IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 5 June 2022, accepted 20 July 2022, date of publication 28 July 2022, date of current version 4 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3194658

APPLIED RESEARCH

A Demand-Driven Model for Reallocating Workers in Assembly Lines

RANDALL MAURICIO PEREZ-WHEELOCK[1,2], WEI OU[3], PISAL YENRADEE[4], AND VAN-NAM HUYNH[1], (Member, IEEE)

[1]Graduate School of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Nomi, Ishikawa 932-1292, Japan
[2]Corporate Planning Office, Oyabe Seiki Co. Ltd., Oyabe, Toyama 932-0137, Japan
[3]International Business School, Zhejiang Gongshang University, Hangzhou, Zhejiang 314423, China
[4]School of Manufacturing Systems and Mechanical Engineering, Sirindhorn International Institute of Technology, Thammasat University, Rangsit, Pathum Thani 12120, Thailand

Corresponding author: Randall Mauricio Pérez-Wheelock (randall.m.perez@jaist.ac.jp)

The work of Randall Mauricio Pérez-Wheelock was supported by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) Scholarship.

ABSTRACT This paper introduces the *demand-driven assembly line rebalancing problem* (DDALRP) and proposes a non-linear, multi-objective, combinatorial optimization model to solve it. A DDALRP arises whenever the production output of the assembly line (AL) must be continuously readjusted along a planning horizon in order to satisfy as much as possible a given demand forecast; thus, dealing not with a one-time rebalance, but with a multi-period rebalance, fact that exponentially increases the complexity and combinatorial nature of the problem. Adapting or regulating the production output of the AL to a particular demand forecast or production plan is a relatively new idea in the assembly line balancing (ALB) / rebalancing (ALR) literature; and the novelty of this work is the rebalancing mechanism employed to solve the problem: we address the problem by reallocating workers to stations, taking into consideration their learning and forgetting (L&F) curves. Our proposed model was solved by implementing a genetic algorithm (GA) in 162 cases (three problem instances under 54 scenarios each), which produced useful insights about the dynamics of worker reallocation under different situations: optimistic, most-likely, pessimistic L&F coefficients; experienced and inexperienced workers; and different demand scenarios.

INDEX TERMS Assembly line, demand forecast, learning & forgetting curves, multi-period rebalancing, worker reallocation.

I. INTRODUCTION

Assembly line rebalancing (ALR) is a new research topic within the wider research field of assembly line systems. An assembly line (AL) may need to be rebalanced for different reasons, including: 1) internal drivers such as quality enhancements that lead to changes in product design or product features (thus, resulting in different processing times of tasks and, possibly, a new task precedence diagram); and 2) external stimulus such as changes in the business environment which result in variable market demand (thus, forcing ALs to modify their cycle time and production output rate).

Continuously ensuring that the AL is optimally balanced is a responsibility of critical industrial importance because, as Falkenauer [10] argues, the efficiency difference between an optimal and a sub-optimal balance may yield economies (or loses) reaching millions of dollars per year. Therefore, a rebalancing method must be dynamic in nature in order to readjust the output (number of units) produced by the AL according to some production plan or demand forecast. However, as Li *et al.* [22] have clearly noted, the explicit consideration of the amount of output to be produced by the AL is a new idea in the assembly line balancing (ALB) literature. Furthermore, Pérez-Wheelock and Huynh [27] observed that demand fluctuations have not been taken into consideration in the assembly line balancing problem (ALBP) when assigning tasks or workers to stations.

The associate editor coordinating the review of this manuscript and approving it for publication was Bilal Alatas.

The need for rebalancing ALs in industry was identified long ago. To the best of our knowledge, the first publication that mentioned the idea of rebalancing was that of Chase [8] in 1974. He conducted a survey of ALs in 111 manufacturing plants and concluded that there was a need of balancing algorithms and procedures as evidenced by the large number of rebalances undertaken each year in many categories of industry. Nevertheless, just recently ALR started to gain research attention. There is little literature addressing the ALRP, and related theories and methods are limited. The methods proposed until now address the rebalancing problem by reassigning tasks to stations. This approach, unfortunately, brings chaos to the fundamental structure of the AL. The installation of a production system is a long-term decision which usually requires a large capital investment [5]. Once installed, the stations comprising the assembly line are fixed, well-defined workplaces. A station is responsible for the execution of a particular subset of tasks. To that end, stations are equipped with specific tooling and equipment in order to provide them the required functionality, so that workers are able to perform that particular subset of tasks. Using the terminology employed by Falkenauer [10], each station has its particular 'identity'. Hence, rebalancing an AL in terms of reassigning tasks to stations is inconvenient –or simply impractical– as a result of long-term decisions that were taken when the production system was originally designed and installed.

In this paper, we make an effort to contribute to ALR without reengineering the line or modifying the original assignment of tasks to stations. We propose a way of readjusting the production output of the AL by means of worker reallocation, considering their learning and forgetting curves, and aiming to satisfy as much as possible a given demand forecast or production plan. In this paper, our modest objectives are: 1) to define the *demand-driven assembly line rebalancing problem* (DDALRP); 2) to describe the family of problems related to the DDALRP; 3) to propose a mathematical formulation for one of the problem variants; and 4) to provide insights regarding the dynamics of worker reallocation under different scenarios. This paper is an extension of [27], where the idea of rebalancing ALs via worker reallocation was originally proposed.

To precisely define the scope of this research, we have to consider a production system composed of three core components; namely: machines, people, and information. 1) Machines: we deal with ALs organized in straight layout (I-shaped configuration) and operated by workers (labor-intensive ALs). Semi-automated and robotic ALs, and other types of layout (e.g., U-shaped ALs, two-sided ALs, ALs with parallel stations, etc.) are out of the scope of this research. 2) People: we take into consideration learning and forgetting curves of workers. Other human factors, such as fatigue-recovery, motivation-boredom, etc., are out of the scope of this research. 3) Information: we take into consideration a demand forecast in order to rebalance the line. The actual operation of an assembly line is supported by information systems that provide additional information, such as parts feeding or material supply to the AL. Information other than the demand forecast is beyond the scope of this research.

In spite of its limitations and narrow scope, this paper should nevertheless contribute to both, theory and practice. 1) Academic contribution / implications (theory): This paper intends to contribute to the body of knowledge in the field of assembly line balancing (ALB) / rebalancing (ALR) by introducing a new family of problems (DDALRP). 2) Practical contribution / implications (practice): 2.1) With our method, it is possible to adjust the production output of the AL without the need of reengineering the line. 2.2) Worker reallocation implies job rotation, which leads to job enrichment and multi-skilled, motivated workers.

Following the introduction, research objectives, scope, and academic and practical implications, the rest of this article is organized as follows: Section II provides a review of works on ALR. Section III describes the DDALRP, its typology, and proposes a mathematical formulation for one of the problem variants. Section IV explains the experimental design of the computational study and discusses the results obtained. Finally, Section V concludes the study and points out the directions of our upcoming research agenda.

II. RELATED WORK

Gamberini *et al.* [11] proposed a multi-objective heuristic algorithm for rebalancing the single-model, stochastic-time AL. Their method is founded on the integration of TOPSIS *(Technique for Order Preference by Similarity to Ideal Solution)* and the well-known Kottas and Lau heuristic approach [18]. The authors aim to minimize two objectives: The first objective is the "assembly cost", defined by the authors as the sum of the unit labor cost and the unit incompletion cost. The second objective is to minimize the reassignment of tasks in order to preserve a high degree of similarity between the original and the new balance, thus, avoiding the costs associated with the movement of tasks. Their method was tested on several problem instances.

Gamberini *et al.* [12] developed a multiple single-pass heuristic algorithm in order to find the most complete set of dominant solutions that represent the Pareto front of the stochastic ALRP. Their objectives are to minimize the total expected completion cost and maximize the mean similarity factor between the initial line balance and the new rebalance. Their method was tested on 240 problem instances and one case study.

Belassiria *et al.* [6] proposed an *integrated model for assembly line rebalancing problem* (I-ALRP). After the occurrence of a disturbance, tasks have to be reassigned to stations according to a new cycle time that is derived from a new demand pattern. The objective function under consideration combines line efficiency and workload balance among stations. In their rebalancing model, assignment restrictions are considered, e.g.: machinery is too heavy to move to a different station, as well as positive and negative zoning constraints. To solve the problem, the authors proposed a genetic algorithm (GA) hybridized with a heuristic priority rule. Their mathematical formulation and algorithm were applied to a case study of a company that manufactures wiring harnesses and automotive connectors.

Rebalancing has emerged as a response to some disruption or disturbance that may occur in one or more station, such as a breakdown or shutdown:

Sancı and Azizoğlu [28] considered the need to rebalance an AL due to disruptions in a station. When a disruption occurs, the tasks in the disrupted station have to be reassigned among the active stations. The trade-off between an efficiency measure (cycle time) and a stability measure (the number of reassigned tasks) was taken into consideration to identify the best rebalance of the line. To solve the problem, the authors developed two optimization algorithms: an algorithm based on mixed integer linear programming (MILP), and a branch and bound (B&B) algorithm. Their method was tested on 200 problem instances.

Karas and Ozcelik [17] introduced the *assembly line worker assignment and rebalancing problem* (ALWARBP). In this problem, the authors reassign tasks and workers when a disruption occurs in a station. The tasks in the disrupted station have to be reassigned among the active stations. Also, workers that were in stations that suffered disruption become unoccupied, the total number of workers becomes greater than the number of active stations, and the reassignment of workers becomes necessary because some tasks cannot be performed by some workers. The new line balance intends to achieve the minimum possible variation in terms of cycle time and the number of reassigned tasks. To solve the problem, the authors developed an artificial bee colony (ABC) algorithm. Their method was tested on 120 problem instances.

Li *et al.* [22] investigated the ALRP in a multi-period context, with tasks having stochastic processing times. Upon the occurrence of a disruption, the processing times of tasks change according to a probability distribution that is built with historical data. Disruptions occur according to a Poisson distribution. When a disruption occurs, maintenance or system upgrade must be performed, and the makespan of the work-in-process (WIP) is delayed. In this problem, the objective function is to minimize total cost, comprised of production cost ($C_1 \times$ makespan) and rebalancing cost ($C_2 \times$ number of rebalances). The authors evaluate four rebalancing policies: 1) continuous rebalancing (low production cost but high rebalancing cost); 2) no rebalancing (high production cost but no rebalancing cost); 3) periodic rebalancing (a midpoint between the previous two policies); and 4) data-driven rebalancing (dynamic rebalancing decisions supported by real-time data).

Girit and Azizoğlu [13] define *total replacement distance* as the amount of movement needed to transfer a task from one station to another. When a disruption occurs, the tasks in the disrupted station must be moved to another station. Furthermore, to achieve a satisfactory workload smoothness index, some of the tasks in the non-disrupted stations may have to be reassigned. Hence, the authors perform rebalancing taking into consideration the trade-off between total squared workload (a fairness measure) and total replacement distance (a stability measure). The authors developed two algorithms: an exact algorithm and a tabu search algorithm. Their method was tested on several problem instances.

Rebalancing has been examined in mixed-model ALs:

The research of Altemeier *et al.* [2] addresses the ''reconfiguration'' of an AL characterized by a make-to-order production process and high variety of products. (In their paper, the authors clarify that their reconfiguration problem is merely a rebalancing problem.) Because there is a new production plan in each planning horizon, the workload of stations changes. Therefore, reassigning and shifting tasks among stations becomes necessary as a mean of achieving the most uniform possible distribution of workload among stations. The authors proposed a decision support system consisting of four phases (analysis, design, selection, implementation) with a mathematical formulation and a local search heuristic embedded in it. The authors tested their decision support system in two industrial case studies (two workshops) of a car manufacturer.

Grangeon *et al.* [14] studied the rebalancing problem in a mixed-model AL of a French automotive firm. The line has to be rebalanced monthly due to changes in demand. This rebalance is made by transferring an operation (and its corresponding tools and parts) from one station to another. The rebalancing proposed by the authors involves three steps: 1) obtain a feasible solution; 2) minimize the number of stations; and 3) smooth the operator workload. The authors proposed three heuristics, one for each step.

Oliveira *et al.* [26] presented a case study regarding the rebalancing of the tasks and operations of a mixed-model AL in the automotive industry. The authors proposed a binary integer programming model that aims to minimize the number of stations, solved it using a B&B algorithm, and evaluated the results in terms of the number of stations created, average workload, and imbalance rate.

Yang *et al.* [34] argue that the reassignment of tasks to stations may lead to the movement of machine tools, relocation of WIP buffers, and retraining of workers. The authors proposed a multi-objective genetic algorithm (moGA) for the reassignment of tasks and workers, so as to minimize: 1) the number of stations used (vertical balancing); 2) the workload variation among different models at each station (horizontal balancing); and 3) the rebalancing cost (the cost of retraining workers). Their algorithm was tested on 23 problem instances.

Rebalancing has been investigated in U-shaped ALs:

Agpak [1] recognizes that some machines and fixtures cannot be rearranged, or the rearrangement may be possible at the expense of considerable time and cost. In this research, the author studied the rebalancing of U-shaped lines with deterministic task times. He proposed a two-stage heuristic based on the COMSOAL algorithm to find a common task sequence that minimizes the number of stations while keeping the location of tasks, equipment, and fixtures unchanged for different cycle time values.

Zha and Yu [35] argue that if all machines could be moved freely, the rebalancing problem would simply reduce to the balancing problem; and recognize that, in practice, some machines are stationary but could be relocated at the expense of certain moving cost. In their paper, the authors present a new hybrid algorithm (ACO and filtered beam search) to solve the U-line rebalancing problem. The objectives are: 1) minimize the sum of moving cost of machines + labor cost; and 2) minimize the walking distance of workers. Their method was tested on 26 problem instances.

