, France*

---

## Abstract

Product customization and frequent market changes force manufacturing companies to employ mixed-model instead of simple assembly lines. To well adjust the line's capacity to production requirements, the line can benefit from the concept of reconfigurability. Our study deals with a reconfigurable mixed-model assembly line where tasks can be dynamically assigned to stations at each takt, workers can move among stations at the end of each takt, and the order of entering product models is infinite and unknown. The equipment assignment to stations occurs at the line design stage, and equipment duplication is allowed. The dynamic task assignment and workers' movements among stations is a Markov Decision Process (MDP) that can be translated as a Linear Program (LP). As a result, the line design problem is formulated as a mixed-integer linear program (MILP) that integrates the MDP model. We propose some reduction rules and a decomposed transition process to reduce the model. The new MILP models taking into account stochastic parameters are built to solve the stochastic and robust problems, with the objectives of expected total cost minimization in all takts and total cost minimization in the worst takt, respectively. Computational experiments with benchmark and generated instances demonstrate the performance of the

---

\*Corresponding author

*Email address:* `seyyed-ehsan.hashemi-petroodi@imt-atlantique.fr` (S. Ehsan Hasemi-Petroodi)

proposed MDP models. The managerial insights provided in the paper show the superiority of the dynamic task assignment over the model-dependent and fixed assignments usually studied in the literature.

*Keywords:* Reconfigurable manufacturing system, Mixed-model assembly line, Workforce assignment, Dynamic task assignment, Walking workers, Markov decision process

---

## 1. Introduction

As the demand for individualization is growing, manufacturing companies face the necessity to incorporate mass customization, which combines a sufficient production volume and product model differentiation. Customization increases product satisfaction and purchases likelihood (Kaiser et al., 2017; Moreau and Herd, 2010; Valenzuela et al., 2009). In many industries, this strategy is no longer just an opportunity to create a competitive advantage but rather a condition of the company’s long-term prosperity. The production system used by a company should be able to manufacture different product models without losing high productivity. Assembly lines, due to their flow shop nature, are the most common type of production systems. Dynamic changes in the market and increasing demand for customization push manufacturing companies to use mixed/multi-model assembly lines instead of single model ones (Kucukkoc and Zhang, 2014b).

The sequence of products to be assembled has a large impact on the performance of a mixed-model assembly line because the items do not have the same operations and operation times. Typically, the sequence is selected to avoid successive items with long process duration at a station (Miltenburg, 1989). For instance, Renault imposes a ratio of items with complex operations in the production sequence (Zufferey, 2016). Unfortunately, there exist many situations where the production sequence on the assembly line may not be perfectly controllable. For instance, the production sequence may depend on the schedule of an upstream workshop, items may be removed from the initial sequence because they fail quality test (Boctor et al.,

2000), urgent orders may be added, some orders may be removed due to part shortages (Liu et al., 2012), or some companies face highly variable demands in make-to-order production strategy (Bukchin et al., 2002).

To overcome this challenge, the line should have a high degree of reconfigurability (Koren et al., 1999). In other words, an ability to quickly adapt the resources, human operators, and machines, to the incoming product models. In this work, we investigate a mixed-model assembly line able to reconfigure to any entering item order that respects given sequencing constraints. The reconfiguration is possible because of the possibility to re-assign tasks (reconfigurable equipment) and to move workers among stations (workforce reconfigurability).

Thus, this paper addresses a multi-manned manual mixed-model assembly line balancing problem with walking workers (MALBP-W). The line’s reconfigurability is achieved by moving workers between stations and equipment duplication. These reconfigurations allow shifting the production capacity from one takt to another one with the required level of productivity. The studied problem has two stages. The design stage decides the number of workers and the equipment assignment to stations for any possible order of products entering the line. At the operational stage, the items entering the line are revealed takt by takt, and tasks and workers are assigned to the stations depending on this information.

Taking into account the highly dynamic and stochastic nature of the studied problem, the choice of the solution method fell on Markov Decision Process (MDP). In our case, the order of entering products to the line is unknown and subject to frequent changes. To the best of our knowledge, this paper is the first attempt to develop an MDP model for a MALBP-W with dynamic task assignment. We consider two variants of this model. The first one is stochastic to minimize the expected cost of workers and equipment. The second one is robust to minimize the workforce and equipment cost for the worst takt. To reduce the complexity of the model, we propose several reduction rules to simplify and enhance its performance. Compared to the majority of studies in the literature, we consider an infinite

unknown order of different products entering the line, and we aim to design a line that can self-adjust to the incoming products of different types (product models). This paper also evaluates the impact of a dynamic task assignments strategy at the operational stage, compared to model-dependent and fixed task assignments. In the fixed task assignment, tasks are performed at the same stations for all possible product models. In the model-dependent task assignment studied in [Hashemi-Petroodi et al. \(2022\)](#), each product model has its own task assignment. Dynamic task assignment means that tasks can be re-assigned at the end of each takt depending on the new product entering the line and depending on the position of other product models on the line.

The paper is structured as follows: Section 2 presents a literature review on manned mixed-model assembly line problems and MDP applications. Section 3 describes the general studied problem. Section 4 explains the MDP approach and how it can be applied for this problem as well as presenting the stochastic and robust versions of the MDP model. Section 5 deals with the restricted task assignment policies and the algorithmic improvements of the proposed MDP. Section 6 provides the computational results of the benchmark instances and a case study, as well as several managerial insights. The paper ends with the conclusion and future research directions in Section 7.

## 2. Literature review

This section reviews the literature on the key concepts considered in this paper, namely, mixed-model assembly line design and balancing, fixed/walking workers in workforce assignments, fixed/dynamic task assignments, and the corresponding optimization approaches. This section concludes with the main contributions of the current study compared to the existing literature.

Assembly line balancing (task assignment) and design (resource assignment) are crucial steps for a mixed-model assembly line (MMAL) ([Boysen et al., 2009](#)). In the literature on

assembly line design and balancing problems ([Baybars, 1986](#); [Scholl and Becker, 2006](#); [Boysen et al., 2008](#); [Battaïa and Dolgui, 2013](#)), a number of restrictive assumptions are commonly made. These assumptions include: assigning a single worker to each station, producing only a single product model in the line, keeping fixed workers and tasks at stations, etc. These assumptions are invalid in certain industries which manufacture different product variants and enhance flexibility through workers and task re-assignment. For instance, in an automotive industry with heavy tasks performed on large-size products, assigning more than one worker to each station is more realistic ([Lopes et al., 2020](#)). Multi-manned line balancing problems are often formulated using different restrictive assumptions, such as fixed workers at stations ([Dimitriadis, 2006](#); [Michels et al., 2019](#); [Becker and Scholl, 2009](#); [Kellegöz, 2017](#)). Moving workers between stations allows such lines to adjust the capacity of stations to the production sequence ([Sikora et al., 2017](#)). Nowadays, due to shortening product lifecycles and frequent market changes, such lines require an improved reconfigurability.

Reconfigurability of an assembly line can be achieved using walking workers who can change the production capacity when and where it is needed ([Koren et al., 1999](#); [Hashemi-Petroodi et al., 2020a](#)). In several publications, workers are fixed at each station, and they are not allowed to move and work at other stations ([Biele and Mönch, 2018](#)). However, some researchers assumed workers' movement between stations ([Battaïa et al., 2015](#); [Naderi et al., 2019](#); [Al-Zuheri et al., 2016](#); [Şahin and Kellegöz, 2019](#)). Such re-assignment of workers may take into account skill sets and task requirements. The processing time of each task can depend on the number of workers who execute it ([Battaïa et al., 2015](#)).

From another perspective, task assignment to stations can be either given ([Battaïa et al., 2015](#)), fixed ([Özcan et al., 2010](#); [Sikora et al., 2017](#)), or dynamic ([Kucukkoc and Zhang, 2014b](#)). The literature on dynamic task assignment for mixed-model assembly line (MMAL) is scarce. [Kucukkoc and Zhang \(2014b\)](#) study a parallel two-sided MALBP with fixed workers where operators work at both sides of the line. The authors consider a dynamic change

of task assignment at each production cycle, where a production cycle corresponds to a certain combination of product models at stations. [Choi \(2009\)](#) consider model-dependent task assignment in a simple MMAL with fixed workers. [Hashemi-Petroodi et al. \(2020b\)](#) propose a scenario-based mixed integer linear program for the MMAL balancing with walking workers and dynamic task assignment. Unlike in the current paper, [Hashemi-Petroodi et al. \(2020b\)](#) assume the order of the product models entering the line is finite and given before deciding task assignment. Moreover, [Hashemi-Petroodi et al. \(2021\)](#) recently evaluated the performance of the model-dependent task assignment compared to the fixed one in a similar mixed-model assembly line with walking workers. In the model-dependent task assignment, each product model has specific task assignment. Except for the mentioned studies, all studies on MMAL balancing with walking workers assume task assignment is either fixed or given ([Battaïa et al., 2015](#); [Delorme et al., 2019](#); [Dolgui et al., 2018](#); [Hwang and Katayama, 2010](#)). Nevertheless, decisions on tasks and workforce assignments have to be made simultaneously, since any change in task assignment may imply changes in workforce assignment ([Cortez and Costa, 2015](#)). The current study considers a more flexible environment, where the task assignment is product-unit dependent and can change in each takt for each product model. In addition we take into account a general and dynamic type of uncertainty, where an infinite unknown order of products can enter the line, and decisions are made without knowledge on the next product to enter the line.

As for solution methods applied to MMAL balancing and design problems, researchers mostly rely on mathematical programming models ([Lopes et al., 2018](#); [Battaïa et al., 2015](#); [Biele and Mönnch, 2018](#); [Kucukkoc and Zhang, 2014a](#); [Giard and Jeunet, 2010](#)), various exact methods ([Bukchin and Rabinowitch, 2006](#); [Li and Gao, 2014](#); [Alghazi and Kurz, 2018](#); [Delorme et al., 2019](#); [Choi, 2009](#)), and (meta-)heuristics ([Tiacchi, 2015](#); [Dolgui et al., 2018](#); [Özcan et al., 2010](#); [Samouei and Fattahi, 2018](#); [AkpıNar et al., 2013](#); [Kucukkoc et al., 2018](#); [Saif et al., 2019](#); [Samouei and Fattahi, 2018](#)). The Markov Decision Process (MDP) is a mod-

elling framework commonly used in reinforcement learning which is one of the main machine learning paradigms (Bengio et al., 2020). Application of the machine learning approaches to operations research problems is becoming more and more attractive (Karimi-Mamaghan et al., 2022; Kang et al., 2020; Morin et al., 2019; Bengio et al., 2020). An MDP represents a sequential decision-making problem under uncertainty. It describes the process of transformation of the system’s current state to another state through actions. Recent researches on the applications of MDP and their integration with optimization methods are reviewed in Ahiska et al. (2013); Salari and Makis (2017); Ahluwalia et al. (2021); Alagoz et al. (2015); Steimle et al. (2021). Most applications concern transport problems (Yu et al., 2019; Kamrani et al., 2020; Li et al., 2021; Qiu et al., 2022), and these studies show that mathematical programming methods (e.g., LP, MILP) perform efficiently to solve MDP problems case-dependently (Alagoz et al., 2015). To overcome the higher complexity of large size instances using mathematical programming, exact optimization approaches such as decomposition (Steimle et al., 2021) and branch-and-bound (Ahluwalia et al., 2021) algorithms can perform well to tackle MDPs.

To the best of our knowledge, this paper is the first attempt to apply an MDP model to a MMAL balancing problem. The problem’s dynamic nature with possible task re-assignments at each takt, uncertainty regarding the product order represent the sources of motivation behind this study.

