Dual-Resource Allocation Problem in a Flow Shop under Human Behavior Uncertainties

1st Yaheng Cui
Normandie Univ., LMAH UR 3821
Université Le Havre Normandie
76600 Le Havre, France
cuiyaheng@gmail.com

2nd Ibrahima Diarrassouba
Normandie Univ., LMAH UR 3821
Université Le Havre Normandie
76600 Le Havre, France
diarrasi@univ-lehavre.fr

3rd Chenghao Wang
LAMIH, CNRS, UMR 8201
Univ. Polytechnique Hauts-de-France
59313 Valenciennes, France
chenghao.wang@uphf.fr

***Abstract*—Industry 5.0 emphasizes human-centered production environments, promoting collaboration between humans and smart machines. This shift introduces new scheduling challenges due to the stochastic nature of human behavior—challenges that traditional deterministic models cannot adequately address. This paper studies a stochastic dual-resource allocation problem in a flow shop setting, where worker behavior impacts production outcomes through random breaks. We model worker states using Markov chains and apply a Sample Average Approximation method to minimize the expected makespan for a given number of jobs. Numerical experiments demonstrate the effectiveness of our approach. We also study the impact of the behavior of the workers on the overall production performance.**

***Index Terms*—Industry 5.0, stochastic programming, production scheduling, human-centric manufacturing systems**

I. Introduction

The advent of Industry 5.0 marks a shift from automation-focused production systems to a human-centric paradigm where humans and smart machines collaborate harmoniously [1]. While Industry 4.0 emphasized technological integration, such as IoT, AI, and robotics, Industry 5.0 highlights the importance of human creativity, adaptability, and decision-making in advanced manufacturing systems [2, 3]. This integration introduces new challenges in *production scheduling*, particularly due to the inherent *uncertainty in human behavior*.

In traditional production scheduling, deterministic models often assume predictable system performance, focusing on machine efficiency and static resource availability. However, considering human behavior in Industry 5.0 working environments introduces dynamic and stochastic issues, such as fluctuations in worker performance, fatigue, skill variability, and decision-making delays [4, 5, 6].

Stochastic optimization plays a pivotal role in addressing the challenges of human-centric Industry 5.0 environments, where uncertainty in human behavior is a critical factor [4, 5]. The stochastic models can incorporate randomness and probabilistic behaviors into decision-making, making them well-suited to account for variability in human performance while aligning with the collaborative principles of Industry 5.0. Leveraging Markov Chains, the dynamic fluctuations in worker's sequential productivity can be effectively captured, yielding more accurate predictions and adaptable scheduling solutions. Unlike traditional deterministic approaches, stochastic models optimize schedules under uncertainty by balancing machine utilization with human-centric constraints. This ensures that schedules remain feasible and effective in dynamic, collaborative environments, thereby enhancing system resilience and providing robust solutions to mitigate disruptions.

Worker's performance can be represented as a time-varying stochastic process, such as a Markov chain [1, 7], to reflect transitions between different states of productivity. Chance constraints can also be employed to ensure that the critical production targets are met with a high level of confidence, provided the resulting processing times follow a known probability distribution. Stochastic optimization herein not only addresses uncertainties but also supports the human-machine synergy central to Industry 5.0. This approach improves worker's satisfaction, enhances productivity, and ensures effective resource utilization, thereby contributing to the human-centric vision of Industry 5.0.

In this paper, we consider a dual-resource assignment problem in a flow shop production system where the availability of the workers is stochastic. We propose an optimization model to find optimal worker-to-workstation assignments and job scheduling under this uncertainty, enhancing the overall system's resilience. The contributions of this paper are three-fold: (i) we develop a stochastic optimization framework that accounts for dynamic worker performance via Markov Chain; (ii) we incorporate a Sample Average Approximation (SAA) method [8] to solve the stochastic model to minimize the expected makespan; and (iii) through a series of numerical experiments, we analyze the impact of human behavior uncertainty on scheduling efficiency considering random data in Industry 5.0 environments.