Çelik *et al.* [7] explain that rebalancing yields a new line balance that may result in more or fewer stations and changes in the position of tasks within the line; and recognize that a rebalance would probably incur in a cost for redesigning the line. In their paper, the authors propose a new approach based on ant colony optimization (ACO) for rebalancing U-lines with stochastic task times (S-ULRBP). Their objective is to minimize the total rebalancing cost which consists of task transportation cost, station opening cost, station closing cost, and station operating cost. Their method was tested on several problem instances.

Serin *et al.* [31] continued the work of Çelik *et al.* [7]. The authors designed a GA to address the S-ULRBP. The algorithm seeks to minimize the total rebalancing cost.

Rebalancing has been studied in two-sided assembly lines:

Zhang *et al.* [36] proposed a station-oriented heuristic algorithm to solve the two-sided ALRP type II[1] (T-ALRP-II). The authors considered two objectives: 1) minimization of the cycle time and 2) minimization of the rebalancing cost, measured as the number of reassigned tasks; and utilized constrained programming (CP) to transform the second objective into a constraint. Their formulation and algorithm was tested in a numerical example.

Zhang *et al.* [37] addressed a real case study, the assembly of a shovel loader in a Chinese company that produces construction machinery. The authors noted unreasonable assignment of tasks in the current balancing of the AL, and therefore, they proposed a modified non-dominated sorting genetic algorithm II (MNSGA-II) to obtain a rebalanced line with minimum cycle time at the lowest possible rebalancing cost.

Zhang *et al.* [38] developed and improved imperialist competitive algorithm (IICA) for rebalancing a two-sided AL, taking into consideration space and resource (SR) constraints. As for the SR constraints, the authors consider tasks that must be assigned to the same station, tasks that cannot be assigned to the same station, tasks that must be assigned to a specified station, and tasks that must be performed synchronously. Three objective functions were considered: 1) minimize the mean absolute deviation of completion time (a workload smooth measure); 2) minimize the cycle time (an efficiency measure); and 3) minimize the rebalancing cost (a cost measure). The authors applied their formulation and algorithm to several benchmark problems and to a case study of a shovel loader manufacturer in China.

Finally, rebalancing has been researched in automated assembly lines (AALs), which are characterized by collaborative learning. In the manual AL (where human workers perform the assembly operations), when a task is reassigned to someone who has never performed that task before, the worker starts developing skill afresh. By contrast, in AALs, an agent receiving a task for the first time also receives the knowledge previously built thanks to supervisory control, distributed control, or related communication.

Li and Boucher [20] introduced the *automated assembly line (re)balancing problem with dynamic task times.* Due to collaborative leaning and the dynamic nature of the processing times of tasks, the current assignment (of tasks to stations) may not be optimal anymore. In this research, the aim of rebalancing is to reduce the length of the line (the number of stations) while keeping the desired cycle time. To solve the problem, the authors proposed backward induction rules. Their method was tested in an industrial case study.

Li [21] extended the work of Li and Boucher [20] by taking into consideration tasks with stochastic processing times and proposing an algorithm, ENCORE (Efficient Non-Constant task time REbalancing), to solve the problem.

A synthesis highlighting selected features of the reviewed literature and this work is organized in Table 13 (appendix) for a quick reference and easier comparison. In column "Demand" we indicate whether or not demand is a direct input in the model (i.e., is demand defined in the authors' notation and is used in their mathematical formulation?). Columns "L" and "F" show whether learning and forgetting effects are considered in the authors' model. If the paper addresses a case study, this is indicated in the last column, along with a brief description of the industrial environment. All other columns are self-explanatory.

III. THE DEMAND-DRIVEN ASSEMBLY LINE REBALANCING PROBLEM

In this section we explain the DDALRP. First, in subsection A, we describe the general overview of the problem while introducing part of the notation to be used. (The rest of the notation is presented progressively in subsections B–D.) Next, in subsection B, we provide a framework for the family of demand-driven assembly line rebalancing problems. Then, in subsection C, we explain the learning and forgetting curves to be used in our model. Finally, in subsection D, we propose a mathematical formulation for one of the DDALRP variants.

A. PROBLEM DESCRIPTION

1) Assembly line: We consider an AL composed of a specific number of stations $j = 1, 2, 3, \ldots, J$, organized in straight layout to produce one homogeneous product (i.e., *single-model* AL).

[1] In the ALB literature, the objective in problem type I is to minimize the number of stations given a specified cycle time, while in problem type II, the objective is to minimize the cycle time given a predefined number of stations. Refer to [29], pp. 20, 32, 42–61.

2) Stations and the flow of materials along the stations: Stations are assumed to have the right tools and instruments that are necessary to perform the subset of tasks that belongs to them. Progressively, on each station, the bill-of-material parts and components are attached to the jobs or workpieces, and these become finished products at the end of the line. Jobs or workpieces advance as soon as stations complete their tasks (i.e., *unpaced* AL). Between two consecutive stations there is a buffer. The buffer behind a station contains the jobs or workpieces to be processed by that station. The buffer in front of a station is used to place the units processed by that station. Each station must guarantee a minimum WIP inventory, WIP_{min}, for its downstream station at the end of each working period in order to ensure smooth production flow at the beginning of the next working period (i.e., avoid the waiting time of feeding the line). Stations are assumed to be reliable; machine breakdowns or downtimes are not considered.
3) Demand: A forecast market demand, $D(\ell)$, is given for each period $\ell = 1, 2, \ldots, L$ of the planning horizon. Demand data may have different patterns, e.g.: increasing, seasonal, erratic, etc. Demand is served at the beginning of each period using the inventory of finished products (IFP) available at the end of the previous period, $I_f(\ell-1)$. When IFP is not enough to cope with the market demand, lost sales occur.
4) Workers and learning and forgetting (L&F) effects: In the factory, there is a fixed number of workers, K, which may (or may not) have previous experience in assembly operations; i.e., workers could have some initial skill inventory, $S_{jk}^{initial}$, defined as the theoretical number of units that worker k would be able to process (in one working period) in his/her first assignment to station j. This capacity improves (according to some learning parameter r) as long as the worker continues performing on the same station, or deteriorates (according to some forgetting parameter f) if he/she discontinues performing on that station. Different workers have different initial skill levels and different learning and forgetting parameters. It is assumed that workers come to the factory every period; worker absenteeism is not considered.
5) Allocation of workers: Since any worker is able to perform the assembly operations of any station, any worker can be assigned to any station. In one period, one worker is to be assigned to one station and one station receives one worker. Therefore, there are as many workers as stations; i.e., $K = J$.
6) The rebalancing mechanisms and its aim: The AL is to be rebalanced by reallocating workers to stations along the planning horizon. In different periods the allocation of workers to stations may vary in order to slow down the production output rate of the AL, or it may be the same in order to take advantage of the learning effect and increase the number of finished goods produced by the AL. By assigning and reassigning workers to stations, the AL shall be balanced in the best possible way in each period $\ell = 0, 1, 2, \ldots, L-1$ in order to satisfy $D(\ell+1)$, the forecast market demand of the next period.

B. DDALRP TYPOLOGY

The solution to a DDALRP seeks to match the production output of the AL to the forecast market demand. The units of finished products available at the end of period ℓ, $I_f(\ell)$, are used to satisfy the market demand of the next period, $D(\ell+1)$. If $I_f(\ell) < D(\ell+1)$, the factory would be losing sales in period $\ell + 1$. The solution to a DDALRP intends to minimize the amount of lost sales. Hence, the objective function:

$$\text{Min } Z_1 = \sum_{\ell=0}^{L-1} \max\left\{0, D(\ell+1) - I_f(\ell)\right\} \tag{1}$$

deals with the first type of DDALRP: minimization of lost sales in terms of units. (We will name this problem DDALRP type A1.)

Moreover, because every unit of unmet demand represents an economic loss, the measure of interest could be lost sales in terms of monetary resources. To reduce the likelihood of losing sales, the AL should build sufficient inventory. However, the fact of holding inventory represents an operating cost which, ideally, should be as small as possible.

If $I_i(\ell)$ and $I_f(\ell)$ are respectively the inventory of finished products at the beginning and at the end of period ℓ, then, $\left[I_i(\ell) + I_f(\ell)\right] \div 2$ is the average inventory held in period ℓ. If h represents the unit cost of holding inventory, and g, the unit cost of lost sale, then, objective function (1) can be reformulated as:

$$\text{Min } Z_1 = g\sum_{\ell=0}^{L-1} \max\left\{0, D(\ell+1) - I_f(\ell)\right\} + h\sum_{\ell=0}^{L} \frac{I_i(\ell) + I_f(\ell)}{2} \tag{2}$$

in order to deal with the second type of DDALRP: minimization of lost sales in terms of monetary resources. (We will refer to this problem as DDALRP type A2.)

Under this analysis, the unmet demand is a lost sale. However, if customers are willing to receive their products later, at the expense of some economic penalty for the factory, then, the unmet demand becomes a backlog or back order. Since there is a penalty for supplying units late, the number of back orders should be minimized. Again, to reduce the likelihood of backlog, enough inventory should be built, and some appropriate balance between the penalty of back orders and the cost of holding inventory shall be sought.

$$\text{Min } Z_1 = b\sum_{\ell=0}^{L-1} \max\left\{0, \sum_{\ell'=0}^{\ell}\left[D(\ell'+1) - Q(J,\ell')\right]\right\} + h\sum_{\ell=0}^{L} \frac{I_i(\ell) + I_f(\ell)}{2} \tag{3}$$

TABLE 1. DDALRP typology.

Objective	Measure of interest	
	1. Units	2. Monetary resources
A. Minimize lost sales	Type A1 Equation (1)	Type A2 Equation (2)
B. Minimize back orders	Type B1 Equation (4)	Type B2 Equation (3)

deals with the third type of DDALRP, minimization of backlog in terms of monetary resources, where b is the unit cost of backlog (the economic penalty for delaying one unit of finished product one period). We will label this problem DDALRP type B2.

Finally, the fourth type of DDALRP deals with the minimization of backlog in terms of units. The corresponding objective function is:

$$\text{Min } Z_1 = \sum_{\ell=0}^{L-1} \max\left\{0, \sum_{\ell'=0}^{\ell}\left[D(\ell'+1) - Q(J, \ell')\right]\right\} \quad (4)$$

and we will call it DDALRP type B1.

The family of demand-driven assembly line rebalancing problems is summarized in Table 1. This paper treats the DDALRP type A1, minimization of lost sales, in units. (Equations (2), (3), and (4) are not used in our computational study; however, these equations were introduced for the purpose of illustrating the DDALRP variants.) We propose a mathematical formulation for problem variant A1 in subsection D. Before that, in subsection C, we need to explain the learning and forgetting curves that will be incorporated in our model.

C. LEARNING AND FORGETTING CURVES

We incorporate the formulations of skill improvement and skill deterioration proposed by Azizi *et al.* [3], which are presented here only for the sake of completeness. Readers are referred to the original work.

On the one hand, when a worker is assigned to a station, his/her skill improves as he/she performs in the same station. A worker's skill improvement may depend on the initial skill (if the worker is assigned by the first time) or the remnant skill (if the worker is re-assigned), the length of the assignment (the elapsed time between the current period, ℓ, and the period on which he/she was assigned, a), and the worker's learning slope. Therefore, skill improvement can be modeled as:

$$S_{jk\ell} = S_j^{max} - \left(S_j^{max} - S_{jka}^{rem}\right) e^{\beta_k(\ell-a)} \quad (5)$$

where $S_{jk\ell}$ is the skill level of worker k in station j in period ℓ. S_j^{max} is the theoretical maximum level of skill at station j. S_{jka}^{rem} is the skill level that worker k had in station j when he/she was assigned to that station (in period a); it is the remnant skill, i.e., the previously gained skill that has not been affected by the forgetting phenomenon. However, at time zero, $\ell = 0$, the skill level of worker k in station j, S_{jk0}^{rem}, corresponds to the worker's initial skill level; hence, $S_{jk0} = S_{jk0}^{rem} = S_{jk}^{initial}$. Therefore, in the case that a worker is assigned to a station for the first time, the remnant skill S_{jka}^{rem} in (5) should be substituted by $S_{jk}^{initial}$. And β_k is the learning slope of worker k given by $\beta_k = (log\ r_k)/(log\ 2)$, where r_k is the learning coefficient of worker k.

According to (5), at infinite time, $\ell \to \infty$, the skill level of worker k performing in station j reaches the maximum level, $S_{jk\infty} = S_j^{max}$. However, achieving the maximum level of skill in infinite time is unrealistic. Therefore, the idea of *achievable upper level* or *skill upper bound* is introduced. This concept is represented by S_j^{UB} and the relationship between S_j^{max} and S_j^{UB} can be expressed as:

$$S_j^{UB} = S_j^{max} - \delta_j \quad (6)$$

where δ_j is the skill upper bound threshold value for station j.

On the other hand, as the worker continues to learn the new skill, his/her previously gained skill decays as a result of the forgetting phenomenon. A worker's skill deterioration may depend on the previous skill level achieved (the worker's skill level at the time of interruption), the length of time that he/she has not been performing (the elapsed time between the current period, ℓ, and the period d at which the worker's experience with that specific station was last interrupted), and the worker's forgetting slope. Therefore, the corresponding skill deterioration formula is:

$$S_{jk\ell}^{rem} = S_j^{min} + \left(S_{jkd} - S_j^{min}\right) e^{\gamma_k(\ell-d)} \quad (7)$$

where $S_{jk\ell}^{rem}$ is the remnant skill of worker k in station j in period ℓ, S_j^{min} is the theoretical minimum level of skill at station j, S_{jkd} is the skill level that worker k had in station j when he/she departed last time (in period d) from that station, and γ_k is the forgetting slope of worker k given by $\gamma_k = (log\ f_k)/(log\ 2)$, where f_k is the forgetting coefficient of worker k.