The superiority of model-dependent task assignment (every model has its own unchanging task assignment) over the fixed one (task assignment for all models is the same) has been already revealed in Hashemi-Petroodi et al. (2022). The goal of this paper consists not only in assessing the efficiency of dynamic task assignment represented by an MDP model, but it also seeks to compare dynamic task assignment to model-dependent and fixed task assignments.

### 3. Problem description

This section describes a multi-manned mixed-model assembly line balancing problem with walking workers (*MALBP-W*). The line includes a set  $\mathcal{S} = \{1 \dots S\}$  of sequential stations, and it assembles a set  $\mathcal{I} = \{1 \dots I\}$  of product models. The line is paced, and at the end of each takt time  $C$ , all products move simultaneously towards the next station. Each product model  $i$  requires the set  $\mathcal{O}_i$  of tasks, and we denote by  $\mathcal{O}$  the total unified set of all tasks of all product models assembled at the line. The processing time  $p_{io}^l$  of task  $o$  varies with product model  $i$ , and it depends on the number of workers  $l$  processing the task. In other words, the task processing time decreases when the number of workers at the station increases ([Battaïa et al., 2015](#)). At most,  $l_{max}$  workers are allowed to work at the same station. The assignment of tasks to stations must respect the set  $A$  of precedence constraints. More precisely,  $A$  contains the pairs of tasks  $(o, o')$  if task  $o$  must precede task  $o'$ . To perform a task  $o$  at a station  $s$ , at least one equipment piece with the ability to perform  $o$  must be installed in  $s$ . Such equipment corresponds to tools and resources the worker needs to perform the task, such as a screwdriver, a gripper, etc. The set of equipment is denoted by  $\mathcal{E}$ , and the requirements are represented with the parameter  $r_{oe}$ . If  $r_{oe}$  is equal to 1, equipment  $e$  has the ability to perform task  $o$ , and  $r_{oe}$  is equal to 0 otherwise. An equipment piece can be duplicated at each station when some of the assigned tasks to the station require the same type of equipment (see [Askin and Zhou, 1997](#); [Tiacchi and Mimmi, 2018](#), for example). Each equipment  $e$  has a cost  $c_{se}$  at each station  $s$ . The equipment cost can be station dependent because of the space restrictions, state of infrastructure at stations, proximity to utilities, removal of previous equipment and difficulty of installing the equipment. However, our model can also handle a less general case in which equipment cost is not station dependent. The cost of a worker  $\alpha$  is the same for all workers because we assume all workers are identical in terms of their ability to perform any tasks. The objective at the design stage is to a line



able to reconfigure to face any possible product model order. We call a reconfiguration the movements of workers and the task re-assignment. The design decisions include, the number  $Y$  of workers to hire and positions (at which station) of fixed pieces equipment. At the operational stage, problem  $MALBP - W$  consists in dynamically assigns the tasks and workers to the stations at each takt depending on the line's state. Workers can move from a station to another at the end of each takt to adapt the production capacity at stations before the arrival of an entering product. We assume that the workers' walking times are negligible taking into account the takt and processing times (Battaïa et al., 2015). Tasks can be assigned to any stations with the required equipment, and equipment can be duplicated at the design stage to increase the reconfigurability.

As products enter the line, they consecutively occupy workstations creating different "pictures" of the line evaluating over time (at the end of each takt). By **picture**, we mean the sequence of pairs "station - product model" that changes (product items shift towards the last station) every takt. As opposed to the concept of the line's picture, a **product order** can be defined as a sequence of product models entering the line whose number is not limited by the number of stations. For example, a product order can be  $(B - A - A - C - A - C - C)$ , where product B enters the line first, then two products A follow, and so on. Suppose that the line consists of four stations. One of the possible pictures of the line for such product order is: (station 1 - product C, station 2 - product A, station 3 - product A, station 4 - product B). In the following takt, the picture of the line is (station 1 - product A, station 2 - product C, station 3 - product A, station 4 - product A). The only product B present in the product order has left the line.

A line able to reconfigure for any possible input sequence requires too many workers, since the sequence may include the items with the largest process duration in all stations. To reduce the cost of the line, the decision-maker may impose constraints on the incoming product orders. In practice, the decision-maker infers such constraints from historical data,

or he may select them such that the scheduling tool always finds a feasible sequence. In this work, we consider two types of constraints, but the proposed solution approach is generally applicable to any constraints that respect the Markov property of the MDP. The first type of constraint prevents the presence of more than  $u_i$  units of an item  $i$  in a picture of the line. For example, to preserve the diversity of out-coming final products, the user decides to have at most two units of models  $A$ ,  $B$ , and  $C$  at 3 stations. The second prevents having more than  $u'_i$  consecutive units of an item  $i$  in the product order. Such restrictions can be caused by producing a relatively rare customized/luxury product model together with more popular mass-production models.

The studied problem  $MALBP - W$  with dynamic task assignment is called  $MALBP - W^{Dyn}$ . To further clarify the proposed dynamic task assignment policy, we provide an illustrative example in [Appendix A](#).

We consider two variants of the problem  $MALBP - W^{Dyn}$ . Both variants seek to minimize the worker and equipment cost, but they compute the workforce cost differently. In the first variant, workers are hired specifically for the assembly line, and the objective is to minimize the number of workers to hire. In the second, the workers may work on other work cells, and the objective is to minimize the expected number of workers in the line.

#### 4. Markov Decision Process (MDP) application

This section presents how the MDP is applied for this problem.

##### 4.1. Markov Decision Process (MDP)

Markov Decision Processes (MDPs) model sequential decision-making problems in dynamic and uncertain environments. An MDP includes states, actions, transition matrix, and transition rewards. In each decision step, the system is in a state  $d \in \mathcal{D}$ , and the agent selects an action  $a$  in the set  $A_d$  of possible actions in state  $d$ . The system switch from a

state to another with a probability, and  $Tr_a(d \mid d')$  gives the probability to transition from state  $d$  to state  $d'$  with action  $a$ . We present below the state, actions, transition probability matrix, and reward function for the considered problem.

**State.** Each state  $d \in \mathcal{D}$ , represents the picture of the line (positions of product models at stations) and sets of tasks that have already been performed for each product model located in the line. The information given by state  $d$  is described by two matrices. The first matrix contains values  $F_{isd}$  equal to 1 when model  $i$  is located at station  $s$  in state  $d$ , 0 otherwise. The second matrix contains values  $P_{osd}$  equal to 1 if task  $o$  has already been executed for the model in station  $s$  when the system passed to state  $d$ . Note that  $P_{osd}$  is equal to 0 for all  $o \in \mathcal{O}_i$ , when product  $i$  comes to the first station ( $s = 1$ ) since there are no performed tasks for an item entering the line. The set of states  $D$  includes all combinations of all pictures of the lines respecting the restriction with all possible sets of performed tasks respecting the precedence constraints.

**Action.** At each state  $d$ , an action  $a \in A_d$  determines a possible task assignment to stations for each product model on the line in the current takt/state. Each action associated with state  $d$  is represented by a binary matrix where each cell  $R_{osa}$  is equal to 1 if action  $a$  performs task  $o$  on the model in station  $s$ . To create the actions, we consider precedence constraints, takt time and the maximum number of workers per station  $l_{max}$ . Given a state  $d$ , we consider for each station  $s$ , the set  $Z_{sd}$  of tasks that are not already performed ( $Z_{sd} = \{o \mid o \in \mathcal{O}_{i(s,d)}, P_{osd} = 0\}$ ), where  $i(s, d)$  denotes the product model in station  $s$  in state  $d$ . The set of action corresponds to all combinations (over station 1 to  $S - 1$ ) of subsets of  $Z_{sd}$  that respect the precedence constraints. For station  $S$ , the set of tasks to perform is exactly  $Z_{sd}$  since the product must be completely assembled in the last station.

The number of workers needed per station is given by an integer parameter  $q_s^a$ . The number  $q_s^a$  of workers required in station  $s$  for action  $a$  is the smallest integer  $l$  such that the total processing time in station  $s$  respect the takt time.  $q_s^a$  is calculated for each station

$s \in \mathcal{S}$  and action  $a \in \mathcal{A}_d$ , where  $d \in \mathcal{D}$ , as follows:

$$q_s^a = \min \left\{ l \mid \sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}_i} p_{io}^l R_{osa} F_{isd} \leq C \right\}, \quad (1)$$

Where  $d$  is the state associated with action  $a$ . If an action requires more than  $l_{max}$  workers, it is not considered. We also denote  $q'_{ad}$  as the number of workers needed in the line for action  $a$  in state  $d$ . We calculate  $q'_{ad}$  by summing up the number of workers needed at each station  $q_s^a$ , as  $q'_{ad} = \sum_{s \in \mathcal{S}} q_s^a$  for each  $a \in \mathcal{A}_d$  and  $d \in \mathcal{D}$ . The task assignment to stations can be related with the item at the station with a binary parameter  $y_{soi}^a$  equal to 1 if task  $o$  of product model  $i$  is performed at station  $s$  within action  $a$ , 0 otherwise.  $y_{soi}^a$  can be generated considering  $R_{osa}$  and  $F_{isd}$  ( $y_{soi}^a = R_{osa} F_{isd}$ ) where  $d$  is the state associated with action  $a$ .

**Transition.** Transition follows a procedure. Given the current state, the selected action gives the set of additional tasks performed on the items. The item in the last station of the current state leaves the line. All items in other stations simultaneously move towards the next station, while a new item enters the line and passes to the first station. As different product models may enter the line one at each time, the system may transition to different states. The transition probability matrix corresponds to the probability of each product model to enter the line. This transition matrix can account for the restriction on the order of products if the restrictions respect the Markov property. In this case, the probability to move to a state that does not respect the maximum number of units constraint must be equal to 0 (such states can simply be removed). Similarly, the probability to move to a state that does not respect the successive number of item constraints must be equal to 0. However, the limit on the number of successive items in the line must be lower or equal to the number of stations to respect the Markov property.

**Reward.** In a large series production context, the line will run without interruption for a long period. Therefore, we consider an infinite horizon MDP, and we optimize for the long run use of the system (i.e. in the steady state situation).

Solution methods that find the optimal policies for MDPs include linear programming (Alagoz et al., 2015; Buchholz and Scheftelowitsch, 2019), policy iteration algorithm (Patek, 2004; Pavitsos and Kyriakidis, 2009), and value iteration algorithm (Zobel and Scherer, 2005). While linear programming has been rarely applied to solve MDPs compared to other approaches (Alagoz et al., 2015), the LP solution methods are efficient and flexible enough to account for various constraints in the MDP (Buchholz and Scheftelowitsch, 2019). The rest of this section provides a linear program to solve the two considered variants of the problem.  $MDP^{Sto}$  minimizes the long run expected number of workers, and it corresponds to the classical application of MDPs, whereas  $MDP^{Ro}$  optimizes for the worst possible reachable state in a long run.