The remainder of this paper is organized as follows: Section 2 presents the problem description, detailing the challenges and the context of the scheduling problem. Section 3 provides the formulation of the stochastic production scheduling model, including the mathematical model and solution methodology. Section 4 describes the numerical experiments, including the model settings, scenario generation, and performance evaluation. Finally, Section 5 concludes the paper, summarizing the

Partial funding for this project has been provided by the European Union and the Region of Normandy through the RIN regional research program by funding the AntiHPert project.

findings and suggesting potential directions for future research.

II. Problem Description

In the context of Industry 5.0, production scheduling must consider both machine constraints and the inherent uncertainty in human performance. Worker's productivity can significantly vary due to factors such as fatigue, motivation, and skill levels, which can substantially affect overall production system efficiency. To address these challenges, we propose using Markov Chains to model the stochastic behavior of workers and stochastic programming to optimize scheduling decisions under uncertainty.

The production environment consists of a set of jobs J that require sequential processing on multiple workstations I. Each job also requires the presence of a worker $w \in W$, which introduces dual-resource constraints. Each job $j \in J$ has a nominal processing time on workstation $i \in I$, denoted by p_{ij}. We consider a flow shop production system, that is, each job must pass trough all workstations in sequence. Also, each worker can take one or more breaks during the time horizon. In the case where a worker is on break, the jobs being assigned to the workstation are not processed until the worker comes back from his/her break.

We assume that the availability of each worker is modeled by a Markov chain with two states: Productive and Non-Productive state. For handling the uncertainties in the problem, we generate multiple scenarios to reflect different possible realizations of workers availability. The scenarios are generated by simulating the worker profiles using Markov chains, where the transition matrices capture the probability of a worker being available at a given time based on their previous state.

The objective of the Stochastic Dual-Resource Allocation Problem (DRAP) is to determine an assignment of the workers to the workstations such that the average makespan for all scenarios is minimum and ensuring production plan effficiency over a given period of time.

For the following subsections, we will explicitly give the details for stochastic optimization approach and the modeling of the worker's performance with using Markov Chain which are used in the rest of the paper.

A. Sample Average Approximation method to Handle Uncertain Human Behaviors

The worker's performance, represented as the worker's availability, is assumed to be known in advance in classical production problems. In this case, the decision-makers typically use deterministic methods to solve the problem. However, in practice, worker's performance is subject to stochastic fluctuations due to correlated factors such as fatigue, motivation, and skill levels. Consequently, the obtained solution of a deterministic assumption can lead to suboptimal or unreliable decisions. To address this uncertainty, we adopt a stochastic model that captures the random nature of worker availability.

To address the stochastic nature of the problem, we employ the Sample Average Approximation method [8] within a set of scenarios from the underlying probability distribution, and use these scenarios to approximate expected outcomes, resulting in more robust and realistic decisions for the DRAP.

The corresponding model assigns workers, sequences jobs, and allocates resources to minimize the expected makespan, which subjects to worker availability, precedence rules, and capacity limits.

B. Markov Chain based Modeling of Worker's Performance

To capture the uncertainty in worker's performance within the production scheduling problem, we employ a **Markov Chain** [7, 9] model that represents the dynamic transitions between different performance states of the worker. The worker's performance is characterized by a discrete set of states, which include *productive states* and *non-productive states*.

1) States:

- **Productive State**: This state represents the period when the worker is actively engaged in tasks, which may fall into two distinct operational realistic cases.
  - *Human-Interaction-Dependent Jobs*: In this case, the time spent completing a job is heavily influenced by dynamic human factors such as the worker's skill level, cognitive or physical fatigue, and task familiarity. While the baseline processing time may follow a known distribution (e.g., exponential or normal), its parameters (e.g., mean, variance) are modulated by these human variables. For instance, a highly skilled worker might reduce the mean processing time, while fatigue could increase variability or introduce delays.
  - *Surveillance-Dependent Jobs*: Here, the job's completion relies primarily on human oversight rather than direct interaction, meaning the nominal processing time is deterministic (e.g., fixed duration). However, interruptions due to scheduled or unscheduled worker breaks—such as rest periods, shift changes, or distractions—can delay progress. These breaks extend the effective processing time but do not inherently alter the task's baseline duration.