According to (7), at infinite time, $\ell \to \infty$, the skill level of worker k performing in station j reaches the minimum level, $S_{jk\infty}^{rem} = S_j^{min}$. However, achieving the minimum level of skill in infinite time is unrealistic. Therefore, the idea of *achievable lower level* or *skill lower bound* is introduced. This concept is represented by S_j^{LB} and the relationship between S_j^{min} and S_j^{LB} can be expressed as:

$$S_j^{LB} = S_j^{min} + \epsilon_j \quad (8)$$

where ϵ_j is the skill lower bound threshold value for station j.

In the following subsection we propose a mathematical formulation for the DDALRP type A1, taking into consideration the learning and forgetting curves presented in this subsection.

D. A MATHEMATICAL FORMULATION FOR DDALRP TYPE A1

1) OBJECTIVE FUNCTIONS

The first objective function comprising this formulation is (1). In addition to minimize lost sales, the solution to the

DDALRP also aims to achieve the smoothest possible production flow. Therefore, workers have to be assigned to stations so as to obtain the most balanced AL with the lowest possible standard deviation of the number of units processed by the stations, and the least possible excess of production. These two objective functions were modified and adapted from those of Song *et al.* [32]. If $\overline{Q}(\ell)$ represents the average number of units processed among all stations in period ℓ, then, the degree of how smooth production is flowing along the AL in period ℓ can be expressed by the standard deviation:

$$Std.Dev(\ell) = \sqrt{\frac{1}{J}\sum_{j=1}^{J}\left[Q(j,\ell) - \overline{Q}(\ell)\right]^2}$$

Hence, the second objective function consists of minimizing the standard deviation of the number of units processed by the stations along the whole planning horizon, and can be written as:

$$\text{Min } Z_2 = \sum_{\ell=0}^{L-1}\sqrt{\frac{1}{J}\sum_{j=1}^{J}\left[Q(j,\ell) - \overline{Q}(\ell)\right]^2} \tag{9}$$

The bottleneck station is the station that processes the fewest number of units. The number of units processed by the bottleneck station is $Q(bn) = min\{Q(1), Q(2), \ldots, Q(J)\}$. The excess of production of station j in period ℓ, represented by $Q_e(j,\ell)$, is defined as the difference between the number of units that station j processed (in period ℓ) and the number of units processed by the bottleneck station (in period ℓ). Mathematically, $Q_e(j,\ell) = Q(j,\ell) - Q(bn,\ell)$. The total production excess (of the whole AL) in period ℓ, represented by $Q_e(\ell)$, is the sum of the production excess of all stations:

$$Q_e(\ell) = \sum_{j=1}^{J}\left[Q(j,\ell) - Q(bn,\ell)\right]$$

Therefore, the third objective function is to minimize the total production excess over the whole planning horizon:

$$\text{Min } Z_3 = \sum_{\ell=0}^{L-1}\sum_{j=1}^{J}\left[Q(j,\ell) - Q(bn,\ell)\right] \tag{10}$$

Finally, the solution to the DDALRP intends to conclude the planning horizon with the lowest possible amount of IFP:

$$\text{Min } Z_4 = I_f(L) \tag{11}$$

Notice that Z_1 and Z_4 are measured in units of finished products, while Z_2 and Z_3, in units of unfinished products. Thus, we proceed to normalize the objective functions.

2) NORMALIZATION OF OBJECTIVE FUNCTIONS

In order to handle this multi-objective formulation, we employ the *desirability function* approach, originally introduced by Harrington [15], and later extended by Derringer and Suich [9]. In a nutshell, this approach transforms a response function into a scale-free function that ranges between zero and one. Thus, the value of the desirability function presents the degree of desirability or satisfaction level (SL) for the corresponding response.

The desirability function was extended by Derringer and Suich [9] into three forms: "the larger the better", "the smaller the better" and "nominal the best". In our case, our four objective functions are of the type "the smaller the better". This concept is very intuitive, and works as follows: Let z_1 be the amount of lost sales expressed as a ratio of the total forecast demand:

$$z_1 = \frac{Z_1}{\sum_{\ell=1}^{L} D(\ell)} \tag{12}$$

z_1 should be as small as possible. If z_1 is smaller than or equal to some lower bound, z_1^{ℓ}, SL will be 1. On the contrary, if z_1 is greater than or equal to some upper bound, z_1^{u}, SL will be 0. For any other amount of lost sales between these two extremes, the corresponding SL is computed linearly:

$$SL_1 = \begin{cases} 1 & \text{if } z_1 \leq z_1^{\ell} \\ 1 - \dfrac{z_1 - z_1^{\ell}}{z_1^{u} - z_1^{\ell}} & \text{if } z_1^{\ell} < z_1 < z_1^{u} \\ 0 & \text{if } z_1 \geq z_1^{u} \end{cases} \tag{13}$$

We treat in a similar way the other three objective functions. We define z_2 as the average standard deviation per station expressed as a ratio of the demand forecast, z_3 as the average production excess per station expressed as a ratio of the demand forecast, and z_4 as the amount of IFP at the end of period L as a ratio of the demand forecast:

$$z_2 = \frac{Z_2}{J\sum_{\ell=1}^{L} D(\ell)} \tag{14}$$

$$z_3 = \frac{Z_3}{J\sum_{\ell=1}^{L} D(\ell)} \tag{15}$$

$$z_4 = \frac{Z_4}{\sum_{\ell=1}^{L} D(\ell)} \tag{16}$$

z_2, z_3, and z_4 shall be as small as possible; and the corresponding expressions to compute the SL associated with objective functions 2, 3, and 4 are identical to (13). For the sake of completeness:

$$SL_m = \begin{cases} 1 & \text{if } z_m \leq z_m^{\ell} \\ 1 - \dfrac{z_m - z_2^{\ell}}{z_m^{u} - z_m^{\ell}} & \text{if } z_m^{\ell} < z_m < z_m^{u} \\ 0 & \text{if } z_m \geq z_m^{u} \end{cases}$$

for $m = 2, 3, 4$. Thus, SL is defined on the basis of lower and upper bounds. Table 2 shows the value of these parameters used in our numerical experiments.

Now, with normalized objective functions, we proceed to scalarize; i.e., convert the original problem with multiple objectives into a single-objective optimization problem.

$$\text{Max } Z = SL_1 + SL_2 + SL_3 + SL_4 \tag{17}$$

TABLE 2. Lower and upper bounds used to define *SL*.

	Lower bounds	Upper bounds
SL_1	$z_1^\ell = 0$	$z_1^u = 0.10$
SL_2	$z_2^\ell = 0$	$z_2^u = 0.04$
SL_3	$z_3^\ell = 0$	$z_3^u = 0.08$
SL_4	$z_4^\ell = 0$	$z_4^u = 0.03$

The solution to the DDALRP intends to maximize total SL, as stated in (17), with a special rule on SL_1: If SL_1 will result in 0 as a consequence of a challenging demand, Z_1 (the number of units of lost sales) shall be as low as possible. This is, in scenarios of challenging demand (where it will not be possible to satisfy at least 90% of the market demand forecast and SL_1 will inevitable be 0), the model shall not simply "give up" on SL_1 and "concentrate" on the best combined performance of SL_2, SL_3, and SL_4. The model shall prioritize the lowest possible amount of lost sales (Z_1).

3) CONSTRAINTS

Equations (18) and (19) are the classical "one-to-one" constraints: in each period, every worker is assigned to one station, and each station receives exactly one worker.

$$\sum_{j=1}^{J} x_{jk\ell} = 1 \quad \forall\, k \in K,\ \ell = 0, 1, \ldots, L-1 \tag{18}$$

$$\sum_{k=1}^{K} x_{jk\ell} = 1 \quad \forall\, j \in J,\ \ell = 0, 1, \ldots, L-1 \tag{19}$$

Equation (20) uses the worker allocation decision to compute the theoretical number of units that can be processed in station j in period ℓ. This theoretical number of units is a function of the worker assigned to station j and his/her respective skill level in period ℓ. (21) is the binary constraint imposed to the decision variables. A decision variable is equal to 1 if worker k is assigned to station j in period ℓ; otherwise, it is equal to 0.

$$P(j,\ell) = \sum_{k=1}^{K} (x_{jk\ell} \cdot S_{jk\ell}) \quad \forall\, j \in J,\ \ell = 0, 1, \ldots, L-1 \tag{20}$$

$$x_{jk\ell} \in \{0, 1\} \tag{21}$$

Equations (22)–(24) regulate the dynamics of the actual number of units processed in the stations. Equation (22) indicates that the actual number of units processed in station 1 equals its own theoretical number of units processed. Equations (23) and (24) compute the number of units processed in the other stations in period 0 (when there is no WIP inventory), and in subsequent periods (when there may exist some WIP inventory), respectively.

$$Q(j,\ell) = P(j,\ell) \qquad j = 1,\ \ell = 0, 1, \ldots, L-1 \tag{22}$$

$$Q(j,\ell) = min\,\{P(j,\ell), Q(j-1,\ell)\} \qquad j = 2, 3, \ldots, J,\ \ell = 0 \tag{23}$$

$$Q(j,\ell) = min\,\{P(j,\ell), Q(j-1,\ell) + WIP(j,\ell-1)\} \qquad j = 2, 3, \ldots, J,\ \ell = 1, 2, \ldots, L-1 \tag{24}$$

Equations (25)–(26) together with constraint (27) control the dynamics of WIP inventory. Equations (25) and (26) compute the WIP inventory that remains at each station at the end of period 0, and at the end of subsequent periods, respectively. Constraint (27) requires the WIP inventory at the stations to be at least the minimum necessary to ensure immediate work at the beginning of each period (i.e., avoid the waiting time of feeding a station). Period 0 is excluded from this WIP constraint because in period 0 the AL is empty; there is no WIP inventory. Station 1 is excluded from these three WIP constraints because station 1 is not fed by WIP inventory from a previous station; instead, it is fed by raw materials.

$$WIP(j,\ell) = Q(j-1,\ell) - Q(j,\ell) \qquad j = 2, 3, \ldots, J,\ \ell = 0 \tag{25}$$

$$WIP(j,\ell) = WIP(j,\ell-1) + Q(j-1,\ell) - Q(j,\ell) \qquad j = 2, 3, \ldots, J,\ \ell = 1, 2, \ldots, L-1 \tag{26}$$

$$WIP(j,\ell) \geq WIP_{min} \qquad j = 2, 3, \ldots, J,\ \ell = 1, 2, \ldots, L-1 \tag{27}$$

Finally, equations (28)–(31) compute the IFP at the beginning and at the end of each period ℓ. The equations are self-explanatory.

$$I_i(\ell) = 0 \quad \ell = 0 \tag{28}$$

$$I_f(\ell) = Q(J,\ell) \quad \ell = 0 \tag{29}$$

$$I_i(\ell) = \max\left\{0, I_f(\ell-1) - D(\ell)\right\} \quad \ell = 1, 2, \ldots, L \tag{30}$$

$$I_f(\ell) = I_i(\ell) + Q(J,\ell) \quad \ell = 1, 2, \ldots, L \tag{31}$$

IV. NUMERICAL EXPERIMENTS

We now present and discuss numerical experiments demonstrating the use of the developed formulation. The AL data for these experiments come from the collection of *simple assembly line balancing problems* (SALBP) that appears in Scholl [30]. All cases in these experiments were solved using a GA implemented in Python NumPy, and were launched in the same computer: Intel(R) Xeon(R) with 32 CPUs E5-2667 v3 at 3.20GHz, 125 GB of memory, and SUSE Linux Enterprise Server 12 SP1 operating system. In the following subsection A, we describe the methodology and experimental design. Afterwards, in subsection B, we present and discuss the results obtained.

A. EXPERIMENTAL DESIGN

The proposed model was tested on 162 cases (three problem instances, each of them was run under 54 variations). The selected problems were:

1) Martens (7 tasks, organized in 2 stations),
2) Jackson (11 tasks, organized in 3 stations), and

TABLE 3. Pre-processing of the problem instances, step 1: Theoretical number of stations and their layout.

Problem instance:	Martens	Jackson	Mitchell
Number of tasks	7	11	21
Sum of the processing time of all tasks	29 min	46 min	105 min
Available productive time per period	480 min	480 min	480 min
Production required per period	30 units	30 units	17 units
Cycle time	16 min/unit	16 min/unit	28 min/unit
Theoretical number of stations	2	3	4
Layout	straight	straight	straight

3) Mitchell (21 tasks, organized in 4 stations).[2]

These problems were prepared for experimentation according to the following methodology.

1) PRE-PROCESSING OF THE PROBLEM INSTANCES

The pre-processing of a problem instance consists of designing a specific AL (number of stations, layout, distribution of tasks among stations), and calculating all of the parameters related to its production capacity intended in its initial design and configuration (S_j^{max}, δ_j, S_j^{min}, ϵ_j).

The first pre-processing step deals with defining the number of stations and their layout (Table 3). Consider, for instance, the Martens problem. The Martens problem consists of 7 tasks, which together require a processing time of 29 minutes. For the purpose of assuming an AL that is already built, assume that: (a) there are 8 hours (480 min.) of available productive time per period, and (b) the AL is intended to produce 30 units per period. Hence, the cycle time is $CT = 480$ min $\div$ 30 units $=$ 16 min/unit. The minimum theoretical number of stations is computed by dividing the processing time of all tasks (29 min.) by the cycle time, and rounding up to the nearest integer, $\lceil 29 \div 16 \rceil = 2$ stations.

The second pre-processing step deals with the distribution of tasks to stations and the calculation of the capacity-related parameters. The distribution of tasks to stations is shown in the first two columns of Table 4. For instance, consider the Martens problem again. Station 1 consists of tasks 1, 2, 4, and 5, resulting in a station load $SL_1 = 14$ min. Therefore, in one working period (480 min.), a worker with the theoretical maximum level of skill performing in station 1 should be able to process $S_1^{max} = 480$ min $\div$ 16 min/unit $= 34.29$ units. According to equation (6), achieving the theoretical maximum level of skill is unrealistic; therefore, a skill upper bound threshold $\delta_1 = 2$ units was defined for station 1. S_j^{min} values were all set equal to 0. According to equation (8), achieving the theoretical minimum level of skill is unrealistic; therefore, a skill lower bound threshold $\epsilon_1 = 3$ units was defined for station 1.