#### 4.2. Stochastic model $MDP^{Sto}$

The LP approach to solve the MDP model relies on continuous variables  $0 \leq X_{ad} \leq 1$  that determine the probability of taking an action  $a$  in state  $d$ . In addition, variable  $W_{se}$  is equal to 1 if equipment  $e$  is chosen for station  $s$ , and 0 otherwise. The corresponding  $MILP^{Sto}$  is represented by (2) - (6) as follows:

$$\min \quad \sum_{d \in \mathcal{D}} \sum_{a \in A_d} \alpha q'_{ad} X_{ad} + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} c_{se} W_{se} \quad (2)$$

s.t.

$$\sum_{d \in \mathcal{D}} \sum_{a \in A_d} T r_a^{dd'} X_{ad} = \sum_{a' \in A_{d'}} X_{a'd'} \quad d' \in \mathcal{D} \quad (3)$$

$$\sum_{d \in \mathcal{D}} \sum_{a \in A_d} X_{ad} = 1 \quad (4)$$

$$y_{soi}^a X_{ad} \leq \sum_{e \in E} r_{oe} W_{se} \quad s \in \mathcal{S}, o \in \mathcal{O}, i \in \mathcal{I}, a \in A_d, d \in \mathcal{D} \quad (5)$$

$$W_{se} \in \{0, 1\}, 0 \leq X_{ad} \leq 1 \quad (6)$$

The objective function (2) is the equipment cost and long run expected cost value of workers needed. Constraints (3) verify that total probability to come to a state ( $d'$ ) from any possible state ( $d$ ) by any action  $a$  is equal to the probability to get out of that state to any other state. Constraint (4) forces the probability to take any actions in the system to be equal to 1. Constraints (5) locate necessary equipment at stations regarding the assigned tasks requirements. Constraints (6) provide the variables domains.

#### 4.3. Robust model $MDP^{Ro}$

For the robust model  $MDP^{Ro}$ , the binary decision variable  $V_a$  is added to the model. It is equal to 1 if the action  $a$  is taken with non zero probability, and 0 otherwise. In addition, variable  $Y$  computes the worst case number of workers. Finally,  $MILP^{Ro}$  is given as follows:

$$\min \quad \alpha Y + \sum_{s \in \mathcal{S}} \sum_{e \in \mathcal{E}} W_{se} c_{se} \quad (7)$$

s.t.

$$V_a \geq X_{ad} \quad a \in A_d, d \in \mathcal{D} \quad (8)$$

$$Y \geq \sum_{s \in \mathcal{S}} q_s^a V_a \quad a \in A_d, d \in \mathcal{D} \quad (9)$$

$$(3) - (6)$$

$$V_a \in \{0, 1\}, Y \geq 0 \quad (10)$$

The objective function (7) calculates the total cost of equipment used and the maximum number of workers needed for the worst possible action to move from any state. Constraints (8) set variable  $V_a$  to 1 if the probability to take action  $a$  is positive (non-zero). Constraints (9) calculate the maximum number of workers needed in the line for the worst case.

[Appendix B](#) provides an MDP graph with all states, actions, and transition probabilities for the example in [Appendix A](#).

## 5. Fixed task assignment policies and algorithmic improvements

This section first presents an extension of the models to yield restricted task assignment policy. In particular, we want to compare the performance of the dynamic task assignment, with policies where tasks are fixed to the station, or fixed for each model. Then, we introduce algorithmic improvements to speed up the computation. Solving the considered model is challenging because the number of states and, especially, actions, grows exponentially with the number of tasks and stations. To alleviate this issue, we propose several reduction rules that eliminate some not-optimal/not-feasible actions and states (Subsection 5.2). Building the transition matrix is the most time and memory consuming part of the pre-processing stage. A decomposition algorithm using sub-matrices related to different pictures of the line is proposed (Subsection 5.3). In addition, several actions and states which cannot be reached are eliminated by the algorithm.

### 5.1. Fixed task assignment policy

We aim to compare the solution quality of  $MALBP - W^{Dyn}$  with model-dependent  $MALBP - W^{Md}$  and fixed task assignment  $MALBP - W^{Fix}$ , studied by Hashemi-Petroodi et al. (2022). In  $MALBP - W^{Md}$  task assignment varies for different product models, but for every product of a certain model it remains the same. In  $MALBP - W^{Fix}$  task assignment is the same for all product models. To clarify these policies, we provide an illustrative example in Appendix C, where the model-dependent and fixed task assignments are drawn compared to the dynamic policy through the same example as in Appendix A. We provide below the constraint added to the MDP to yield such policies. Note that the resulting model is more generic than the one proposed by Hashemi-Petroodi et al. (2022) since it allows modeling any constraint on the order of products that respect the Markov property, and it allows minimizing the expected number of workers.

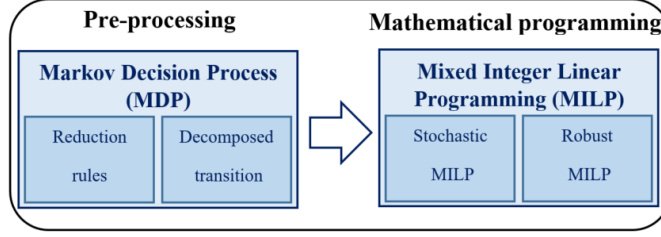


Figure 1: The framework of the proposed methodology.

To restrict the policy to model-dependent ( $MALBP - W^{Md}$ ) and fixed ( $MALBP - W^{Fix}$ ) task assignment, we add a set of constraints on the selected actions. We identify the sets of all pairs of incompatible actions  $a$  and  $a'$ . These sets are denoted as  $\mathcal{A}^{Md}$  and  $\mathcal{A}^{Fix}$  for model-dependent and fixed task assignment, respectively. Two actions are incompatible if we do not observe the same task assignment to a station for a certain product model (resp. not observing the same task assignment for all products) in  $MALBP - W^{Md}$  (resp.  $MALBP - W^{Fix}$ ). Two sets of constraints (11) and (12) are added to  $MILP^{Sto}$  (2) - (6) and  $MILP^{Ro}$  (7) - (10) and (3) - (6) to ensure the resulting policy follows the model-dependent and fixed task assignment policy, respectively. Note that, in this context, Constraints (8) are also added to  $MILP^{Sto}$ .

$$V_a \leq 1 - V_{a'} \quad a, a' \in \mathcal{A}^{Md} \quad (11)$$

$$V_a \leq 1 - V_{a'} \quad a, a' \in \mathcal{A}^{Fix} \quad (12)$$

## 5.2. Reduction rules for states and actions

The proposed approach consists of two stages: pre-processing and mathematical modelling, see Figure 1. At the pre-processing stage, we create all components of the MDP: states, actions, transition probabilities, and reward function. At the second stage, two  $MILP$  models are built for stochastic and robust versions of the problem.



To improve the performance of the proposed MDP method, the number of generated states and, especially, actions must be reduced. Pre-processing accelerates the transition matrix creation, and, subsequently, enhances solving process by MILP. To eliminate states and actions that cannot be present in a feasible solution, we propose the following rules:

1. As the transition matrix does not allow "no product" to enter the line, the long run probability for such state is 0. Therefore, we eliminate states and the corresponding actions where only some stations are occupied.
2. Remove states where the total processing times of the performed tasks in previous takts for a product item requires more than  $(s - 1) l_{Max}$  workers, where  $s$  is the current station at which the item is located ( $l_{Max}$  per station). Such states are not reachable, since it would require actions with more than  $l_{Max}$  workers. This concerns all states  $d$  where there exist  $s$  such that :

$$\sum_{i \in \mathcal{I}} \sum_{o \in \mathcal{O}_i} F_{isd} \cdot P_{osd} \cdot p_{io}^{l_{Max}} > (s - 1)C.$$

3. [Hashemi-Petroodi et al. \(2022\)](#) provides an efficient method to solve  $MDP^{Ro}$  for the special case of a model-dependent policy without restrictions on the number of successive items on the line. The solution to this special case provides an upper bound for the generic version of  $MDP^{Ro}$  and  $MDP^{Sto}$ . This upper bound is valid for  $MDP^{Ro}$  because the solution of the model-dependent policy requires adding constraints (11). Therefore the solution to the special case of a model-dependent policy is a feasible but non necessarily optimal solution to  $MDP^{Ro}$ . In addition, the special case with less restriction on the order of products leads to larger costs than considering a restricted order, since it accounts for a larger uncertainty set. Obviously,  $MDP^{Ro}$  provides an upper bound to  $MDP^{Sto}$ . We remove the actions which need more workers than the

upper limit on the number of workers in the line. Subsequently, we remove the states with no incoming transition.

### 5.3. Decomposed transition

Since creating the transition matrix demands a lot of computational effort, we propose a decomposition algorithm to make it more efficient. According to this algorithm, the whole transition matrix with all states and actions is decomposed in sub-matrices associated with each possible pictures of the line. Subsequently, the actions related to each subset of states are decomposed into actions corresponding to these stats.

A state  $d'$  can be reached from all states  $d$  where the items in position 1 to  $S-1$  correspond to the items in position 2 to  $S$  in  $d'$ , and where no task performed in  $d$  remain to be performed in  $d'$ . We denote by  $\widehat{\mathcal{D}}_{d'}$  these states that may precede state  $d$ . For all the other states,  $Tr_a^{dd'}$  is equal to 0. Therefore, Constraints (3) can be applied only to preceding states of state  $d'$  regarding the line pictures. Constraints 3 are reformulated as follows:

$$\sum_{d \in \widehat{\mathcal{D}}_{d'}} \sum_{a \in A_d} Tr_a^{dd'} x_{ad} = \sum_{a' \in A_{d'}} x_{a'd'} \quad d' \in \mathcal{D} \quad (13)$$

Rather than generating a transition matrix with many 0, we generate a transition matrix for each set of starting states that correspond to identical sub-pictures from stations 1 to  $S-1$ . These states are the only ones that may lead to ending states with these sub-pictures in position 2 to  $S$ . When generating this transition matrix, we remove ending states that cannot be reached. For instance, reaching some states would require actions with more workers than the upper bound. The procedure iterates until no more state may be removed.

If there are no restrictions on the product model order, the total number of line pictures is equal to  $I^S$ . However, the restrictions may significantly reduce the number  $PN$  of line pictures. For each picture, we independently generate the set of states that correspond to

different numbers of performed tasks. This process yields  $PN$  sets of states, denoted  $N_k$ , where  $k = 1, \dots, PN$ . Subsequently, the states existing in  $N_k$  determine  $PN$  subsets of actions, denoted as  $M_k$  where  $k = 1, \dots, PN$ . Finally, the set of possible resulting states  $d'$  characterized by line pictures, task, workforce and equipment assignments is denoted as  $N'_k$  where  $k = 1, \dots, PN$ , that contains the aggregation of sets  $N_k$  ( $N'_k = \bigcup_{k \in \{1, \dots, PN\}} N_k$ ). Note that if a picture remains the same from the current state to the resulting state, since the task assignment of the line changes, the system does not move to the same state and this state (current state) must be removed from the set of resulting states ( $N'_k = (\bigcup_{k \in \{1, \dots, PN\}} N_k) - d$ ).

Note that,  $|A0|$  and  $|D0|$  denote as the number of all generated initial actions and states, respectively. Next,  $|A|$  actions and  $|D|$  states remain after applying reduction rules. Decomposed transition results in a less and final number of states ( $|D1|$ ) and actions ( $|A1|$ ) which enter the proposed *MILPs*.

Following the same simple example as in [Appendix A](#), [Appendix B](#), and [Appendix C](#), we demonstrate the decomposition process in [Appendix D](#).

## 6. Computational experiments and results

This section first provides an adaptable approach to generate data for *MALBP - W* using benchmark data generators from the literature. Second, to evaluate the performance of proposed MDP models and algorithmic improvements, we conduct extensive computational experiments using the generated instances for *MDP<sup>Ro</sup>* and *MDP<sup>Sto</sup>*. Finally, we formulate managerial recommendations regarding the benefits of using the dynamic task assignment after comparing it to model-dependent and fixed task assignments. The problems are solved with IBM ILOG CPLEX Optimization Studio V12.10. The experiments were run on an Intel(R) Core(TM) i7-8650U CPU @ 1.90GHz - 2.11 GHz processor with 32 GB of RAM in MS Windows 10 Pro (64 bit) operational system.