[2]Note that we do not consider ALs longer than 4 stations. Recall that in the first few cycles, the AL is being filled with products. The longer the AL, the more cycles it takes to feed the entire AL. This requires a different modeling from the one proposed in this paper. The equations that we have proposed [(22)–(26)] are deterministic, and therefore, can approximate the dynamics of ALs composed of a small number of stations. We leave the investigation of longer ALs –characterized by throughput variability and uncertainty– as part of our next research agenda.

TABLE 4. Pre-processing of the problem instances, step 2: Assignment of tasks to stations and definition of parameters associated with capacity.

Station	Tasks	SL_j (min)	S_j^{max} (units)	δ_j (units)	S_j^{min} (units)	ϵ_j (units)
Pre-processing of Martens problem						
$j = 1$	1, 2, 4, 5	14	34.29	2	0	3
$j = 2$	3, 6, 7	15	32.00	1	0	3
Pre-processing of Jackson problem						
$j = 1$	1, 2, 4, 5	16	30.00	1	0	3
$j = 2$	3, 6, 7, 8	16	30.00	1	0	3
$j = 3$	9, 10, 11	14	34.29	2	0	3
Pre-processing of Mitchell problem						
$j = 1$	1, 2, 3, 4, 21	28	17.14	1	0	2
$j = 2$	5, 6, 7, 8	28	17.14	1	0	2
$j = 3$	9–16	26	18.46	1	0	2
$j = 4$	17, 18, 19, 20	23	20.87	2	0	2

2) SCENARIOS AND CASES

Each of the 3 problem instances was run under different scenarios, creating a total of 54 cases for each problem instance. The scenarios were created for: 1) learning and forgetting (L&F) coefficients (optimistic, most-likely, and pessimistic scenarios); 2) initial skill inventory (workers with some initial skill inventory and workers with no initial skill inventory); 3) demand challenge or level of difficulty of the demand forecast (easy to achieve, intermediate, and difficult to achieve); and 4) demand pattern (increasing, seasonally increasing, and erratic).

First, optimistic, most-likely, and pessimistic values were defined for L&F coefficients. Learning coefficients ranged between 0.700 and 0.850. (The lower the value, the faster the learning effect.) Optimistic values: [0.700, 0.750), most-likely values: [0.750, 0.800), pessimistic values: [0.800, 0.850). Forgetting coefficients ranged between 0.950 and 0.800. (The higher the value, the slower the forgetting effect.) Optimistic values: [0.950, 0.900), most-likely values: [0.900, 0.850), pessimistic values: [0.850, 0.800). Based on the value of L&F coefficients, three cases have been developed.

It is possible to distinguish between the "best" worker, "average" workers, and the "worst" worker. If workers are sorted in order from the best to the worst, then, the best worker has the lowest r value (fastest learning) and the highest f value (slowest forgetting). Progressively, r values increase and f values decrease, until reaching the worst worker, who has the highest r value (slowest learning) and the lowest f value (fastest forgetting).

Second, in regard to the initial skill inventory, two scenarios were created: (a) workers are experienced, and hence, possess some initial skill inventory ($S^{initial}$ values were obtained randomly following a uniform distribution in the interval that ranges between 40% and 60% of the lowest S_j^{max} value); and (b) workers are new operators or new hires, and hence, do not have any initial skill inventory ($S^{initial} = 0$ for all workers). This variation, combined with the optimistic, most likely, and pessimistic scenarios described above, generates six cases.

Third, the demand challenge: each of the previous six cases was run under three different levels of difficulty: easy, intermediate, and difficult. In the easy and intermediate scenarios, workers, at their skill upper bound, are able to produce during the length of the planning horizon (52 periods) a total number of units that exceeds by approximately 30% and 10%, respectively, the total demand forecast that shall be satisfied. Of course, workers do not start performing the assembly operations at their skill upper bound from the beginning. Reaching the skill upper bound takes time, especially if workers are new operators or new hires (i.e., workers with no initial skill inventory). Therefore, the "margins" of 30% and 10% are actually less. In the difficult scenario, workers, at their skill upper bound, are able to satisfy approximately 90% of the total market demand.

Consider the Martens problem again. The lowest S_j^{max} value is that of station 2, $S_2^{max} = 32$ units. Hence, according to (6) the skill upper bound is $S_2^{UB} = S_2^{max} - \delta_2 = 31$ units. Since the planning horizon in these experiments consist of 52 periods, 31 units $\times$ 52 = 1612 units. Accordingly, the easy (1612 $\div$ 1.3), intermediate (1612 $\div$ 1.1), and difficult (1612 $\div$ 0.9) demand scenarios consist of 1240, 1465, and 1791 units, respectively. These three variations in the demand challenge, combined with the previous scenarios, generate 18 cases.

Finally, the three levels of difficulty described above were run under three scenarios of demand forecast pattern: increasing, seasonally increasing, and erratic. This is, the demand challenge computed above, was distributed differently across the planning horizon, shaping increasing, seasonally increasing, and erratic patterns. This variation, combined with the previous scenarios, generates a total of 54 cases for each problem instance. The deployment of cases run is organized in Table 5.

3) GENETIC ALGORITHM

According to Battaïa and Dolgui [4], GAs seem to be the most popular solution method employed to solve ALBPs. A GA is a powerful metaheuristic technique inspired by the process of natural selection. This technique was initially developed by Holland [16] in 1975. Today, GAs are employed in different research domains because they can provide nearly optimal solutions in reasonable time [19]. Nevertheless, the fine tuning of our GA parameters was the most time-consuming activity in this research. We systematically tested values and, after 300 trial runs, we identified the parameters the led to the best solutions. To run our numerical experiments, we coded a GA with the following characteristics and parameters:

- Initialization: An initial population of size 1% (of the total number of possible allocations) is randomly generated. (The total number of possible allocations is $K!^L$, where K is the number of workers and L is the number of periods in the planning horizon.)

TABLE 5. Deployment of cases run.

Group 1 (increasing demand pattern)

Case	Demand challenge	Skill inventory	r and f parameters
1	Easy	Workers have some initial skill inventory	optimistic values
2			most-likely values
3			pessimistic values
4		$S^{initial} = 0$ for all workers	optimistic values
5			most-likely values
6			pessimistic values
7	Intermediate	Workers have some initial skill inventory	optimistic values
8			most-likely values
9			pessimistic values
10		$S^{initial} = 0$ for all workers	optimistic values
11			most-likely values
12			pessimistic values
13	Difficult	Workers have some initial skill inventory	optimistic values
14			most-likely values
15			pessimistic values
16		$S^{initial} = 0$ for all workers	optimistic values
17			most-likely values
18			pessimistic values

Group 2 (seasonally increasing demand pattern) conforms the next 18 cases, following the same structure, above, and numbered as cases 19–36.
Group 3 (erratic demand pattern) conforms the next 18 cases, 37–54.

- Selection: A portion (0.1%) of the existing population is selected to breed a new generation. Individual solutions are selected on the basis of the fitness function defined in equation (17) (total satisfaction level), which serves as a measure of the solution quality.
- Genetic Operators: Successive generations were obtained by recombination; more specifically, *crossover two points* with a rate of 0.75. In addition, mutation was implemented randomly and with very low probability of occurrence, 0.01.
- Termination: Our GA stops when either, it has found the best possible satisfaction level it is capable of finding or when the maximum allowed time of 5 hours is reached.

The solution representation of our problem is straightforward and is illustrated in Table 6. For each station listed on the left, a gene string is deployed on the right, indicating the worker allocation from period 0 to 51.

Our GA was run 10 times for each of the 162 cases, and the best solution found (i.e., that with the highest SL) is reported. We present and discuss the results in the following subsection B.

B. RESULTS AND DISCUSSION

1) RESULTS

Tables 7–9 present the results of all 162 cases that were run in the three problem instances. For each case listed, results of the objective functions Z_1 (lost sales, LS), Z_2 (standard deviation, SD), Z_3 (production excess, PE), and Z_4 (inventory of finished products, IFP) are presented, followed by their corresponding contribution to the satisfaction level. Total satisfaction level is presented in column SL, followed by the total number of reallocations that each worker k experienced throughout the length of the planning horizon. Worker reallocation was

TABLE 6. Solution representation.

	Period, ℓ									
Station	0	1	2	3	4	5	6	7	...	51
$j = 1$	2	2	1	1	4	4	1	1	...	2
$j = 2$	1	1	2	2	3	3	2	2	...	1
$j = 3$	4	4	3	3	2	2	3	3	...	4
$j = 4$	3	3	4	4	1	1	4	4	...	3

measured by the number of times that a worker changed station. The last column in Tables 7–9 presents the computation time, measured in seconds.

Furthermore, Tables 10–12 present additional details of the worker reallocation observed in the three problem instances and all of its cases under scenarios of ''easy'' and ''intermediate'' demand. These tables do not contain results for the ''difficult'' cases because under this scenario there was no worker reallocation. (Workers remained in the same station throughout the whole planning horizon.) A dot indicates the periods where worker reallocation took place. The first allocation of workers occurs in period 0; therefore, any reallocation may start as early as period 1. The last column on the right shows the total number of times that a worker changed station; i.e., the count of dots in a given row.

2) VALIDATION OF RESULTS

The purpose of this subsection is to show that the results obtained are reasonable. To that end, we employed simulation and *what-if* analysis in order to observe how total SL would be affected by modifications in the worker allocation decision. For example, Jackson case 3 draws attention because of its 9.6 units of lost sales (Table 8). A simulation of this case shows that reducing worker reallocation from 23 to 22, increases 21.3 units the production output of the AL; thus, helping eliminate lost sales, but also leaving 11.8 units of IFP. At the end, total SL results lower with 11.8 units of IFP ($SL = 3.662$) than with 9.6 units of lost sales ($SL = 3.917$). Moreover, because 9.6 units of lost sales is less than 10% of the total demand forecast, SL_1 brings some contribution (to the total satisfaction level) that is greater than zero, and therefore, there is no need to apply the special priority rule on SL_1 –prioritize the lowest possible amount of lost sales. (This rule applies in cases where SL_1 would result in 0 as a consequence of a challenging demand scenario.)

To provide another example, consider Martens case 46 in Table 10. What if worker reallocation were shifted one period earlier (i.e., if dots moved one position to the left)? We simulated this situation and observed that total SL would be reduced from 3.032 to 3.031. Furthermore, we simulated two additional situations: 1) add one more reallocation in period 19, and 2) remove the reallocation of period 36. Both situations resulted in a lower total SL.

Thus, results were validated via simulation and *what-if* analysis in order to ensure that the results are reasonable. Upon validation of the results obtained, the basic behavior of worker reallocation is summarized as follows:

3) THE DYNAMICS OF WORKER REALLOCATION

1) *In the periods when reallocation was observed, all workers moved to a different station.*

For instance, in the Jackson problem, with three stations and three workers, there is no period where only two workers switch stations while the third worker remains in the same station. When reallocation occurs, all workers change station. This means that being ''best'', ''average'', or ''worst'' worker did not affect the decision as to whether or not to reallocate. The explanation for this phenomenon has to do with reasons related to smooth production flow. If the ''best'' worker (''worst'' worker) had remained in the same station, production excess (bottleneck) would have occurred. Therefore, the movement of all workers was required so as to achieve a balanced production flow.

Regarding the ''easy'' and ''intermediate'' demand scenarios:

2) From Tables 10–12, it is clear that *the ''easy'' demand scenario allows more room for the reallocation of workers than the ''intermediate'' demand scenario.*

In the ''easy'' demand scenario, the natural ability of workers in the assembly operations leads to a higher number of units produced than that required by the demand forecast. Therefore, in order to match the demand forecast, worker reallocation is used to slow down the production output of the AL.

3) *Within each subgroup of six cases, a higher number of reallocations occurs when workers have some initial skill inventory, compared to the scenario of workers with no skill inventory.*

For instance, cases 25–27 exhibit more reallocations than their respective counterparts, cases 28–30. The fact of having some initial skill requires more reallocation to adjust the production output of the AL. By contrast, the scenario of workers with no initial skill requires less reallocation in order to make progress in the learning curve.

4) *More worker reallocation is needed as we move from the pessimistic to the most likely, to the optimistic scenario.*

For instance, more worker reallocation is observed as we move from case 3 to 2 to 1; or from case 6 to 5 to 4. In the experiments, in general, within each subgroup of six cases, the number of reallocations is greatest in the ''optimistic, with some initial skill inventory'' scenario; and least in the ''pessimistic, no initial skill inventory'' scenario. Only one exception was observed: Jackson cases 48, 47, 46 did not follow this pattern. Jackson case 47 exhibits more reallocations (11) than case 46 (10).

5) *The demand pattern, whether increasing or seasonally increasing, has little effect on the worker allocation solution.* For instance, compare cases 1–6 vs. 19–24, or cases 7–12 vs. 25–30 in Tables 10–12.

Martens case 12 (in the intermediate, increasing scenario) and case 30 (its counterpart in the intermediate, seasonally increasing scenario) was the only pair of cases that showed a different allocation solution.

TABLE 7. Results of the 54 cases of the Martens problem.

		Case	LS (units)	SD (units)	PE (units)	IFP (units)	SL_1	SL_2	SL_3	SL_4	SL	Reallocations $k=1$	Reallocations $k=2$	Run time (seconds)
Increasing demand pattern	Easy	1	0.0	64.4	128.8	108.6	1.000	0.351	0.351	0.000	1.701	51	51	695
		2	0.0	55.8	111.6	1.5	1.000	0.438	0.438	0.961	2.836	36	36	1101
		3	0.0	42.8	85.5	12.7	1.000	0.569	0.569	0.659	2.797	21	21	1065
		4	0.0	54.9	109.9	62.8	1.000	0.446	0.446	0.000	1.892	50	50	962
		5	0.0	39.3	78.6	4.4	1.000	0.604	0.604	0.881	3.088	28	28	1133
		6	0.0	28.1	56.1	6.0	1.000	0.717	0.717	0.838	**3.272**	18	18	822
	Intermediate	7	0.0	46.0	92.0	7.6	1.000	0.607	0.607	0.827	3.040	20	20	669
		8	3.4	41.9	83.8	0.0	0.977	0.642	0.642	1.000	3.260	12	12	782
		9	6.8	38.5	76.9	0.0	0.954	0.671	0.671	1.000	3.296	8	8	781
		10	3.0	33.1	66.2	3.3	0.979	0.717	0.717	0.924	3.337	10	10	769
		11	4.1	29.1	58.2	0.0	0.972	0.751	0.751	1.000	**3.474**	6	6	771
		12	10.8	26.7	53.4	0.0	0.926	0.772	0.772	1.000	3.469	4	4	694
	Difficult	13	213.4	34.5	69.0	0.0	0.000	0.759	0.759	1.000	2.518	0	0	713
		14	219.8	34.2	68.5	0.0	0.000	0.761	0.761	1.000	2.522	0	0	601
		15	229.7	33.6	67.2	0.0	0.000	0.765	0.765	1.000	2.531	0	0	752
		16	243.4	30.9	61.9	0.0	0.000	0.784	0.784	1.000	2.568	0	0	692
		17	256.2	29.9	59.8	0.0	0.000	0.791	0.791	1.000	2.582	0	0	732
		18	276.8	28.0	56.0	0.0	0.000	0.804	0.804	1.000	**2.609**	0	0	602
Seasonally increasing demand pattern	Easy	19	0.0	64.4	128.8	108.6	1.000	0.351	0.351	0.000	1.701	51	51	676
		20	0.0	55.8	111.6	1.5	1.000	0.438	0.438	0.961	2.836	36	36	829
		21	0.0	42.8	85.5	12.7	1.000	0.569	0.569	0.659	2.797	21	21	1154
		22	0.0	54.9	109.9	62.8	1.000	0.446	0.446	0.000	1.892	50	50	815
		23	0.0	39.3	78.6	4.4	1.000	0.604	0.604	0.881	3.088	28	28	977
		24	0.0	28.1	56.1	6.0	1.000	0.717	0.717	0.838	**3.272**	18	18	836
	Intermediate	25	0.0	46.0	92.0	7.6	1.000	0.607	0.607	0.827	3.040	20	20	666
		26	3.4	41.9	83.8	0.0	0.977	0.642	0.642	1.000	3.260	12	12	779
		27	6.8	38.5	76.9	0.0	0.954	0.671	0.671	1.000	3.296	8	8	946
		28	1.0	33.1	66.2	1.3	0.993	0.717	0.717	0.969	3.396	10	10	787
		29	4.1	29.1	58.2	0.0	0.972	0.751	0.751	1.000	3.474	6	6	772
		30	0.0	27.6	55.2	1.6	1.000	0.764	0.764	0.963	**3.491**	1	1	717
	Difficult	31	213.4	34.5	69.0	0.0	0.000	0.759	0.759	1.000	2.518	0	0	674
		32	219.8	34.2	68.5	0.0	0.000	0.761	0.761	1.000	2.522	0	0	734
		33	229.7	33.6	67.2	0.0	0.000	0.765	0.765	1.000	2.531	0	0	742
		34	243.4	30.9	61.9	0.0	0.000	0.784	0.784	1.000	2.568	0	0	705
		35	256.2	29.9	59.8	0.0	0.000	0.791	0.791	1.000	2.582	0	0	769
		36	276.8	28.0	56.0	0.0	0.000	0.804	0.804	1.000	**2.609**	0	0	756
Erratic demand pattern	Easy	37	11.6	62.5	124.9	135.4	0.907	0.370	0.370	0.000	1.647	47	47	697
		38	20.1	66.1	132.2	0.0	0.838	0.334	0.334	1.000	2.505	42	42	833
		39	19.6	56.8	113.6	13.3	0.842	0.427	0.427	0.641	2.338	26	26	1334
		40	38.4	50.7	101.5	114.8	0.690	0.489	0.489	0.000	1.668	46	46	872
		41	45.5	48.9	97.7	6.3	0.633	0.507	0.507	0.831	2.479	36	36	846
		42	55.2	32.4	64.9	9.2	0.554	0.673	0.673	0.751	**2.652**	22	22	913
	Intermediate	43	26.4	52.8	105.7	0.0	0.820	0.548	0.548	1.000	2.916	29	29	678
		44	43.1	42.0	83.9	0.0	0.705	0.641	0.641	1.000	2.988	16	16	937
		45	33.7	42.4	84.8	10.0	0.770	0.637	0.637	0.773	2.817	10	10	693
		46	53.3	35.3	70.6	0.0	0.635	0.698	0.698	1.000	**3.032**	20	20	679
		47	61.6	32.1	64.3	0.0	0.579	0.725	0.725	1.000	3.029	12	12	858
		48	72.2	28.7	57.3	7.0	0.506	0.755	0.755	0.840	2.855	8	8	682
	Difficult	49	213.4	34.5	69.0	0.0	0.000	0.759	0.759	1.000	2.518	0	0	708
		50	219.8	34.2	68.5	0.0	0.000	0.761	0.761	1.000	2.522	0	0	746
		51	229.7	33.6	67.2	0.0	0.000	0.765	0.765	1.000	2.531	0	0	755
		52	243.4	30.9	61.9	0.0	0.000	0.784	0.784	1.000	2.568	0	0	680
		53	256.2	29.9	59.8	0.0	0.000	0.791	0.791	1.000	2.582	0	0	727
		54	276.8	28.0	56.0	0.0	0.000	0.804	0.804	1.000	**2.609**	0	0	753

[16]LS = lost sales (units), SD = standard deviation (units), PE = production excess (units), IFP = inventory of finished products (units), $k = 1, 2, 3, \ldots, K$ identifies the worker. SL_i = the satisfaction level (dimensionless) associated with objective function Z_i, SL = total satisfaction level (dimensionless). **Highest** and lowest SL are highlighted within each subgroup of six cases.

VOLUME 4, 2016

TABLE 8. Results of the 54 cases of the Jackson problem.

Demand pattern	Difficulty	Case	LS (units)	SD (units)	PE (units)	IFP (units)	SL_1	SL_2	SL_3	SL_4	SL	Number of reallocations $k=1$	$k=2$	$k=3$	Run time (seconds)
Increasing demand pattern	Easy	1	0.0	0.0	0.0	76.4	1.000	1.000	1.000	0.000	3.000	51	51	51	959
		2	0.0	0.0	0.0	1.4	1.000	1.000	1.000	0.960	3.960	35	35	35	1369
		3	9.6	0.0	0.0	0.0	0.917	1.000	1.000	1.000	3.917	23	23	23	2226
		4	0.0	0.0	0.0	33.9	1.000	1.000	1.000	0.026	3.026	51	51	51	1682
		5	0.0	0.0	0.0	7.6	1.000	1.000	1.000	0.781	3.781	29	29	29	1496
		6	0.9	0.0	0.0	0.0	0.992	1.000	1.000	1.000	**3.992**	19	19	19	899
	Intermediate	7	0.0	0.0	0.0	2.5	1.000	1.000	1.000	0.940	3.940	17	17	17	1513
		8	6.4	0.0	0.0	0.0	0.953	1.000	1.000	1.000	3.953	11	11	11	1471
		9	3.5	0.0	0.0	0.0	0.975	1.000	1.000	1.000	**3.975**	7	7	7	1926
		10	4.3	0.0	0.0	0.0	0.969	1.000	1.000	1.000	3.969	9	9	9	1663
		11	4.2	0.0	0.0	0.0	0.969	1.000	1.000	1.000	3.969	5	5	5	1492
		12	8.5	0.0	0.0	0.0	0.938	1.000	1.000	1.000	3.938	3	3	3	1347
	Difficult	13	206.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1746
		14	214.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	911
		15	226.8	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	887
		16	230.0	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1867
		17	242.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1659
		18	262.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1264
Seasonally increasing demand pattern	Easy	19	0.0	0.0	0.0	76.4	1.000	1.000	1.000	0.000	3.000	51	51	51	1119
		20	0.0	0.0	0.0	1.4	1.000	1.000	1.000	0.960	3.960	35	35	35	1248
		21	9.6	0.0	0.0	0.0	0.917	1.000	1.000	1.000	3.917	23	23	23	2412
		22	0.0	0.0	0.0	33.9	1.000	1.000	1.000	0.026	3.026	51	51	51	1599
		23	0.0	0.0	0.0	7.6	1.000	1.000	1.000	0.781	3.781	29	29	29	1633
		24	0.9	0.0	0.0	0.0	0.992	1.000	1.000	1.000	**3.992**	19	19	19	1472
	Intermediate	25	0.0	0.0	0.0	2.5	1.000	1.000	1.000	0.940	3.940	17	17	17	1500
		26	6.4	0.0	0.0	0.0	0.953	1.000	1.000	1.000	3.953	11	11	11	1640
		27	3.5	0.0	0.0	0.0	0.975	1.000	1.000	1.000	**3.975**	7	7	7	884
		28	4.3	0.0	0.0	0.0	0.969	1.000	1.000	1.000	3.969	9	9	9	1682
		29	4.2	0.0	0.0	0.0	0.969	1.000	1.000	1.000	3.969	5	5	5	1713
		30	8.5	0.0	0.0	0.0	0.938	1.000	1.000	1.000	3.938	3	3	3	1058
	Difficult	31	206.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1795
		32	214.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1014
		33	226.8	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	896
		34	230.0	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1891
		35	242.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1012
		36	262.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1114
Erratic demand pattern	Easy	37	16.9	0.0	0.0	107.6	0.854	1.000	1.000	0.000	2.854	47	47	47	1292
		38	23.6	0.0	0.0	0.0	0.797	1.000	1.000	1.000	**3.797**	37	37	37	1166
		39	29.5	0.0	0.0	0.0	0.746	1.000	1.000	1.000	3.746	23	23	23	2203
		40	37.4	0.0	0.0	91.2	0.678	1.000	1.000	0.000	2.678	45	45	45	1417
		41	49.4	0.0	0.0	0.0	0.574	1.000	1.000	1.000	3.574	36	36	36	1277
		42	51.6	0.0	0.0	0.6	0.555	1.000	1.000	0.983	3.538	23	23	23	930
	Intermediate	43	23.2	2.9	9.1	0.0	0.831	0.983	0.972	1.000	**3.785**	18	18	18	1625
		44	26.7	4.3	13.8	1.5	0.805	0.974	0.958	0.964	3.701	11	11	11	1659
		45	37.4	7.3	23.1	0.0	0.727	0.956	0.930	1.000	3.613	9	9	9	947
		46	49.4	1.0	3.0	4.5	0.640	0.994	0.991	0.890	3.515	10	10	10	1479
		47	56.8	0.0	0.0	0.6	0.585	1.000	1.000	0.985	3.571	11	11	11	1659
		48	66.6	0.0	0.0	1.0	0.514	1.000	1.000	0.975	3.489	7	7	7	910
	Difficult	49	206.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	2122
		50	214.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	916
		51	227.0	0.1	0.2	0.0	0.000	0.999	0.999	1.000	2.999	0	0	0	1009
		52	230.0	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1131
		53	242.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1044
		54	262.4	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	1005

LS = lost sales (units), SD = standard deviation (units), PE = production excess (units), IFP = inventory of finished products (units), $k = 1, 2, 3, \ldots, K$ identifies the worker. SL_i = the satisfaction level (dimensionless) associated with objective function Z_i, SL = total satisfaction level (dimensionless). **Highest** and lowest SL are highlighted within each subgroup of six cases.

TABLE 9. Results of the 54 cases of the Mitchell problem.