### 6.1. Instances generation

To perform computational experiments, we adapt the data generator proposed by [Otto et al. \(2013\)](#) to the specificity of  $MALBP - W$ . Each of our instances merges  $\mathcal{I}$  consecutive instances of [Otto et al. \(2013\)](#). For example, our first generated instance contains the data of  $\mathcal{I}$  first instances of [Otto et al. \(2013\)](#) and has  $\mathcal{I}$  product models with different processing times and precedence graphs. The second instance contains the data of  $\{2 \dots \mathcal{I}+1\}$  first instances, and so on. We generate our instances using the set of benchmark instances with 20 tasks from [Otto et al. \(2013\)](#). Note that product models have different task processing times and precedence relationships. Naturally, our approach can handle the case of products with the same precedence graphs. Equipment costs at each station are generated randomly using a uniform distribution in the range  $[100, 300]$ . Three different values for workers' salary are considered: less, within, and more than the range for equipment cost ( $\alpha = \{50, 200, 500\}$ ). Two different numbers of stations ( $S = \{2, 3\}$ ) and product models ( $I = \{2, 3\}$ ) are defined. To consider smaller size instances, 8, 10, and 15 tasks among the 20 tasks of the instances from [Otto et al. \(2013\)](#) are selected randomly ( $O = \{8, 10, 15\}$ ). The instances' sizes are determined by the 3-tuple  $(\mathcal{I}, \mathcal{S}, \mathcal{O})$ , where  $\mathcal{I}$ ,  $\mathcal{S}$ , and  $\mathcal{O}$  represent the number of product models, stations, and tasks, respectively. To generate the compatibility matrix,  $R_{oe}$  is set to 1 with probability  $\frac{\bar{c}_e}{\bar{c}}$  (and 0 otherwise), where  $\bar{c}_e$  is the average cost of equipment  $e$  (over all stations), and  $\bar{c}$  is the average equipment cost (over all equipment and stations). The takt time in the instances of [Otto et al. \(2013\)](#) is set to 1000. In  $MALBP - W$ , several workers may perform tasks in a station, and the processing time decreases with a higher number of workers. Therefore, we use a lower takt time than the one given in [Otto et al. \(2013\)](#). Here, according to some initial tests, the takt time takes a value that provides "proper" results, i.e., feasible and with more than one worker per station.

To cover different production situations, several classes of instances are considered. These classes are mainly distinguished based on the set of tasks required for products, the ratio

between products' task processing times, the restrictions on the product models order, and the type of transition matrix. The class characteristics are explained below.

**Single/different set of tasks for product models:** having the same set of tasks  $O$  for all product models (class  $\#same$ ) as [Otto et al. \(2013\)](#) raises no issues. To have different sets of tasks  $O_i$  for product models (class  $\#diff$ ), we initially set  $O_i$  to  $O$ , and we randomly eliminate 40% to 60% tasks in each item specific set of task  $O_i$ .

**Ratio between products' task processing times:** the total processing times per product are almost close to each other, see [Otto et al. \(2013\)](#) (class  $\#1$ ). To differentiate this ratio, we select the product model with maximum total processing time, the so-called "bottleneck product" and denote it as ( $i = lux$ ). The bottleneck product can be a luxury product that is rarely produced in a given production period. In the other two classes, the total processing time for product  $lux$  remains fixed, but for all other products, it is either divided by 1.5 (class  $\#1.5$ ) or by 2 (class  $\#2$ ). We consider another class (class  $\#diverse$ ) for instances with three product models ( $I = 3$ ). The total processing time for product  $lux$  remains fixed, while it is divided by 1.5 for the product with the second-highest total processing time, and by 2 for the product with the third-highest total processing time.

**Restrictions on the number of units of product models:** As mentioned, the number of units of an item  $i$  in a picture of the line is restricted to  $u_i$ , as well as the limit on the number of consecutive units of an item  $i$  in the product order is restricted to  $u'_i$ . In the non-restricted case (denoted by  $\#non - rest$ ), there is no restriction on the number of successive units of any product models in the order ( $u'_i = \infty$  for all  $i \in \mathcal{I}$ ), and there is no restriction on the pictures of the line ( $u_i = S$  for all  $i \in \mathcal{I}$ ). In the restricted class  $\#rest - 1$ , the order cannot have more than  $S$  (the number of stations) successive units of the same product model  $i$  ( $u'_i = S$  for all  $i \in \mathcal{I}$ ), and similarly,  $u_i = S$  for all  $i \in \mathcal{I}$ . In the other restricted class  $\#rest - 2$ , the order cannot have more than  $S$  (the number of stations) successive units of the bottleneck product  $lux$  ( $u'_{lux} = S$  and  $u'_i = \infty, \forall i \neq lux$ ), and  $u_i = S$  for all  $i \in \mathcal{I}$ .

The class  $\#rest - 2$  aims to analyze the impact of the bottleneck product  $lux$  compared to the class  $\#rest - 2$  where all products are similarly restricted. In the last restricted class  $\#rest - 3$ , the order cannot have more than  $S - 1$  successive units of the bottleneck product  $lux$  ( $u'_{lux} = S - 1$ ) and more than  $S$  successive units of other products ( $u'_i = S$  if  $i \neq lux$ ), and a picture of the line cannot have more than  $S - 1$  units of the bottleneck product  $lux$  and  $S$  units of others ( $u_{lux} = S - 1$  and  $u_i = S$  if  $i \neq lux$ ).

**Type of transition matrix:** For all classes, all computational tests are performed over both formulated and random transition matrices which are denoted as  $\#not - rand$  and  $\#rand$ , respectively. In the first case ( $\#not - rand$ ), the transition probability depends on the number of product model units already located in the line at the current state. Note that this number is limited by  $u_i$  which allows adjusting production to the demand. The probability  $Tr_a^{dd'}$  of moving from state  $d$  to state  $d'$  by taking action  $a$  is calculated by formula (14). Note that the summation of all  $Tr_a^{dd'}$  over all product models  $i \in \mathcal{I}$  is equal to 1.

$$Tr_a^{dd'} = \left\{ \frac{u_i - \sum_{s=1}^{S-1} F_{isd}}{\sum_{i \in \mathcal{I}} u_i - \sum_{i \in \mathcal{I}} \sum_{s=1}^{S-1} F_{isd}} \mid F_{i1d'} = 1 \right\} \quad d, d' \in \mathcal{D}, a \in A_d, i \in \mathcal{I} \quad (14)$$

In the second case ( $\#rand$ ), each item  $i$  has a given probability  $r_i$  to enter the line, see formula (15). The values  $r_i$  are chosen randomly such that the sum of probabilities over  $|I|$  products is equal to 1. In practice, the values  $r_i$  may correspond to the demand ratio of products.

$$Tr_a^{dd'} = \{r_i \mid F_{i1d'} = 1, \sum_{i=1}^I r_i = 1\} \quad d, d' \in \mathcal{D}, a \in A_d \quad (15)$$

Notice that, when there are restrictions on successive units of product models in the order (classes  $\#rest - 1$ ,  $\#rest - 2$ , and  $\#rest - 3$ ), if the product model  $i$  with transition probability  $Tr_a^{dd'} \neq 0$  cannot enter the line by moving from state  $d$  to state  $d'$  through the action  $a$ , then  $Tr_a^{dd'} = 0$ . Therefore, the transition probabilities of other product models  $i' \neq i$  entering the line sum-up with  $Tr_a^{dd'} / (|I| - 1)$  to obtain the sum of probabilities equal

to 1. This means that the probability of other product models entering the line increases with the same ratio.

For each size and each class, 5 instances from all instances of [Otto et al. \(2013\)](#) are randomly selected, and it leads to a total number of instances equal to almost 2500.

## 6.2. Analysis of the MDP models

This subsection evaluates the performance of the proposed stochastic and robust MDP models, with reduction rules and decomposition algorithm in terms of solution quality, the number of generated actions and states, and computational time.

Tables 1 and 2 show how reduction rules and the decomposition algorithm eliminate redundant actions and states. The computational times of pre-processing and MILP solving of  $MALBP - W^{Dyn}$  (stochastic and robust) are also given in these tables. Solving *MILPs* takes more time than for  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$ , since more constraints and variables are generated which use a lot of memory. Table 2 is categorized by different classes of instances in terms of products' processing time variation. In these tables,  $|A0|$  refers to the number of all generated initial actions,  $|A|$  the number of actions remaining after applying reduction rules, and finally  $|A1|$  the number of actions imported to *MILPs* for solving the problem after aggregation during the decomposition. Similarly,  $|D0|$ ,  $|D|$ , and  $|D1|$  refer to the number of all initial states, the reduced number of states, and the final number of states importing to *MILPs*, respectively. In both tables, we can see the reduction process and decomposition algorithm decreased the total number of actions and states, effectively. Decomposition becomes more effective when the number of stations and the number of decompositions/pictures ( $PN$ ) increase.

Table 1 shows that the number of states and actions grows exponentially with the number  $O$  of tasks, and they are more sensitive to the number of stations ( $S$ ) than to the number of product models ( $I$ ). While the execution time increases with the size of the model, it remains

Table 1: The number of actions and states, and the computational times based on the size of instances.

Size (I,S,O)	N° actions			N° states			CPU time (s)		
	A1	A	A0	D1	D	D0	Pre-processing	MILP <sup>Ro</sup>	MILP <sup>Sto</sup>
(3,2,10)	<b>856</b>	950	1852	<b>79</b>	102	579	<b>5.5</b>	<b>2.4</b>	<b>6.2</b>
(3,2,15)	<b>2541</b>	2603	4677	<b>125</b>	161	1317	<b>68.0</b>	<b>3.5</b>	<b>82.0</b>
(2,3,10)	<b>3753</b>	5371	6542	<b>218</b>	324	960	<b>754.2</b>	<b>319.5</b>	<b>457.4</b>

reasonable. The main issue with this approach is the time required to build the model and memory consumption.

Table 2 shows that the numbers of actions and states increase with the diversity of the total processing time of products. This behavior is expected since more actions and states are generated with a smaller processing times and the takt time remains fixed.

Table 2: The number of actions and states, and the computational times based on process time variety of products.

	N° actions			N° states			CPU time (s)		
	A1	A	A0	D1	D	D0	Pre-processing	MILP <sup>Ro</sup>	MILP <sup>Sto</sup>
#1	<b>1010</b>	1467	2867	<b>88</b>	154	850	<b>21.8</b>	<b>12.3</b>	<b>27.5</b>
#1.5	<b>2687</b>	3261	4674	<b>153</b>	201	956	<b>327.1</b>	<b>43.1</b>	<b>110.7</b>
#diverse	<b>2770</b>	3427	4757	<b>157</b>	209	993	<b>357.7</b>	<b>50.4</b>	<b>118.3</b>
#2	<b>3066</b>	3745	5131	<b>164</b>	219	1009	<b>397.0</b>	<b>328.0</b>	<b>471.0</b>

Table 3 provides the percentage of cost-saving from the dynamic task assignment compared to the proposed model-dependent and fixed task assignment, as well as the cost-saving from the model-dependent task assignment compared to the fixed one using the robust MDP model. Moreover, this table provides the expected cost saving of  $MALBP - W^{Dyn}$  compared to  $MALBP - W^{Md}$  using the stochastic MDP model. We report only the gap between dynamic and model-dependent task assignment in the stochastic model as  $MALBP - W^{Fix}$  requires too many constraints and it is time consuming to solve. The gaps between  $MALBP -$



$W^{Md}$  and  $MALBP - W^{Fix}$  in  $MDP^{Sto}$  (2.54%, 1.68%, and 1.00%) are lower than the gaps found by  $MDP^{Rob}$  (4.50%, 3.64%, and 1.93%), since except for the worst picture of the line (worst takt), in other pictures almost the same number of workers are required in the line. The analogy of  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$  for a given set of products order entering the line has been discussed in detail in (Hashemi-Petroodi et al., 2022).

Table 3: Solution quality of fixed, model-dependent, and dynamic task assignments.

Size (I,S,O)	MDP <sup>Ro</sup>			MDP <sup>Sto</sup>
	Dyn/Md	Dyn/Fix	Md/Fix	Dyn/Md
	Gap (%)	Gap (%)	Gap (%)	Gap (%)
(3,2,10)	4.50	5.79	1.58	2.54
(3,2,15)	3.64	6.98	3.60	1.68
(2,3,10)	1.93	2.74	1.01	1.00

### 6.3. Case study

This section provides numerical results on mini-bus assembly line case study presented by (Finco et al., 2021, 2019). To solve this realistic instance with 80 tasks, we embed the proposed model ( $MALBP - W^{Dyn}$ ) in a heuristic framework. The case study presented by (Finco et al., 2021, 2019) consider a single-model. We adapt the case study to our problem by introducing 2 product models with the same precedence graph. The processing times are set to the values given in Finco et al. (2021) for the first model, and we divide these processing times by two for the second product model. Since the dynamic policy results the same as model-dependent in non-restricted case, we consider restricted case in which only one product unit with larger process time can exist in each pair of adjacent product units. The assembly line has 10 stations with a cycle time of 1000. We set the costs of workers to 500, and generate the costs of equipment in the same manner as explained earlier.

The heuristic starts with the search for an initial solution with task, worker and equipment assignments. To get this initial solution, we solve  $MALBP - W^{Md}$  with the fix-and-optimize

approach proposed in (Hashemi-Petroodi et al., 2022). Note that the solution of  $MALBP - W^{Md}$  provides a feasible but non-necessarily optimal solution for the case with dynamic task assignments. The heuristic solves  $MALBP - W^{Dyn}$  for each two adjacent stations, and it takes into account the set of tasks assigned to the two stations in the current solution. More precisely, the heuristic starts with the two first stations, then the pair of third and fourth stations, and so on. Therefore, at each iteration, the heuristic solves  $MALBP - W^{Dyn}$  for two stations with the approach proposed in this paper. The considered sets of tasks are the ones assigned to the two corresponding stations in the initial solution of  $MALBP - W^{Md}$ . The initial number of workers used in the two stations in this solution is equal to the upper bound of the total number of workers as explained in the third point of Section 5.2. In the end, if we have  $S$  stations, we solve  $S/2$  sub- $MALBP - W^{Dyn}$  and sum-up all objective values of  $S/2$  sub-problems. Note that in case of having an odd total number of stations, three consecutive stations can be taken into account in one of the sub-problems.

Tables 4 and 5 present the results of the case study in terms of the algorithm’s performance and computational time. Table 4 shows that the proposed heuristic is able to handle a large size instance as in the real case study. The heuristic consumes significant CPU time. Since the design and balancing problem has a strategic nature, it is acceptable. Such decisions are not frequent and their impact is long-term. However, searching for a more efficient and faster approach is a promising research direction. The number of states/actions corresponds to the total number of states/actions for the entire assembly line. Table 5 shows the cost savings of using dynamic task assignment compared to the model-dependent and fixed cases. Table 5 shows that dynamic assignment provides a lower expected cost compared to the model-dependent one.

The proposed heuristic is able to solve large-size instances with a large number of stations and a relatively small cycle time. In such a case, the number of states and actions remains at an acceptable level. However, if the number of stations decreases together with an increase

in cycle time, it results in a high number of tasks assigned to stations in the initial solution. Then, solving sub-problems fails due to the out-of-memory issue. Overcoming this difficulty is a promising research direction. To do so, an approximated dynamic programming or a policy-based algorithms can be taken into account.

Table 4: The number of actions and states, and the computational times for the case study.

N° actions			N° states			CPU time (s)		
A1	A	A0	D1	D	D0	Pre-processing	MILP <sup>Ro</sup>	MILP <sup>Sto</sup>
<b>22541</b>	27473	37054	<b>1347</b>	1648	4782	<b>4933.6</b>	<b>2105.1</b>	<b>2866.2</b>

Table 5: The impact of classes of instances on the solution quality of fixed, model-dependent, and dynamic task assignments for the case study.

MDP <sup>Ro</sup>			MDP <sup>Sto</sup>
Dyn/Md	Dyn/Fix	Md/Fix	Dyn/Md
Gap (%)	Gap (%)	Gap (%)	Gap (%)
3.4	11.6	8.2	1.3

Table 6 shows the number of task re-assignments in three different scenarios: dynamic, model-dependent, and fixed. Table 6 also provides the minimum, average, and maximum numbers of workers required in the line (over all possible states of the line) in the robust model. Herein, the number of re-assigned tasks is the total number of tasks, among 80 tasks, which are re-assigned over all states of the line. The number of task re-assignments is slightly higher for the dynamic scenario compared to the model-dependent one. While task re-assignment can be problematic in practice, such a small difference does not present an issue, especially when considering the total cost reduction associated with the dynamic task assignment.

Table 6: Re-assignment of tasks and workers in  $MDP^{Dyn}$ ,  $MDP^{Md}$ , and  $MDP^{Fix}$

Problem	N° of re-assigned tasks	Min. N° worker	Avr. N° worker	Max. N° worker
$MALBP - W^{Dyn}$	27	14	16.6	21
$MALBP - W^{Md}$	22	14	16.8	22
$MALBP - W^{Fix}$	-	16	19.1	24

#### 6.4. Managerial insights

Herein, we evaluate the impact of the cost of worker ( $\alpha$ ) and different classes of instances on the cost saving of  $MALBP - W^{Dyn}$  compared to  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$ , as well as on the optimal number of workers and equipment cost. All analyzes in this section are performed over small size instances solved in Section 6.2, because the instances are solved optimally.

Table 7 evaluates the cost-saving of  $MALBP - W^{Dyn}$  compared to  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$  based on the cost of workers ( $\alpha$ ). The gaps between the problems increase with the cost of workers. For instance, the average gap between dynamic task assignment and model-dependent increases from 3.12% to 5.94% when the cost of workers increases from 50 to 500. This observation was expected because a lower ratio between the cost of workers and equipment leads to a production system with more flexibility to re-assign tasks and workers. As the cost of workers is large compared to the equipment cost, it is desirable to duplicate equipment and re-assign the tasks and workers when needed.

Table 8 shows the performance of the  $MALBP - W^{Dyn}$  compared to  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$  for different classes of instances. The gap between the models is large when: 1) products have the same set of tasks (i.e., for Dyn/Md, 3.77% versus 3.02%); 2) product models have processing times with large variance (i.e., for Dyn/Md, 4.97% versus 1.09%); 3) the processing time of a product (a luxury product) is significantly larger than

Table 7: The impact of the cost of workers on the solution quality of fixed, model-dependent, and dynamic task assignments.

$\alpha$	$\mathbf{MDP}^{Ro}$			$\mathbf{MDP}^{Sto}$
	Dyn/Md	Dyn/Fix	Md/Fix	Dyn/Md
	Gap (%)	Gap (%)	Gap (%)	Gap (%)
<b>50</b>	3.12	4.46	1.78	1.68
<b>200</b>	4.89	6.79	2.29	2.40
<b>500</b>	5.94	8.02	2.60	3.13

the processing times of other products (i.e., for Dyn/Md, 4.56% versus 3.83%); 4) the user defines restrictions on the number of luxury product units in the whole order of products ( $u'_{lux}$ ) (i.e., for Dyn/Md, 5.48% versus 4.99%); 5) the user gives restrictions on the number of all product models in the whole the order of products ( $u'_i$ ) (i.e., for Dyn/Md, 4.99% versus 3.48%); 6) the user limits the number of product models units in the picture of the line ( $u_i$ ) (i.e., for Dyn/Md, 3.48% versus 0%).

Note that when there is no restriction on the order,  $MALBP - W^{Dyn}$  provides the same result as  $MALBP - W^{Md}$  for both robust and stochastic models since the worst takt corresponds to the situation where the bottleneck product (which has the longest total processing time) is in all stations. When the bottleneck product can be on all stations, the worst-case number of workers is the one computed by the model-dependent task assignment. For the classes of instances with restrictions and process time variations, dynamic task assignment leads to better solutions since the task assignment can be adapted to the product mix on the station at each takt (e.g. when a mix of bottleneck and non-bottleneck products are in stations), whereas model-dependent task assignment forces the user to keep model specified task assignment for all takts.

Tables 9 and 10 show the impact of the cost of worker on the equipment cost and on the number of workers required for the worst-case (in  $MDP^{Ro}$ ) and the expected value (in

Table 8: The impact of classes of instances on the solution quality of fixed, model-dependent, and dynamic task assignment.

MALBP-W	MDP <sup>Ro</sup>			MDP <sup>Sto</sup>	
	Dyn/Md	Dyn/Fix	Md/Fix	Dyn/Md	
	Gap (%)	Gap (%)	Gap (%)	Gap (%)	
#same	3.77	6.40	2.59	2.10	
#diff	3.02	4.39	1.72	1.96	
#1	1.09	3.23	2.21	1.38	
#1.5	3.83	5.50	1.93	1.73	
#diverse	4.97	7.03	2.53	2.09	
#2	4.56	6.35	2.08	3.00	
#non-rest	0.00	2.30	2.30	0.00	
#rest-1	4.99	6.99	2.30	2.51	
#rest-2	5.48	7.01	2.30	3.28	
#rest-3	3.48	5.27	2.01	2.33	

$MDP^{Sto}$ ), respectively. Table 9 shows that increasing  $\alpha$  increases the cost of equipment by duplicating equipment pieces in stations, while the number of required workers decreases. In addition, the number of workers in the worst takt of dynamic task assignment is less than the one in model-dependent and fixed cases, whereas the cost of equipment increases from the fixed task assignment to model-dependent and dynamic cases. To reduce the number of workers, dynamic and model-dependent cases rely on more capable equipment pieces. While these equipment pieces are more expensive, more types of tasks can be re-assigned. A higher cost of workers ( $\alpha = \{200, 500\}$ ) provides almost the same number of workers for the worst takt (in  $MDP^{Ro}$ ) which was expected since we could observe almost similar results in our previous work related to the model-dependent and fixed cases (Hashemi-Petroodi et al., 2022). Note that such an observation might be true with larger size instances.

Table 10 shows that the impact of  $\alpha$  on the expected number of workers and on the equipment cost is not that considerable. As mentioned earlier, except for the worst takt, the number of workers is almost the same in other takts when we minimize the expected

Table 9: The impact of the cost of workers on the cost of equipment and the number of workers in the robust model  $MDP^{Ro}$ .

$\alpha$	<b>MALBP-W<sup>Dyn</sup></b>		<b>MALBP-W<sup>Md</sup></b>		<b>MALBP-W<sup>Fix</sup></b>	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker	Eq. Cost	N° Worker
<b>50</b>	407.6	5.36	406.8	5.74	407.8	6.01
<b>200</b>	416.0	5.32	413.8	5.68	411.9	5.90
<b>500</b>	416.0	5.32	413.8	5.68	411.9	5.90

total cost of workers and equipment over all the taks. Contrarily to  $MILP^{Ro}$ ,  $MILP^{Sto}$  provides lower equipment costs for dynamic task assignment than for the model-dependent one.  $MILP^{Sto}$  focuses only on the worst takt, and it invests in expensive equipment to reduce the number of workers in the worst takt. On the contrary,  $MILP^{Sto}$  provides almost the same expected number of workers for both dynamic and model-dependent task assignments, but lower cost of equipment for the dynamic case.

Table 10: The impact of the cost of workers on the cost of equipment and the number of workers in the stochastic model  $MDP^{Sto}$ .