Demand pattern	Difficulty	Case	LS (units)	SD (units)	PE (units)	IFP (units)	SL_1	SL_2	SL_3	SL_4	SL	Number of reallocations $k=1$	$k=2$	$k=3$	$k=4$	Run time (seconds)
Increasing demand pattern	Easy	1	0.0	3.6	16.7	64.7	1.000	0.965	0.919	0.000	2.885	51	51	51	51	18,000
		2	0.1	11.3	52.2	0.0	0.988	0.891	0.747	1.000	3.636	43	43	43	43	18,000
		3	0.0	8.7	42.5	5.8	1.000	0.916	0.795	0.703	3.413	25	25	25	25	18,000
		4	0.0	0.0	0.0	31.0	1.000	1.000	1.000	0.000	3.000	51	51	51	51	18,000
		5	0.2	6.4	29.3	0.0	0.977	0.939	0.858	1.000	3.793	35	35	35	35	18,000
		6	0.4	3.9	17.9	0.0	0.993	0.962	0.913	1.000	**3.869**	21	21	21	21	18,000
	Intermediate	7	1.0	3.6	16.7	0.0	0.987	0.970	0.932	1.000	3.890	25	25	25	25	18,000
		8	2.0	5.1	23.0	0.0	0.974	0.959	0.906	1.000	3.838	13	13	13	13	18,000
		9	0.0	4.2	20.3	2.8	1.000	0.965	0.917	0.877	3.759	7	7	7	7	18,000
		10	1.0	0.0	0.0	0.8	0.987	1.000	1.000	0.965	**3.952**	9	9	9	9	18,000
		11	1.0	0.0	0.0	1.0	0.987	1.000	1.000	0.957	3.944	1	1	1	1	18,000
		12	1.0	0.0	0.0	8.5	0.987	1.000	1.000	0.627	3.614	2	2	2	2	18,000
	Difficult	13	109.7	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		14	112.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		15	117.2	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		16	127.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		17	134.1	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		18	144.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
Seasonally increasing demand pattern	Easy	19	0.0	3.6	16.7	64.7	1.000	0.965	0.919	0.000	2.885	51	51	51	51	18,000
		20	0.1	11.3	52.2	0.0	0.988	0.891	0.747	1.000	3.636	43	43	43	43	18,000
		21	0.0	8.7	42.5	5.8	1.000	0.916	0.795	0.703	3.413	25	25	25	25	18,000
		22	0.0	0.0	0.0	31.0	1.000	1.000	1.000	0.000	3.000	51	51	51	51	18,000
		23	0.2	6.4	29.3	0.0	0.977	0.939	0.858	1.000	3.793	35	35	35	35	18,000
		24	0.4	3.9	17.9	0.0	0.993	0.962	0.913	1.000	**3.869**	21	21	21	21	18,000
	Intermediate	25	1.0	3.6	16.7	0.0	0.987	0.970	0.932	1.000	3.890	25	25	25	25	18,000
		26	2.0	5.1	23.0	0.0	0.974	0.959	0.906	1.000	3.838	13	13	13	13	18,000
		27	0.0	4.2	20.3	2.8	1.000	0.965	0.917	0.877	3.759	7	7	7	7	18,000
		28	1.0	0.0	0.0	0.8	0.987	1.000	1.000	0.965	**3.952**	9	9	9	9	18,000
		29	1.0	0.0	0.0	1.0	0.987	1.000	1.000	0.957	3.944	1	1	1	1	18,000
		30	1.0	0.0	0.0	8.5	0.987	1.000	1.000	0.627	3.614	2	2	2	2	18,000
	Difficult	31	109.7	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		32	112.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		33	117.2	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		34	127.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		35	134.1	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		36	144.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
Erratic demand pattern	Easy	37	0.2	2.4	11.1	76.0	0.997	0.977	0.946	0.000	2.920	46	46	46	46	18,000
		38	1.9	2.4	10.9	3.9	0.970	0.977	0.947	0.801	**3.696**	40	40	40	40	18,000
		39	3.9	2.4	10.8	4.4	0.940	0.977	0.948	0.775	3.640	26	26	26	26	18,000
		40	18.9	0.0	0.0	57.3	0.707	1.000	1.000	0.000	2.707	46	46	46	46	18,000
		41	21.0	8.1	36.1	3.0	0.674	0.921	0.825	0.844	3.266	40	40	40	40	18,000
		42	25.0	5.6	23.3	7.6	0.614	0.945	0.887	0.605	3.052	24	24	24	24	18,000
	Intermediate	43	10.2	2.4	11.1	0.1	0.866	0.980	0.955	0.998	**3.799**	30	30	30	30	18,000
		44	11.9	2.4	10.9	3.2	0.844	0.981	0.955	0.860	3.640	14	14	14	14	18,000
		45	13.9	2.4	10.8	0.7	0.818	0.965	0.956	0.970	3.725	11	11	11	11	18,000
		46	28.9	0.0	0.0	2.5	0.621	1.000	1.000	0.891	3.512	20	20	20	20	18,000
		47	34.9	2.7	6.3	0.0	0.543	0.978	0.974	1.000	3.494	14	14	14	14	18,000
		48	44.6	2.5	5.7	0.0	0.415	0.980	0.977	1.000	3.372	10	10	10	10	18,000
	Difficult	49	109.7	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		50	112.6	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		51	117.2	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		52	127.5	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		53	134.1	0.0	0.0	0.0	0.000	1.000	1.000	1.000	3.000	0	0	0	0	18,000
		54	142.0	1.0	2.3	0.0	0.000	0.993	0.992	1.000	2.985	0	0	0	0	18,000

LS = lost sales (units), SD = standard deviation (units), PE = production excess (units), IFP = inventory of finished products (units), $k = 1, 2, 3, \ldots, K$ identifies the worker. SL_i = the satisfaction level (dimensionless) associated with objective function Z_i, SL = total satisfaction level (dimensionless). **Highest** and lowest SL are highlighted within each subgroup of six cases.

Both, the increasing and seasonally increasing demand patterns share the characteristic of *gradual* change, which possibly leads to the same (or nearly the same) worker allocation solution; while in the erratic demand pattern, the change is *abrupt*; thus, leading to a different allocation of workers.

6) *When the demand pattern is erratic, worker reallocation is never observed from the beginning of the planning horizon.*

In the cases with erratic demand (37–48), due to a sudden high demand in early periods, no worker reallocation is observed at the beginning of the planning horizon. Workers remain in the same station, and, if enough IFP has been assured, worker reallocation takes place in order to slow down the production output of the AL and conclude the planning horizon with the lowest possible amount of IFP.

By contrast, in the increasing and seasonally increasing demand patterns, we cannot categorically state whether such reallocations occur earlier or later in the planning horizon. For instance, in the Jackson problem (Table 11), cases 1–7 of the increasing demand pattern, worker reallocation started early in the planning horizon; while in cases 8–12, late. Moreover, in cases 19–25 of the seasonally increasing demand pattern, worker reallocation started early; while in cases 26–30, late.

In the "difficult" demand scenario:

7) *No worker reallocation occurs in the cases belonging to the "difficult" scenario.*

Finally, under the "difficult" cases observed in the three problem instances, the assignment of workers in period 0 remained the same throughout the whole planning horizon, indicating the need to capitalize on learning effects in order to cope with the challenging demand and satisfy the forecast to the extent possible.

At this point, we have already addressed the research objectives intended in this paper. However, because we believe that Tables 7–12 are dense, and for reasons of exhaustiveness, we would like to proceed with additional discussion on the results obtained. The following subsection is for readers interested in further details.

4) DISCUSSION

Among cases 1–6, case 1 presents the most favorable conditions for production (optimistic L&F, workers have some initial skill inventory); however, it resulted in the lowest *SL*. The low *SL* is mainly driven by $SL_4 = 0$ as a result of high IFP at the end of the planning horizon. In order to slow down the production output of the AL, worker reallocation is needed, and case 1 already presents reallocation in all periods. (See column "Number of reallocations" in Tables 7–9 showing 51 reallocations in case 1 of the three problem instances.) Therefore, optimistic L&F and initial skill inventory do not seem to be an advantage in scenarios of "easy" demand; thus, suggesting that additional *production and operations management* (P&OM) initiatives (e.g., employ that workforce and AL in the production of other families of products) must be considered.

Case 6, by contrast, resulted in the highest *SL* within this first subgroup of six cases. In case 6, production conditions are the least favorable (L&F scenario is pessimistic and workers have no initial skill inventory). Nevertheless, this situation did not have a negative impact on lost sales and, additionally, it was favorable for the purpose of concluding the planning horizon with low IFP.

Next, the following question deserves discussion: How was it possible that Jackson cases 1–6 achieved a perfect satisfaction level of 1 in both, SD and PE? Consider, for instance, case 1 of Jackson problem. In period 0,

- Station 1 received worker 2 ($S_{12}^{initial} = 12$ units);
- Station 2 received worker 1 ($S_{21}^{initial} = 13$ units); and
- Station 3 received worker 3 ($S_{33}^{initial} = 14$ units).

With this distribution, the three stations process 12 units; thus, resulting in 0 standard deviation and 0 production excess, and leading to SL_2 and SL_3 be equal to 1. This situation was observed in several other cases of the Jackson problem. (See Jackson cases 7–42 and 47–54 in Table 8.)

Martens' SL_2 and SL_3 also merit discussion because in each of its 54 cases, the two performance metrics resulted in exactly the same score. With only two stations, the computation of SD (in any given case) results in exactly half the value of PD (for that same case). By coincidence, the lower and upper bounds used to define

- SL_2 ($z_2^{\ell} = 0$, $z_2^{u} = 0.04$) and
- SL_3 ($z_3^{\ell} = 0$, $z_3^{u} = 0.08$) (refer to Table 2)

transform SD and PD figures in satisfaction levels that have the same numerical value. This suggests that, in short ALs (i.e., ALs composed of a few number of stations), objective functions Z_2 and Z_3 lead to SL_2 and SL_3 figures that are similar; thus, suggesting the possibility that the employment of either Z_2 or Z_3 could have led to nearly the same results shown here with the employment of both, Z_2 and Z_3.

(For an assembly line composed of *n* stations, what values of z_2^{ℓ}, z_2^{u}, z_3^{ℓ}, and z_3^{u} lead to the same SL_2 and SL_3 scores? Are Z_2 and Z_3 equivalent (redundant to each other) for the purpose of achieving the same degree of smoothness in the production flow? These and other questions are proposed as part of a mathematical analysis of the DDALRP. See further research directions in Section V.)

Cases 7–12: With the exception of Jackson cases 8 and 12, and Mitchell 12, cases 7–12 of the three problem instances show higher *SL* than their respective counterparts in cases 1–6. In cases 7–12, the demand challenge is "intermediate", and this level of difficulty is more in accordance with the production capability of workers than in the previous six cases. SL_1 resulted very close to 1, satisfying nearly the entire demand forecast. Lost sales –if any– resulted in less than 1% of the demand forecast. Martens case 7, Jackson case 7, and Mitchell case 9 resulted in flawless LS. Moreover, due to that better consonance between the demand forecast and the capability of workers, SL_4 was a driver of better performance (compared to cases 1–6), as evidenced by lower IFP achieved by the end of the planning horizon. Additionally, in Martens

TABLE 10. Worker reallocation, Martens problem.

			Period, ℓ																																																		
		Case	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
Increasing demand	Easy	1		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		2		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*					*	*								*	*		*	*				36
		3		*	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*					*	*					*	*																					*	21
		4		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		50
		5		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*									*	*		*	*											*	*		*	*			28
		6																		*	*					*	*				*	*											*	*	*	*	*	*	*	*	*	*	*	*	18
	Intermediate	7		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																20
		8		*	*	*	*	*	*	*	*	*	*	*	*																																						12
		9		*	*	*	*	*	*	*	*																																										8
		10		*	*	*	*	*	*	*	*	*	*																																								10
		11																																														*	*	*	*	*	*	6
		12																																																*	*	*	*	4
Seasonally increasing demand	Easy	19		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		20		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*					*	*								*	*		*	*				36
		21		*	*	*	*	*	*	*	*	*	*	*	*	*	*			*	*					*	*					*	*																					*	21
		22		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		50
		23		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*		*	*									*	*		*	*											*	*		*	*			28
		24																		*	*					*	*				*	*											*	*	*	*	*	*	*	*	*	*	*	*	18
	Intermediate	25		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																20
		26		*	*	*	*	*	*	*	*	*	*	*	*																																						12
		27		*	*	*	*	*	*	*	*																																										8
		28		*	*	*	*	*	*	*	*	*	*																																								10
		29																																														*	*	*	*	*	*	6
		30																																																		*		1
Erratic demand	Easy	37						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	47
		38											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	42
		39																											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	26
		40							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	46
		41												*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*						36
		42																						*	*	*	*						*	*					*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	22
	Intermediate	43																								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	29
		44																											*	*											*	*	*	*	*	*	*	*	*	*	*	*	*	*	16
		45																																										*	*	*	*	*	*	*	*	*	*	10
		46															*	*	*	*																		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	20
		47																																						*	*	*	*	*	*	*	*	*	*	*	*		12
		48																																												*	*	*	*	*	*	*	*	8

TABLE 11. Worker reallocation, Jackson problem.

			Period, ℓ																																																	
		Case	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
Increasing demand	Easy	1		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		2		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	35
		3		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																													23
		4		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		5		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																							29
		6		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																	19
	Intermediate	7		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																			17
		8																																									*	*	*	*	*	*	*	*	*	*	*	11
		9																																													*	*	*	*	*	*	*	7
		10																																											*	*	*	*	*	*	*	*	*	9
		11																																															*	*	*	*	*	5
		12																																																	*	*	*	3
Seasonally increasing demand	Easy	19		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		20		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	35
		21		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																													23
		22		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		23		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																							29
		24		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																	19
	Intermediate	25		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																																			17
		26																																									*	*	*	*	*	*	*	*	*	*	*	11
		27																																													*	*	*	*	*	*	*	7
		28																																											*	*	*	*	*	*	*	*	*	9
		29																																															*	*	*	*	*	5
		30																																																	*	*	*	3
Erratic demand	Easy	37						*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	47
		38								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*									37
		39														*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	23
		40								*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	45
		41																	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	36
		42																														*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	23
	Intermediate	43																																			*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	18
		44																																									*	*	*	*	*	*	*	*	*	*	*	11
		45																																											*	*	*	*	*	*	*	*	*	9
		46																																										*	*	*	*	*	*	*	*	*	*	10
		47																																									*	*	*	*	*	*	*	*	*	*	*	11
		48																																												*	*	*	*		*	*	*	7

TABLE 12. Worker reallocation, Mitchell problem.

			Period, ℓ																																																	
		Case	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	
Increasing demand	Easy	1		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		2		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*									43
		3		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																											25
		4		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		5		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	35
		6		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																															21
	Intermediate	7		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																											25
		8		*	*	*	*	*	*	*	*	*	*	*	*	*																																							13
		9		*	*	*	*	*	*	*																																												7
		10																																											*	*	*	*	*	*	*	*	*	9
		11																																																*				1
		12																																																		*	*	2
Seasonally increasing demand	Easy	19		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		20		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*									43
		21		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																											25
		22		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	51
		23		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																	35
		24		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																															21
	Intermediate	25		*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*																											25
		26		*	*	*	*	*	*	*	*	*	*	*	*	*																																							13
		27		*	*	*	*	*	*	*																																												7
		28																																											*	*	*	*	*	*	*	*	*	9
		29																																																*				1
		30																																																		*	*	2
Erratic demand	Easy	37							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	46
		38													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	40
		39																											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	26
		40							*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	46
		41													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	40
		42																													*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	24
	Intermediate	43																	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*							30
		44																													*	*	*	*	*	*	*	*	*	*	*	*	*	*											14
		45																																									*	*	*	*	*	*	*	*	*	*	*	11
		46																											*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*	*							20
		47																																						*	*	*	*	*	*	*	*	*	*	*	*	*	*	14
		48																																										*	*	*	*	*	*	*	*	*	*	10

and Mitchell, SL_2 and SL_3 were drivers of better performance. In Jackson cases 7–12, SL_2 and SL_3 continued with the perfect score observed in cases 1–6.