$\alpha$	<b>MALBP-W<sup>Dyn</sup></b>		<b>MALBP-W<sup>Md</sup></b>	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker
<b>50</b>	413.0	4.57	420.4	4.73
<b>200</b>	418.0	4.55	424.8	4.72
<b>500</b>	418.0	4.53	424.8	4.69

Tables 11 and 12 draw the equipment cost and the number of workers for different classes of instances. Manufacturing companies may use fewer workers but face slightly higher costs of equipment, when: 1) products require different sets of tasks, 2) they can impose restrictions on product orders. In addition, dynamic task assignment reduces the number of workers in all classes of instances except for when there is no restriction on the number of product models in the line. However, dynamic task assignment leads to large equipment costs. The model-dependent task assignment requires fewer workers compared to the fixed one. In some cases,

the equipment cost decreases (besides decreasing the number of workers) in the dynamic task assignments since forcing fixed or model-dependent task assignments may require equipment pieces with more capabilities (e.g., class #1.5). This point is also valid when we compare the model-dependent and fixed cases since the model-dependent case can also benefit from the flexibility to assign the same task to different stations.

Table 11: The impact of the classes of instances on the cost of equipment and the number of workers in the robust model  $MDP^{Ro}$ .

	MALBP-W <sup>Dyn</sup>		MALBP-W <sup>Md</sup>		MALBP-W <sup>Fix</sup>	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker	Eq. Cost	N° Worker
#same	435.6	5.6	433.0	6.0	433.6	6.4
#diff	388.7	5.0	389.9	5.4	389.1	5.5
#1	396.3	6.0	415.6	6.2	423.3	6.4
#1.5	427.7	5.4	416.8	5.7	416.8	6.0
#diverse	386.5	4.6	389.0	4.9	372.0	5.2
#2	429.6	5.1	416.8	5.7	418.5	5.9
#non-rest	411.3	6.1	411.4	5.9	411.3	6.1
#rest-1	412.1	5.5	411.4	5.9	411.3	6.1
#rest-2	412.1	5.2	411.4	5.9	411.3	6.1
#rest-3	412.1	5.2	411.4	5.5	411.3	5.8

Table 12: The impact of the classes of instances on the cost of equipment and the number of workers in the stochastic model  $MDP^{Sto}$ .

	MALBP-W <sup>Dyn</sup>		MALBP-W <sup>Md</sup>	
	Eq. Cost	N° Worker	Eq. Cost	N° Worker
#same	434.5	4.9	434.5	5.1
#diff	398.2	4.2	412.1	4.3
#1	422.8	5.5	424.5	5.6
#1.5	421.9	4.4	428.9	4.6
#diverse	386.5	3.7	391.5	3.9
#2	424.2	4.2	437.8	4.4
#non-rest	414.7	4.9	414.7	4.9
#rest-1	416.3	4.7	414.7	4.9
#rest-2	416.3	4.2	414.7	4.9
#rest-3	417.0	4.4	414.7	4.6

To improve the line's efficiency, manufacturing companies must consider the flexibility



measures studied in this paper: walking workers, dynamic and model-dependent task assignments. However, companies must note that such measures may cause ergonomic issues for workers. Several studies focus on the ergonomic aspect of production lines. For example, [Otto and Battaia \(2017\)](#) discusses the ergonomic issues at the planning stage of the assembly lines. Stronger interactions between production managers, ergonomists, and operations researchers are required. Employing a dynamic task assignment as a response to frequent changes in product orders may cause stress and fatigue among workers. The decision makers should take into account this factor. Moreover, as an attractive research topic, productivity of the line with walking workers can be optimized in addition to the cost. For example, some advantages of using walking workers related to the improvement in motivation, accountability, and responsibility of workers are discovered in [Bischak \(1996\)](#).

## 7. Conclusion

This study addresses a multi-manned mixed-model assembly line balancing problem with walking workers ( $MALBP - W$ ). We evaluate the impact of dynamic task assignment to stations, where tasks can be re-assigned at the end of each takt depending on the product models located in the line and already performed tasks. Besides, we compare the proposed dynamic task assignment with model-dependent and fixed task assignments. In the model-dependent task assignment strategy, each product model can have its own task assignment which remains fixed, while the fixed task assignment case assumes the same task assignment for all product models. We assume an infinite unknown sequence of product models entering the line. Two objectives are considered. The first objective is to minimize the expected total cost over all possible takts. The second goal consists in minimizing the total cost in the worst takt.

At the design stage, the number of workers to hire is determined and the required equipment is installed at stations where each piece of equipment can be duplicated if it is needed

at other stations. At the operational stage, at the end of each takt, workers can move from one station to another while tasks can be re-assigned to other stations.

In view of the problem's dynamic and stochastic nature, a Markov Decision Process (MDP) model is developed. To the best of our knowledge, this approach has never been applied to MALBP before. To accelerate the solution process, several reduction rules and a decomposition algorithm have been formulated to solve instances efficiently. After the pre-processing of the MDP model, the problem is solved using the proposed stochastic and robust mixed-integer linear programming (MILP) models. Moreover, a heuristic technique using the proposed approach is developed to solve larger/real size cases. The case study of a mini-bus assembly line found in the literature has been efficiently solved.

The computational results show that dynamic task assignment leads to larger cost savings compared to model-dependent and fixed task assignments studied in the literature. Extensive managerial insights are given and discussed for different classes of instances. For example, dynamic task assignment performs better in a company where one or some luxury product models are produced in a limited quantity compared to other common product models. An example can be one or several luxury car model(s) which is(are) produced at a certain time period. However, dynamic task assignment performs identically to the model-dependent assignment in the case when total processing times of all products are almost the same and there is no restriction on the number of a product model's units produced uninterruptedly.

This study is our initial attempt to apply MDP using mathematical programming to a line balancing problem. Our study has the potential to be improved, especially on the methodological side. Recent studies in the literature which integrate MDP and combinatorial optimization approaches show a promising perspective for this research as well. The proposed approach performs well when the line contains a large number of stations and the cycle time is relatively small. Such combination generates a reasonable number of states and actions in the MDP model. Therefore, our main future research focus is to develop an optimization

approach that will handle cases with a high number of states and actions.

A future work prospect also consists in developing an approximate dynamic programming approach to solve large-size instances efficiently. The main goal is to decrease the number of actions, which causes an out-of-memory problem during the solution process.

An important feature that can be taken into account in future research concerns the ergonomic impact of dynamic task assignment on workers: side effects, stress, overload, de-routinization of work tasks, etc. Moreover, as discussed in the managerial insights, taking into account workers' experience would open some interesting future research avenues related to workers' learning, multiple skills, cross-training, etc.

## **Acknowledgement**

The authors of this paper would like to thank the Region Pays de la Loire, France ([www.paysdelaloire.fr](http://www.paysdelaloire.fr)) which supported financially this research. We also would like to thank the project ASSISTANT (<https://assistant-project.eu/>) that is funded by the European Commission, under grant agreement number 101000165, H2020-ICT -38-2020, artificial intelligence for manufacturing.

## **References**

- Ahiska, S.S., Appaji, S.R., King, R.E., Warsing Jr, D.P., 2013. A markov decision process-based policy characterization approach for a stochastic inventory control problem with unreliable sourcing. *International Journal of Production Economics* 144, 485–496.
- Ahluwalia, V.S., Steimle, L.N., Denton, B.T., 2021. Policy-based branch-and-bound for infinite-horizon multi-model markov decision processes. *Computers & Operations Research* 126, 105108.

- AkpıNar, S., Bayhan, G.M., Baykasoglu, A., 2013. Hybridizing ant colony optimization via genetic algorithm for mixed-model assembly line balancing problem with sequence dependent setup times between tasks. *Applied Soft Computing* 13, 574–589.
- Al-Zuheri, A., Luong, L., Xing, K., 2016. Developing a multi-objective genetic optimisation approach for an operational design of a manual mixed-model assembly line with walking workers. *Journal of intelligent manufacturing* 27, 1049–1065.
- Alagoz, O., Ayvaci, M.U., Linderoth, J.T., 2015. Optimally solving markov decision processes with total expected discounted reward function: Linear programming revisited. *Computers & Industrial Engineering* 87, 311–316.
- Alghazi, A., Kurz, M.E., 2018. Mixed model line balancing with parallel stations, zoning constraints, and ergonomics. *Constraints* 23, 123–153.
- Askin, R., Zhou, M., 1997. A parallel station heuristic for the mixed-model production line balancing problem. *International Journal of Production Research* 35, 3095–3106.
- Battaïa, O., Delorme, X., Dolgui, A., Hagemann, J., Horlemann, A., Kovalev, S., Malyutin, S., 2015. Workforce minimization for a mixed-model assembly line in the automotive industry. *International Journal of Production Economics* 170, 489–500.
- 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., 2009. Balancing assembly lines with variable parallel workplaces: Problem definition and effective solution procedure. *European Journal of Operational Research* 199, 359–374.

- Bengio, Y., Lodi, A., Prouvost, A., 2020. Machine learning for combinatorial optimization: a methodological tour d’horizon. *European Journal of Operational Research* .
- Biele, A., Mönch, L., 2018. Hybrid approaches to optimize mixed-model assembly lines in low-volume manufacturing. *Journal of Heuristics* 24, 49–81.
- Bischak, D.P., 1996. Performance of a manufacturing module with moving workers. *IIE transactions* 28, 723–733.
- Boctor, F., Danjou, F., Giard, V., et al., 2000. Rank changes in production/assembly lines: impact and analysis. Université Paris I, Panthéon-Sorbonne, Institut d’administration des entreprises.
- Boysen, N., Fliedner, M., Scholl, A., 2008. Assembly line balancing: Which model to use when? *International Journal of Production Economics* 111, 509–528.
- Boysen, N., Fliedner, M., Scholl, A., 2009. Production planning of mixed-model assembly lines: overview and extensions. *Production Planning and Control* 20, 455–471.
- Buchholz, P., Scheftelowitsch, D., 2019. Light robustness in the optimization of markov decision processes with uncertain parameters. *Computers & Operations Research* 108, 69–81.
- Bukchin, J., Dar-El, E.M., Rubinovitz, J., 2002. Mixed model assembly line design in a make-to-order environment. *Computers & Industrial Engineering* 41, 405–421.
- Bukchin, Y., Rabinowitch, I., 2006. A branch-and-bound based solution approach for the mixed-model assembly line-balancing problem for minimizing stations and task duplication costs. *European Journal of Operational Research* 174, 492–508.

- Choi, G., 2009. A goal programming mixed-model line balancing for processing time and physical workload. *Computers & Industrial Engineering* 57, 395–400.
- Cortez, P.M., Costa, A.M., 2015. Sequencing mixed-model assembly lines operating with a heterogeneous workforce. *International Journal of Production Research* 53, 3419–3432.
- Delorme, X., Dolgui, A., Kovalev, S., Kovalyov, M.Y., 2019. Minimizing the number of workers in a paced mixed-model assembly line. *European Journal of Operational Research* 272, 188–194.
- Dimitriadis, S.G., 2006. Assembly line balancing and group working: A heuristic procedure for workers' groups operating on the same product and workstation. *Computers & Operations Research* 33, 2757–2774.
- Dolgui, A., Kovalev, S., Kovalyov, M.Y., Malyutin, S., Soukhal, A., 2018. Optimal workforce assignment to operations of a paced assembly line. *European Journal of Operational Research* 264, 200–211.
- Finco, S., Abdous, M.A., Battini, D., Calzavara, M., Delorme, X., 2019. Assembly line design with tools vibration. *IFAC-PapersOnLine* 52, 247–252.
- Finco, S., Abdous, M.A., Calzavara, M., Battini, D., Delorme, X., 2021. A bi-objective model to include workers' vibration exposure in assembly line design. *International Journal of Production Research* 59, 4017–4032.
- Giard, V., Jeunet, J., 2010. Optimal sequencing of mixed models with sequence-dependent setups and utility workers on an assembly line. *International Journal of Production Economics* 123, 290–300.

- Hashemi-Petroodi, S.E., Dolgui, A., Kovalev, S., Kovalyov, M.Y., Thevenin, S., 2020a. Work-force reconfiguration strategies in manufacturing systems: a state of the art. *International Journal of Production Research* , 1–24.
- Hashemi-Petroodi, S.E., Thevenin, S., Kovalev, S., Dolgui, A., 2020b. The impact of dynamic tasks assignment in paced mixed-model assembly line with moving workers, in: *IFIP International Conference on Advances in Production Management Systems*, Springer. pp. 509–517.
- Hashemi-Petroodi, S.E., Thevenin, S., Kovalev, S., Dolgui, A., 2021. Model-dependent task assignment in multi-manned mixed-model assembly lines with moving workers. Submitted-under review .
- Hashemi-Petroodi, S.E., Thevenin, S., Kovalev, S., Dolgui, A., 2022. Model-dependent task assignment in multi-manned mixed-model assembly lines with walking workers. *Omega* , 102688.
- Hwang, R., Katayama, H., 2010. Integrated procedure of balancing and sequencing for mixed-model assembly lines: a multi-objective evolutionary approach. *International Journal of Production Research* 48, 6417–6441.
- Kaiser, U., Schreier, M., Janiszewski, C., 2017. The self-expressive customization of a product can improve performance. *Journal of Marketing Research* 54, 816–831.
- Kamrani, M., Srinivasan, A.R., Chakraborty, S., Khattak, A.J., 2020. Applying markov decision process to understand driving decisions using basic safety messages data. *Transportation Research Part C: Emerging Technologies* 115, 102642.
- Kang, Z., Catal, C., Tekinerdogan, B., 2020. Machine learning applications in production lines: A systematic literature review. *Computers & Industrial Engineering* 149, 106773.

- Karimi-Mamaghan, M., Mohammadi, M., Meyer, P., Karimi-Mamaghan, A.M., Talbi, E.G., 2022. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. *European Journal of Operational Research* 296, 393–422.
- Kellegöz, T., 2017. Assembly line balancing problems with multi-manned stations: a new mathematical formulation and gantt based heuristic method. *Annals of Operations Research* 253, 377–404.
- Koren, Y., Heisel, U., Jovane, F., Moriwaki, T., Pritschow, G., Ulsoy, G., Van Brussel, H., 1999. Reconfigurable manufacturing systems. *Annals of the CIRP* 48, 2.
- Kucukkoc, I., Li, Z., Karaoglan, A.D., Zhang, D.Z., 2018. Balancing of mixed-model two-sided assembly lines with underground workstations: A mathematical model and ant colony optimization algorithm. *International Journal of Production Economics* 205, 228–243.
- Kucukkoc, I., Zhang, D.Z., 2014a. Mathematical model and agent based solution approach for the simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *International Journal of Production Economics* 158, 314–333.
- Kucukkoc, I., Zhang, D.Z., 2014b. Simultaneous balancing and sequencing of mixed-model parallel two-sided assembly lines. *International Journal of Production Research* 52, 3665–3687.
- Li, J., Gao, J., 2014. Balancing manual mixed-model assembly lines using overtime work in a demand variation environment. *International Journal of Production Research* 52, 3552–3567.
- Li, M., Carter, A., Goldstein, J., Hawco, T., Jensen, J., Vanberkel, P., 2021. Determining



- ambulance destinations when facing offload delays using a markov decision process. *Omega* 101, 102251.
- Liu, W., Zheng, L., Sun, D., Liao, X., Zhao, M., Su, J., Liu, Y., 2012. Rfid-enabled real-time production management system for loncin motorcycle assembly line. *International Journal of Computer Integrated Manufacturing* 25, 86–99.
- Lopes, T.C., Michels, A.S., Sikora, C.G.S., Molina, R.G., Magatão, L., 2018. Balancing and cyclically sequencing synchronous, asynchronous, and hybrid unpaced assembly lines. *International Journal of Production Economics* 203, 216–224.
- Lopes, T.C., Pastre, G.V., Michels, A.S., Magatão, L., 2020. Flexible multi-manned assembly line balancing problem: Model, heuristic procedure, and lower bounds for line length minimization. *Omega* 95, 102063.
- Michels, A.S., Lopes, T.C., Sikora, C.G.S., Magatão, L., 2019. A benders’ decomposition algorithm with combinatorial cuts for the multi-manned assembly line balancing problem. *European Journal of Operational Research* 278, 796–808.
- Miltenburg, J., 1989. Level schedules for mixed-model assembly lines in just-in-time production systems. *Management Science* 35, 192–207.
- Moreau, C., Herd, K.B., 2010. To each his own? how comparisons with others influence consumers’ evaluations of their self-designed products. *Journal of Consumer Research* 36, 806–819.
- Morin, M., Gaudreault, J., Brotherton, E., Paradis, F., Rolland, A., Wery, J., Laviolette, F., 2019. Machine learning-based models of sawmills for better wood allocation planning. *International Journal of Production Economics* 222, 107508.

- Naderi, B., Azab, A., Borooshan, K., 2019. A realistic multi-manned five-sided mixed-model assembly line balancing and scheduling problem with moving workers and limited workspace. *International Journal of Production Research* 57, 643–661.
- Otto, A., Battaïa, O., 2017. Reducing physical ergonomic risks at assembly lines by line balancing and job rotation: A survey. *Computers & Industrial Engineering* 111, 467–480.
- Otto, A., Otto, C., Scholl, A., 2013. Systematic data generation and test design for solution algorithms on the example of salbpge for assembly line balancing. *European Journal of Operational Research* 228, 33–45.
- Özcan, U., Çerçioğlu, H., Gökçen, H., Toklu, B., 2010. Balancing and sequencing of parallel mixed-model assembly lines. *International Journal of Production Research* 48, 5089–5113.
- Patek, S.D., 2004. Policy iteration type algorithms for recurrent state markov decision processes. *Computers & Operations Research* 31, 2333–2347.
- Pavitsos, A., Kyriakidis, E., 2009. Markov decision models for the optimal maintenance of a production unit with an upstream buffer. *Computers & Operations Research* 36, 1993–2006.
- Qiu, H., Wang, S., Yin, Y., Wang, D., Wang, Y., 2022. A deep reinforcement learning-based approach for the home delivery and installation routing problem. *International Journal of Production Economics* 244, 108362.
- Şahin, M., Kellegöz, T., 2019. Balancing multi-manned assembly lines with walking workers: problem definition, mathematical formulation, and an electromagnetic field optimisation algorithm. *International Journal of Production Research* 57, 6487–6505.

- Saif, U., Guan, Z., Zhang, L., Zhang, F., Wang, B., Mirza, J., 2019. Multi-objective artificial bee colony algorithm for order oriented simultaneous sequencing and balancing of multi-mixed model assembly line. *Journal of Intelligent Manufacturing* 30, 1195–1220.
- Salari, N., Makis, V., 2017. Comparison of two maintenance policies for a multi-unit system considering production and demand rates. *International Journal of Production Economics* 193, 381–391.
- Samouei, P., Fattahi, P., 2018. An algorithm for integrated worker assignment, mixed-model two-sided assembly line balancing and bottleneck analysis. *Journal of Industrial and Systems Engineering* 11, 151–174.
- 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.
- Sikora, C.G.S., Lopes, T.C., Magatão, L., 2017. Traveling worker assembly line (re) balancing problem: Model, reduction techniques, and real case studies. *European Journal of Operational Research* 259, 949–971.
- Steimle, L.N., Ahluwalia, V.S., Kamdar, C., Denton, B.T., 2021. Decomposition methods for solving markov decision processes with multiple models of the parameters. *IIE Transactions* , 1–58.
- Tiacci, L., 2015. Simultaneous balancing and buffer allocation decisions for the design of mixed-model assembly lines with parallel workstations and stochastic task times. *International Journal of Production Economics* 162, 201–215.
- Tiacci, L., Mimmi, M., 2018. Integrating ergonomic risks evaluation through ocr index and balancing/sequencing decisions for mixed model stochastic asynchronous assembly lines. *Omega* 78, 112–138.

- Valenzuela, A., Dhar, R., Zettelmeyer, F., 2009. Contingent response to self-customization procedures: Implications for decision satisfaction and choice. *Journal of Marketing Research* 46, 754–763.
- Yu, X., Gao, S., Hu, X., Park, H., 2019. A markov decision process approach to vacant taxi routing with e-hailing. *Transportation Research Part B: Methodological* 121, 114–134.
- Zobel, C.W., Scherer, W.T., 2005. An empirical study of policy convergence in markov decision process value iteration. *Computers & operations research* 32, 127–142.
- Zufferey, N., 2016. Tabu search approaches for two car sequencing problems with smoothing constraints, in: *Metaheuristics for Production Systems*. Springer, pp. 167–190.

## Appendix A. Examples of dynamic task assignment in $MALBP - W^{Dyn}$

The line consists of two stations, and assembles two models:  $A$  and  $B$  which can enter the line in any order as an infinite unknown sequence. A picture of the line is denoted as  $(1-i, \dots, S-j)$ ,  $i, j \in \mathcal{I}$ . It determines the sequence of pairs station-product model in a certain takt. Thus, the only possible pictures of the line (Pic.) are  $(1-A, 2-B)$ ,  $(1-B, 2-B)$ ,  $(1-A, 2-A)$ , and  $(1-A, 2-B)$ . Notice that, we consider no restrictions on the picture of the line ( $u_A = u_B = S$ , where  $S = 2$ ), but the restricted number of the same product model in the whole order of products ( $u'_A = u'_B = S$ , where  $S = 2$ ). More precisely, we are not allowed to have more than two items from the same model in the whole order, which means that we cannot move from state possessing two of the same product model on the line to the state itself (from  $(1-A, 2-A)$  to  $(1-A, 2-A)$  or from  $(1-B, 2-B)$  to  $(1-B, 2-B)$ ).

Figure A.2 gives the precedence graphs and processing times for each product model with a common set of 5 tasks. Table A.13 presents the compatibility between equipment and tasks, and the cost of using the equipment at each station. The cost of each equipment is related to the number of tasks that it can perform. The cost of a worker is  $\alpha = 500$ , the takt time is  $C = 25$ , and at most  $l_{max} = 3$  workers can work at the same station simultaneously.

Table A.13: Compatibility between tasks and equipment, and the cost of equipment at each station.

	Task 1	Task 2	Task 3	Task 4	Task 5	Station 1	Station 2
<b>Equipment 1</b>	✓				✓	132	122
<b>Equipment 2</b>	✓		✓	✓	✓	172	148
<b>Equipment 3</b>	✓	✓	✓	✓	✓	224	200

Figure A.3 demonstrates an example of the optimal solution for  $MALBP - W^{Dyn}$  which is the objective of the current study. As one can see, the tasks are re-assigned in each takt considering four possible pictures of the line. Figure A.3 shows the tasks, equipment and workers assigned to the stations for each picture through some possible takts. It also points out the worst picture(s) marked in red, number of workers, used equipment, and total cost.

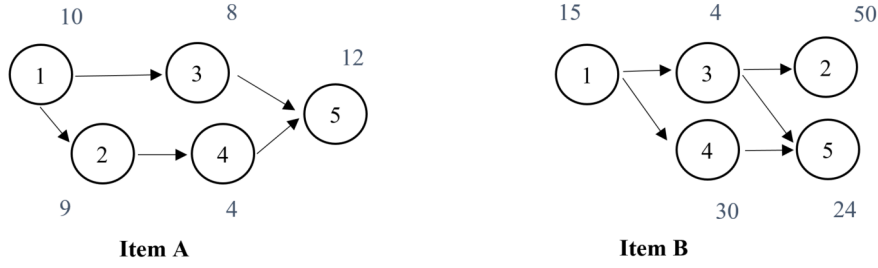


Figure A.2: The precedence graphs for each model.