In cases 7–12, worker reallocation was significantly less (than that observed in cases 1–6). The number of reallocations were reduced at least 50% (as in Mitchell case 7). The highest

TABLE 13. Summary of the reviewed literature and this work.

Publication	Rebalancing characteristics	Assembly line characteristics	Demand	L	F	Objectives	Solution	Case study
Gamberini *et al.* [11]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: straight Task time: stochastic	No	No	No	1) Minimize the assembly cost 2) Minimize the reassignment of tasks	Integration of TOPSIS and the well-known Kottas and Lau heuristic	
Gamberini *et al.* [12]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: straight Task time: stochastic	No	No	No	1) Minimize the total expected completion cost 2) Maximize the mean similarity factor between the initial line balance and the new rebalance	1) Multiple single-pass heuristic algorithm 2) Multi-objective genetic algorithm	Precedence diagram, mean performance time of tasks, and other data of a case study is provided. There is no description of the industrial environment.
Belassiria *et al.* [6]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Several lines. Each line can assemble one product model; total number of lines = total number of product models. Layout: straight Task time: deterministic; may vary depending on the line	No. However, in the case study, new customer orders are captured by the information system. "Demand arrivals" are used to compute the new required cycle time.	No	No	One combined objective (line efficiency and workload balance among stations)	Genetic algorithm hybridized with a heuristic priority rule	A company that manufactures wiring harnesses and automotive connectors.
Sancı and Azizoğlu [28]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: straight Task time: deterministic	No	No	No	1) Minimize the cycle time 2) Minimize the number of reassigned tasks	1) An algorithm based on mixed integer linear programming 2) A branch and bound algorithm	
Karas and Ozcelik [17]	☑ Reassign tasks ☑ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: straight Task time: deterministic; differs depending on the worker	No	No	No	One combined objective: achieve minimum possible variation in terms of cycle time and the number of reassigned tasks	Artificial bee colony algorithm	
Li *et al.* [22]	☑ Reassign tasks ☐ Reallocate workers ☐ One-time rebalancing ☑ Multi-period rebal.	Single-model AL Layout: straight Task time: stochastic	Yes	No	No	Minimize total cost: production cost + rebalancing cost	Branch, bound, and remember algorithm	
Girit and Azizoğlu [13]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: straight Task time: deterministic (case 1) and variable (case 2, two discrete uniform distributions were used to generate low and high task time scenarios)	No	No	No	1) Total squared workload (a fairness measure) 2) Total replacement distance (a stability measure)	1) Exact algorithm 2) Tabu search algorithm	
Altemeier *et al.* [2]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Mixed-model AL Layout: straight Task time: deterministic	No. A new production program is taken into consideration to find the production frequency of tasks, necessary input for rebalancing the AL.	No	No	Minimize the costs incurred by overload situations, regular workers, and reconfiguration (rebalancing) measures.	A decision support system consisting of four phases (analysis, design, selection, implementation) with a mathematical formulation and a local search heuristic embedded in it.	Two industrial case studies (two workshops) of a car manufacturer characterized by a make-to-order production process and high product variety.
Grangeon *et al.* [14]	☑ Reassign tasks ☐ Reallocate workers ☐ One-time rebalancing ☑ Multi-period rebal.	Mixed-model AL Layout: straight Task time: deterministic and constant (the duration of a task is the same irrespective of the type of vehicle)	Yes. Each month, the master sequencing provides a description and the quantities of vehicles to produce for the next month.	No	No	1) Minimize the number of stations 2) Minimize the gap of the workload among stations	Three heuristics: 1) Obtain a feasible solution 2) Minimize the number of stations 3) Smooth the operator workload	A French automotive firm.
Oliveira *et al.* [26]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Mixed-model AL Layout: straight Task time: deterministic	No	No	No	1) Minimize the number of stations 2) Average workload 3) Imbalance rate	Branch and bound algorithm	A company in the automotive industry.
Yang *et al.* [34]	☑ Reassign tasks ☑ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Mixed-model AL Layout: straight Task time: deterministic; the processing time of task i depends on the model j	No. However, the authors use w_j, the ratio of the number of product units of model j to the overall demand.	No	No	Minimize: 1) the number of stations used (vertical balancing) 2) the workload variation among different models at each station (horizontal balancing) 3) the rebalancing cost (the cost of retraining workers)	Multi-objective genetic algorithm	
Agpak [1]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: U-shaped Task time: deterministic	No	No	No	Minimize the number of stations while keeping the location of tasks, equipment, and fixtures unchanged for different cycle time values	A two-stage heuristic based on the COMSOAL algorithm	
Zha and Yu [35]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model AL Layout: U-shaped Task time: deterministic	No	No	No	1) Minimize the sum of moving cost of machines + labor cost 2) Minimize the walking distance of workers	A new hybrid algorithm (ant colony optimization and filtered beam search)	The precedence diagram of a mechanical product is provided.

TABLE 13. ***(Continued.)*** **Summary of the reviewed literature and this work.**

Publication	Rebalancing characteristics	Assembly line characteristics	Demand	L	F	Objectives	Solution	Case study
Çelik *et al.* [7]	☑ Reassign tasks ☐ Reallocate workers Different rebalances for different cycle times	Single-model AL Layout: U-shaped Task time: stochastic	No. However, scenarios of increased and decreased demand were assumed in order to generate different cycle times as requirement to rebalance the line.	No	No	Minimize total rebalancing cost (task transportation cost + station opening cost + station closing cost + station operating cost)	Ant colony optimization	
Serin *et al.* [31]	☑ Reassign tasks ☐ Reallocate workers Different rebalances for different cycle times	Single-model AL Layout: U-shaped Task time: stochastic	No. However, changes in demand were assumed in order to generate different cycle times.	No	No	Minimize total rebalancing cost (task transportation cost + station opening cost + station closing cost + station operating cost)	Genetic algorithm	
Zhang *et al.* [36]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model, two-sided AL Layout: straight Task time: deterministic	No	No	No	1) Minimize the cycle time 2) Minimize the rebalancing cost (measured as the number of reassigned tasks)	A workstation-oriented heuristic algorithm	
Zhang *et al.* [37]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model, two-sided AL Layout: straight Task time: deterministic	No	No	No	1) Minimize the cycle time 2) Minimize the rebalancing cost	A modified non-dominated sorting genetic algorithm II	The assembly of a shovel loader in a Chinese company that produces construction machinery.
Zhang *et al.* [38]	☑ Reassign tasks ☐ Reallocate workers ☑ One-time rebalancing ☐ Multi-period rebal.	Single-model, two-sided AL Layout: straight Task time: deterministic	No	No	No	1) Minimize the mean absolute deviation of completion time (a workload smooth measure) 2) Minimize the cycle time (an efficiency measure) 3) Minimize the rebalancing cost (a cost measure)	An improved imperialist competitive algorithm	The assembly of a shovel loader in a Chinese company that produces construction machinery.
Li and Boucher [20]	☑ Reassign tasks ☐ Reallocate workers *Continuous* rebalancing: tasks are reassigned in each repetition	Automated, single-model AL Layout: straight Task time: dynamic	No	Learning automata, collaborative learning	No	Minimize the number of stations	Backward induction rules	Assembly robots can serve two assembly line conveyors (one at a time, not simultaneously).
Li [21]	☑ Reassign tasks ☐ Reallocate workers *Periodic* rebalancing: tasks are reassigned in each production cycle	Automated, single-model AL Layout: straight Task time: stochastic	No	Learning automata, collaborative learning	No	Minimize the cycle time	ENCORE (Efficient Non-Constant task time REbalancing) algorithm	
This paper	☐ Reassign tasks ☑ Reallocate workers ☐ One-time rebalancing ☑ Multi-period rebal.	Single-model AL Layout: straight Task time: not a direct input in this research. We compute the skill level of workers based on L&F effects. By definition, task times are said to be *dynamic* in models that consider L&F.	Yes	Yes	Yes	One combined objective for: 1) Minimize lost sales 2) Minimize the standard deviation of the units processed by the stations 3) Minimize production excess 4) Minimize the IFP that remains at the end of the planning horizon	Genetic algorithm	

reduction observed was in Mitchell case 11, where workers changed station only one time (vs. 35 times in Mitchell case 5); followed by Mitchell case 12 (only two reallocations vs. 21 in Mitchell case 6).

In cases 13–18 there was no reallocation. Workers remained in the same station to make progress in their learning curves, at the particular station where they were assigned. In spite of this, it was not possible to meet at least 90% of the demand forecast (the threshold from which SL_1 starts bringing a positive contribution to the total satisfaction level.) Accordingly, SL_1 resulted in 0 in cases 13–18. Lost sales range between 11.9% and 15.7% (of the demand forecast), being more severe in case 18 (least favorable production conditions; e.g., pessimistic L&F, no initial skill inventory), and less severe in case 13 (most favorable production conditions). As workers remained in the same station throughout the length of the planning horizon, there was an effort to procure the lowest possible amount of lost sale (Z_1), as it was meant to be according to the special priority rule on SL_1. Nevertheless, additional P&OM initiatives must be considered (e.g., overtime, temporal workers in night shift, etc.) in order to further reduce Z_1.

In the set of cases of increasing demand pattern (1–18), those belonging to the scenario of "intermediate" demand (7–12), along with cases 2, 3, 5, and 6, could better maneuver worker reallocation showing better balance between LS and IFP and thus, better combined performance $SL_1 + SL_4$. By contrast, in the other cases (1, 4, and 13–18), a trade-off is usually observed between objective functions 1 and 4. When there is significant amount of LS (implying $SL_1 = 0$), the planning horizon naturally concluded with 0 units of IFP (implying $SL_4 = 1$), as observed in cases 13–18. On the contrary, cases 1 and 4 showed flawless LS (perfect score in SL_1) with a high amount of IFP at the end of the planning horizon (score 0 (or nearly 0) in SL_4).

The six "easy" cases of the seasonally increasing demand pattern (cases 19–24) exhibit the same results observed in the six "easy" cases of the increasing demand pattern (cases 1–6); hence, apparently suggesting that the demand pattern has no effect on the worker allocation decision. However, the six "easy" cases of the erratic demand pattern (cases 37–42) reveal a different solution, suggesting that the demand pattern does impact the allocation decision. Recall that the total demand that must be satisfied in the "easy, erratic" scenario is the same as that of the "easy, increasing" and that of the "easy, seasonally increasing" scenarios. However, due to the nature of the erratic pattern, it is possible to have a sudden high demand early in the planning horizon. Therefore, as an effort to have IFP available to satisfy these unexpected demand peaks, there is no worker reallocation in the first few periods of the cases belonging to the erratic scenario. (Refer to Tables 10–12.)

With the exception of Martens cases 28 and 30 (Table 7), the six "intermediate" cases of the seasonally increasing demand pattern (cases 25–30) exhibit the same results observed in the six "intermediate" cases of the increasing demand pattern (cases 7–12). However, the results of the six "intermediate" cases of the erratic demand pattern exhibit a different solution. The reason relies on the fact that the erratic demand pattern presents abrupt changes, as opposed to the gradual changes of the increasing and seasonally increasing demand patterns.

In the cases belonging to the erratic demand scenario, due to a sudden demand peak, workers remain in the same station at the beginning of the planning horizon in order to take advantage of the learning effect. If enough inventory has been assured, then, worker reallocation takes place in order to reduce the production output rate of the AL. The number of worker reallocations is reduced as we move from the optimistic, to the most-likely, to the pessimistic scenario. In some cases, the fact of having or lacking some initial skill inventory had a small impact on the number of reallocations. For instance, compare cases 38 and 39 (some initial skill inventory) vs. cases 41 and 42 (no initial skill inventory); as well as cases 44 and 45 vs. cases 47 and 48 in Jackson and Mitchell problems (Tables 11 and 12).

Due to abrupt peaks of demand in the erratic scenario, lost sales in cases 37–48 were higher than in their counterparts, cases 1–12 (increasing scenario) and cases 19–30 (seasonally increasing scenario). In the "easy" and "intermediate" scenarios of both, increasing and seasonally increasing demand, lost sales resulted in less than 1% of their respective demand forecast. By contrast, in the "easy, erratic" scenario, lost sales reached up to 4.5%, 4.3%, and 3.9% in Martens, Jackson, and Mitchell, respectively. In the "intermediate, erratic" scenario, lost sales reached up to 4.9% in Martens and Jackson, and up to 5.8% in Mitchell.

Lost sales in the six subgroups of the "difficult, erratic" scenario (cases 49–54) resulted in the same amount as in their counterparts, "difficult, increasing" (13–18) and "difficult, seasonally increasing" (31–36). In these cases, there was no worker reallocation. Therefore, the learning curves of workers progressed in the same way throughout the planning horizon; thus, producing the same number of finished goods and satisfying the market demand at the same level. As a consequence, the amount of lost sales observed in cases 13–18, 31–36, and 49–54, is the same.

A detailed discussion of the results obtained has been provided. Now, we proceed to conclude this study.

V. CONCLUSION AND FUTURE RESEARCH DIRECTIONS

In this paper we introduced the *demand-driven assembly line rebalancing problem.* The underlying mechanism for rebalancing the AL is the reallocation of workers, by which the AL will produce more (or fewer) finished goods based on the learning (or forgetting) curve of workers. In general, the reallocation of workers aims to satisfy a given demand forecast or production plan. More specifically, four objectives were established: minimize lost sales in terms of units (A1) or in terms of economic loss (A2), minimize back orders in terms of units (B1) or in terms of economic loss (B2); thus, giving rise to a family of four problem variants. A non-linear, multi-objective, combinatorial optimization model was proposed for DDALRP type A1. The model was tested in 162 cases (three problem instances under 54 variations each) using a GA, and results were organized to obtain useful insights about the dynamics of worker reallocation under different scenarios (different L&F coefficients and skill inventory of workers, as well as different demand scenarios).