Since the order of products is unknown, to handle all possible pictures of the line, the optimal solution of  $MALBP - W^{Dyn}$  requires 4 workers with total equipment and workforce cost of 2424.

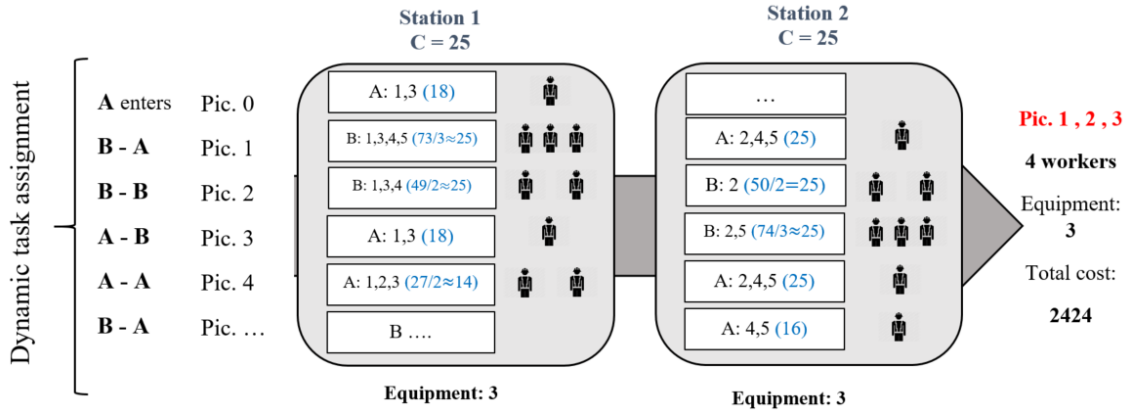


Figure A.3: The optimal solution of  $MALBP - W^{Dyn}$  in the simple example.

## Appendix B. Example of an MDP graph.

Here, an MDP graph for the example in [Appendix A](#) is given. [Figure B.4](#) demonstrates the results of the MDP approach for both stochastic and robust models. The information concerning the states and actions are highlighted on the right-side. The probabilities in red and green color give the values of  $X_{ad}$  and  $Tr_a^{dd'}$  (calculated by Equation (14)), respectively.

The restrictions on the product order are the same as in [Appendix A](#):  $u_A = u_B = S$  and  $u'_A = u'_B = S$ ;  $S = 2$ . Such restrictions mean that the system is not allowed to move from picture  $(1 - A, 2 - A)$  to picture  $(1 - A, 2 - A)$  or from picture  $(1 - B, 2 - B)$  to picture  $(1 - B, 2 - B)$ . The main impact comes from the bottleneck product which has the maximum (with significant difference) tasks' processing time. In this example it is product  $B$ . Such restrictions on the whole order of products using  $u'_i$  can be justified for a highly customized or luxury product which requires higher processing times compared to the other product models. In this example, even if we only apply this restriction for product  $B$  and not for product  $A$  ( $u_B = 2$ ), the solution is the same. The total cost of equipment and workers for the worst-case robust model and the expected costs of stochastic model are 2424 and 2122, respectively ( $Cost^{Ro} = 2424$ , and  $Cost^{Sto} = 2122$ ).

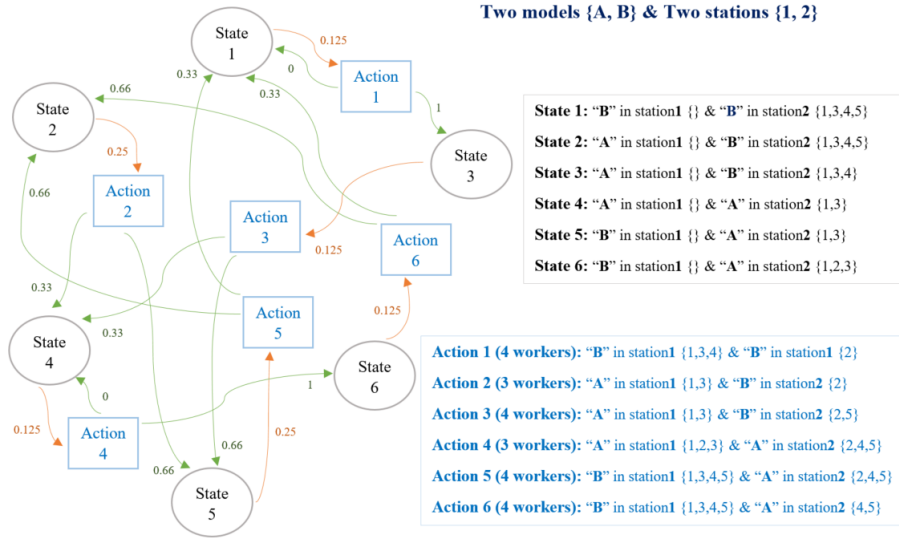


Figure B.4: The MDP graph for the presented illustrative example and the optimal solution of  $MDP^{Sto}$  and  $MDP^{Ro}$ .

## Appendix C. Examples of model-dependent and fixed task assignments.

Compared to the solution of dynamic policy in Appendix A, here, Figure C.5 shows the optimal solution obtained from  $MALBP - W^{Fix}$  and  $MALBP - W^{Md}$  for four possible pictures of the line. The mathematical model, optimization approaches, and an extensive computational results were presented in (Hashemi-Petroodi, et al., 2021). Figure C.5 also shows that the optimal solution of  $MALBP - W^{Fix}$  requires 6 workers and costs 3372, whereas  $MALBP - W^{Md}$  leads to a solution with 5 workers and a total equipment and workforce cost of 2872. As expected, dynamic assignment of  $MALBP - W^{Dyn}$  provides a solution with less total cost of 2424 compared to 2872 for  $MALBP - W^{Md}$  and 3372 for  $MALBP - W^{Fix}$ .

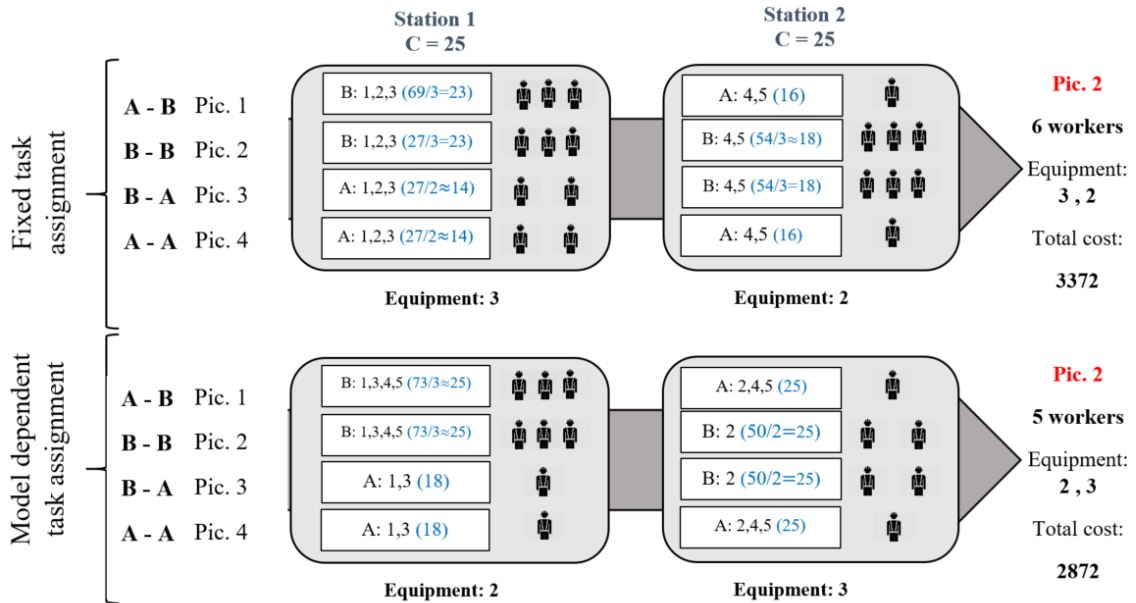


Figure C.5: The optimal solution of  $MALBP - W^{Md}$  and  $MALBP - W^{Fix}$  in the simple example.



## Appendix D. Example of the proposed decomposed transition

Figure D.6 demonstrates the performance of the proposed decomposition algorithm. The line is composed of 2 stations and produces 2 product models. Upon the completion of reduction rules, the number of  $|D0| = 30$  states and  $|A0| = 165$  actions has been reduced to  $|D| = 26$  and  $|A| = 102$ , respectively. They are decomposed into  $PN = I^S = 2 \times 2 = 4$  subsets corresponding to each possible picture of the line with no restriction. To build the transition matrix the problem is divided into  $PN = 4$  sub-problems. Figure D.6 shows all possible line pictures and demonstrates the decomposition process in the context of initial states, actions and resulting states. This example shows that 26 states and 102 actions remain after the reduction process. All 26 states classify into  $PN = 4$  sub-sets corresponding to each picture of the line, and as well as the same for 102 actions. Knowing that for the resulting states, products move to the following stations, the same correlated states can happen with possible pictures of the line as the resulting ones. For example, when product "A" is at the first station in an initial state ( $k = 1, k = 2$ ), for each one of these two pictures product A goes to the second station for the following resulting state ( $k = 2, k = 3$ ). The set  $N'_1$  is equal to the union of two sets  $N_2$  and  $N_3$  ( $N'_1 = N_2 \cup N_3$ ), whereas the set  $N'_2$  is equal to the union of the sets  $N_1$  and  $N_2$  except for the same state which the line is currently on for this picture since the task assignment of products to stations changes ( $N'_2 = (N_2 \cup N_3) - d$ ). It is the same for pictures  $k = 3$  and  $k = 4$ , where product "B" is in the first station ( $N'_3 = N_1 \cup N_4$ ,  $N'_4 = (N_1 \cup N_4) - d$ ). Finally, since we eliminate some states and actions which have never been reached,  $|D1| = 22$  states and  $|A1| = 91$  actions will be involved in the whole final transition matrix as the input of *MILPs*.

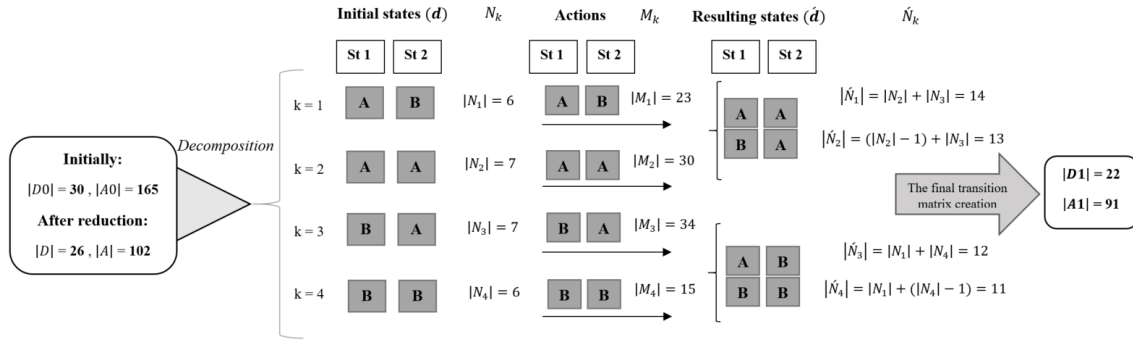


Figure D.6: Illustrative example of the decomposition process.