Now that we have grasped the basic behavior of worker reallocation under different scenarios, a further research step in problem variant A1 may involve analytical approaches for sensitivity analysis. For instance, to what extent may input parameters (L&F coefficients, lower and upper bounds of $z_1, \ldots, z_4$, etc.) be increased or reduced without causing a modification in the worker allocation solution? How much less reallocation is likely to be observed if the number of units required by the demand forecast increased 7% (or, if the erraticity of the demand data increased 7% across the planning horizon)? In case of lost sales, how much improvement in the learning (or forgetting) coefficients is required in order to avoid such lost sales? Furthermore, given that learning tends to last and, when forgetting occurs, remembering takes shorter time than learning for the first time; then, as time passes, worker reallocation will be having a weaker impact on slowing down the production output rate of the AL and, in the long run, worker reallocation will have no significant impact on the number of units produced (i.e., any worker allocation would lead to nearly the same production output). At this point, which we may call the *steady state*, it can be said that all workers have achieved the skill upper bound in all assembly operations and have become fully multi-skilled. How long does it take to achieve the steady state and what are its practical implications for the purpose of line rebalancing? An analytical approach to answer these questions is desirable in order to have a deeper understanding of both, capabilities and limitations of this model. For this reason, the mathematical analysis of the DDALRP is being planned in our next research agenda.

Additionally, our future research directions include:

1) *Investigate the DDALRP in long I-shaped assembly lines.* ALs consisting of a larger number of stations require a more complex modeling than the one proposed in this paper. In the current formulation, we are using deterministic equations to model the number of units processed by the stations as well as the flow of WIP inventory along the AL. When a production system is modeled purely deterministic (i.e., randomness is not taken into consideration), the queue time of the jobs or workpieces is zero and it is said that the manufacturing system is being operated in the "ideal" situation [33]. However, in the presence of randomness, queue time is observed, and the actual number of units produced by the AL is certainly less than that computed by the deterministic model. As Wu *et al.* argue, variability is a fundamental property of production lines [33].

2) *Develop heuristics especially dedicated to solve the DDALRP,* capable of finding better solutions within shorter computation time. One important feature that we want to challenge is *meta-GA;* i.e., a GA that optimizes the value of parameters and operates inside the main GA (the GA that finds the best allocation of workers). In this way, we would be automating the fine-tuning process and avoiding the time-consuming activity of systematically testing values. Recently, in the ALB literature, innovative optimization algorithms are being adapted and/or improved to solve specific assembly line balancing problems; for example: *lexicographical whale optimization algorithm* for the type-II ALBP considering preventive maintenance [24]; *water-flow like algorithm* for solving U-shaped ALBPs [25]; and *multi-objective genetic flatworm algorithm* for solving stochastic, mixed-model, two-sided disassembly lines [23]; to mention a few.

3) *Undertake a comparative study of the four problem variants* (A1, A2, B1, B2).

ACKNOWLEDGMENT

Randall Mauricio Pérez-Wheelock is grateful to Yusuke Ide, an Associate Professor at the Department of Mathematics, Nihon University, and William Holden, a Professor at JAIST, for their valuable comments.

REFERENCES

[1] K. Agpak, "An approach to find task sequence for re-balancing of assembly lines," *Assem. Autom.*, vol. 30, no. 4, pp. 378–387, Sep. 2010.

[2] S. Altemeier, M. Helmdach, A. Koberstein, and W. Dangelmaier, "Reconfiguration of assembly lines under the influence of high product variety in the automotive industry—A decision support system," *Int. J. Prod. Res.*, vol. 48, no. 21, pp. 6235–6256, Nov. 2010.

[3] N. Azizi, S. Zolfaghari, and M. Liang, "Modeling job rotation in manufacturing systems: The study of employee's boredom and skill variations," *Int. J. Prod. Econ.*, vol. 123, no. 1, pp. 69–85, Jan. 2010.

[4] O. Battaïa and A. Dolgui, "A taxonomy of line balancing problems and their solution approaches," *Int. J. Prod. Econ.*, vol. 142, no. 2, pp. 259–277, 2013.

[5] C. Becker and A. Scholl, "A survey on problems and methods in generalized assembly line balancing," *Eur. J. Oper. Res.*, vol. 168, no. 3, pp. 649–715, 2006.

[6] I. Belassiria, M. Mazouzi, S. ELfezazi, A. Cherrafi, and Z. ELMaskaoui, "An integrated model for assembly line re-balancing problem," *Int. J. Prod. Res.*, vol. 56, no. 16, pp. 5324–5344, Aug. 2018.

[7] E. Celik, Y. Kara, and Y. Atasagun, "A new approach for rebalancing of U-lines with stochastic task times using ant colony optimisation algorithm," *Int. J. Prod. Res.*, vol. 52, no. 24, pp. 7262–7275, Dec. 2014.

[8] R. B. Chase, "Survey of paced assembly lines," *Ind. Eng.*, vol. 6, no. 2, pp. 14–18, 1974.

[9] G. Derringer and R. Suich, "Simultaneous optimization of several response variables," *J. Qual. Technol.*, vol. 12, no. 4, pp. 214–219, 1980.

[10] E. Falkenauer, "Line balancing in the real world," in *Proc. Int. Conf. Product Lifecycle Manage.*, vol. 5, 2005, pp. 360–370.

[11] R. Gamberini, A. Grassi, and B. Rimini, "A new multi-objective heuristic algorithm for solving the stochastic assembly line re-balancing problem," *Int. J. Prod. Econ.*, vol. 102, no. 2, pp. 226–243, Aug. 2006.

[12] R. Gamberini, E. Gebennini, A. Grassi, and A. Regattieri, "A multiple single-pass heuristic algorithm solving the stochastic assembly line rebalancing problem," *Int. J. Prod. Res.*, vol. 47, no. 8, pp. 2141–2164, Apr. 2009.

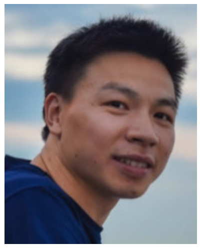
[13] U. Girit and M. Azizoğlu, "Rebalancing the assembly lines with total squared workload and total replacement distance objectives," *Int. J. Prod. Res.*, vol. 59, no. 22, pp. 6702–6720, Nov. 2021.

[14] N. Grangeon, P. Leclaire, and S. Norre, "Heuristics for the re-balancing of a vehicle assembly line," *Int. J. Prod. Res.*, vol. 49, no. 22, pp. 6609–6628, Nov. 2011.

[15] E. C. Harrington, "The desirability function," *Ind. Qual. Control*, vol. 21, no. 10, pp. 494–498, 1965.

[16] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MICH, USA: Univ. Michigan Press, 1975.
[17] A. Karas and F. Ozcelik, "Assembly line worker assignment and rebalancing problem: A mathematical model and an artificial bee colony algorithm," *Comput. Ind. Eng.*, vol. 156, pp. 1–16, Jun. 2021.
[18] J. F. Kottas and H. S. Lau, "A cost-oriented approach to stochastic line balancing," *AIIE Trans.*, vol. 5, pp. 164–171, Jun. 1973.
[19] O. Kramer, *Genetic Algorithm Essentials* (Studies in Computational Intelligence), vol. 679. Cham, Switzerland: Springer, 2017.
[20] Y. Li and T. O. Boucher, "Assembly line balancing problem with task learning and dynamic task reassignment," *Int. J. Adv. Manuf. Technol.*, vol. 88, nos. 9–12, pp. 3089–3097, Feb. 2017.
[21] Y. Li, "The type-II assembly line rebalancing problem considering stochastic task learning," *Int. J. Prod. Res.*, vol. 55, no. 24, pp. 7334–7355, 2017.
[22] Y. Li, Z. Li, and F. Saldanha-da-Gama, "New approaches for rebalancing an assembly line with disruptions," *Int. J. Comput. Integr. Manuf.*, pp. 1–18, Jun. 2021, doi: 10.1080/0951192X.2021.1925967.
[23] J. Liang, S. Guo, and W. Xu, "Balancing stochastic mixed-model two-sided disassembly line using multiobjective genetic flatworm algorithm," *IEEE Access*, vol. 9, pp. 138067–138081, 2021, doi: 10.1109/ACCESS.2021.3117070.
[24] K. Meng, Q. Tang, Z. Zhang, and X. Qian, "An improved lexicographical whale optimization algorithm for the type-II assembly line balancing problem considering preventive maintenance scenarios," *IEEE Access*, vol. 8, pp. 30421–30435, 2020, doi: 10.1109/ACCESS.2020.2972619.
[25] A. Nourmohammadi, M. Fathi, M. Zandieh, and M. Ghobakhloo, "A water-flow like algorithm for solving U-shaped assembly line balancing problems," *IEEE Access*, vol. 7, pp. 129824–129833, 2019, doi: 10.1109/ACCESS.2019.2939724.
[26] F. S. Oliveira, K. Vittori, R. M. O. Russel, and X. L. Travassos, "Mixed assembly line rebalancing: A binary integer approach applied to real world problems in the automotive industry," *Int. J. Automot. Technol.*, vol. 13, no. 6, pp. 933–940, Oct. 2012.
[27] R. M. Pérez-Wheelock and V.-N. Huynh, "Balancing assembly lines and matching demand through worker reallocation," in *Proc. 15th Int. Conf. Model. Decisions Artif. Intell.*, vol. 11144, 2018, pp. 206–217.
[28] E. Sanci and M. Azizoğlu, "Rebalancing the assembly lines: Exact solution approaches," *Int. J. Prod. Res.*, vol. 55, no. 20, pp. 5991–6010, Oct. 2017.
[29] A. Scholl, *Balancing and Sequencing of Assembly Lines*, 2nd ed. Heidelberg, Germany: Physica-Verlag, 1999.
[30] A. Scholl, "Data of assembly line balancing problems," Schriften zur Quantitativen Betriebswirtschaftslehre, Technische Universität, Darmstadt, Germany, Tech. Rep. 16/93, Nov. 1993.
[31] F. Serin, S. Mete, and E. Çelik, "An efficient algorithm for U-type assembly line re-balancing problem with stochastic task times," *Assem. Autom.*, vol. 39, no. 4, pp. 581–595, Sep. 2019.
[32] B. L. Song, W. K. Wong, J. T. Fan, and S. F. Chan, "A recursive operator allocation approach for assembly line-balancing optimization problem with the consideration of operator efficiency," *Comput. Ind. Eng.*, vol. 51, no. 4, pp. 585–608, Dec. 2006.
[33] K. Wu, Y. Zhou, and N. Zhao, "Variability and the fundamental properties of production lines," *Comput. Ind. Eng.*, vol. 99, pp. 364–371, Sep. 2016.
[34] C. Yang, J. Gao, and L. Sun, "A multi-objective genetic algorithm for mixed-model assembly line rebalancing," *Comput. Ind. Eng.*, vol. 65, no. 1, pp. 109–116, May 2013.
[35] J. Zha and J.-J. Yu, "A hybrid ant colony algorithm for U-line balancing and rebalancing in just-in-time production environment," *J. Manuf. Syst.*, vol. 33, no. 1, pp. 93–102, Jan. 2014.
[36] Y. Zhang, X. Hu, and C. Wu, "Heuristic algorithm for type II two-sided assembly line rebalancing problem with multi-objective," in *Proc. MATEC Web Conf.*, 2018, Art. no. 03063, doi: 10.1051/matecconf/201817503063.
[37] Y. Zhang, X. Hu, and C. Wu, "A modified multi-objective genetic algorithm for two-sided assembly line re-balancing problem of a shovel loader," *Int. J. Prod. Res.*, vol. 56, no. 9, pp. 3043–3063, May 2018.
[38] Y. Zhang, X. Hu, and C. Wu, "Improved imperialist competitive algorithms for rebalancing multi-objective two-sided assembly lines with space and resource constraints," *Int. J. Prod. Res.*, vol. 58, no. 12, pp. 3589–3617, Jun. 2019.

RANDALL MAURICIO PEREZ-WHEELOCK received the bachelor's degree in industrial and systems engineering from Universidad Thomas More, Nicaragua, in 2003, and the Master of Business Administration degree from the University of Tsukuba, Japan, in 2007. He is currently a Ph.D. candidate at Japan Advanced Institute of Science and Technology, Nomi, Ishikawa, Japan. He is also a Business Planner at Oyabe Seiki Co. Ltd., Oyabe, Toyama, Japan. His research interests include applications of operations research and data science in production and logistics systems.



WEI OU received the Ph.D. degree in knowledge science from the Japan Advanced Institute of Science and Technology, in 2018. He is currently working as an Assistant Professor at the International Business School, Zhejiang Gongshang University, China. His current research interests include machine learning, recommender systems, and natural language processing.

PISAL YENRADEE received the B.Eng. degree in production engineering from the King Mongkut's Institute of Technology and the M.Eng. and D.Eng. degrees in industrial engineering and management from the Asian Institute of Technology. He is currently an Associate Professor at the Industrial Engineering and Logistics Systems Program, Sirindhorn International Institute of Technology (SIIT), Thammasat University. His research interests include production planning and inventory control systems, JIT, ERP, TOC, finite capacity scheduling systems, aggregate production planning, master production scheduling, supply chains, and applied operations research.

VAN-NAM HUYNH (Member, IEEE) received the Ph.D. degree in mathematics from the Vietnam Academy of Science and Technology, in 1999. He is currently a Professor at the School of Knowledge Science, Japan Advanced Institute of Science and Technology (JAIST). His current research interests include machine learning and data mining, AI reasoning, argumentation, multi-agent systems, decision analysis and management science, and kansei information processing and applications. He currently serves as an Area Editor for *International Journal of Approximate Reasoning*, the Editor-in-Chief for *International Journal of Knowledge and Systems Science*, and the Editorial Board Member of the *Array* journal.