  In both cases, the worker contributes to job progress as scheduled, but the underlying drivers of processing time differ: human-interaction-dependent tasks are shaped by intrinsic worker-state variability, while surveillance-dependent tasks are affected extrinsically by interruptions. The choice of model depends on whether the task's completion is governed by active human effort or passive monitoring.
- **Non-Productive State**: This state represents periods when the worker is not contributing to job completion, such as during breaks, fatigue, or delays waiting for machines or other resources. In this state, the worker is idle or performing minimal tasks, resulting in a delay in job progress.

The transitions between these states are governed by a set of probabilities that are represented in the transition matrix P. The key transition probabilities are: a) P_{PP} is the probability that the worker remains in the productive state; b) P_{PN} is the probability that the worker transitions from the productive state to the non-productive state; c) P_{NP} is the probability that the worker transitions from the non-productive state to the productive state; d) P_{NN} is the probability that the worker remains in the non-productive state.

The transition matrix P herein can be expressed as:

$$P = \begin{pmatrix} P_{PP} & P_{PN} \\ P_{NP} & P_{NN} \end{pmatrix},$$

where $P_{PP} + P_{PN} = 1$ and $P_{NP} + P_{NN} = 1$.

2) Modeling of Worker's Performance in Scheduling: In production scheduling, the state of the worker at any given moment directly influences the processing time of jobs. When a worker is in the productive state, tasks are completed more efficiently, whereas a transition to a non-productive state increases the processing time due to downtime or slower task completion. This time-varying behavior can be incorporated into a stochastic programming framework, allowing for dynamic scheduling decisions that minimize delays and optimize resource utilization, despite the uncertainty introduced by variable human performance.

3) Stationary Distribution of the Markov Chain: In this model, the humans' actions are defined by a Markov chain with one productive state and several non-productive states, each associated with a time length. The expected production time and non-production time can be calculated as follows:

Let the Markov chain transition matrix P describe the probabilities of transitioning between the states. Assume there are n total states, with one productive state (denoted as state 1) and $n-1$ non-productive states (denoted as states 2, 3, ..., n). The transition matrix P is an $n \times n$ matrix, which can be written as:

$$P = \begin{pmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{pmatrix},$$

where P_{ij} represents the probability of transitioning from state i to state j.

Let t_i denote the expected time spent in state i. Specifically, we define:

- t_1 as the time spent in the productive state.
- $t_2, t_3, \ldots, t_n$ as the time spent in each of the non-productive states.

The stationary distribution [10] of the Markov Chain represents the long-term probabilities of a worker being in each state (productive or non-productive) after many transitions. It is crucial because it allows us to calculate the expected production and non-production times, reflecting the worker's behavior over time. By using the stationary distribution, we can make informed decisions about resource allocation and scheduling, optimizing the system's performance under uncertainty and accounting for the dynamic nature of human behavior in industrial settings.

To calculate the expected time spent in each state, we first determine the *stationary distribution* $\boldsymbol{\pi} = (\pi_1, \pi_2, \ldots, \pi_n)$, which represents the long-term probability of being in each state. The stationary distribution satisfies the following system of equations [10]:

$$\boldsymbol{\pi} P = \boldsymbol{\pi}, \sum_{i=1}^{n} \pi_i = 1,$$

where π_i is the probability of being in state i in the long run.

Once the stationary distribution $\boldsymbol{\pi}$ is determined, the expected production and non-production times can be calculated as follows:

- The expected production time is the expected time spent in the productive state, weighted by the stationary probability of being in the productive state:

$$\mathbb{E}[\text{Production Time}] = \pi_1 \cdot t_1$$

- The expected non-production time is the sum of the expected times spent in each non-productive state, weighted by the respective stationary probabilities:

$$\mathbb{E}[\text{Non-Production Time}] = \sum_{i=2}^{n} \pi_i \cdot t_i$$

III. Problem Formulation

In this section, we provide two Mixed Integer Linear Programming models, for the *Dual-Resource Allocation problem (DRAP) Flow-Shop*. In the first formulation, we suppose the problem is deterministic and the workers availability profile is given. The second formulation considers the data to be stochastic and is designed to find an optimal solution regarding scenarios.

A. Assumptions

1) Each worker can be assigned to at most one workstation, and each workstation should be assigned exactly one worker.
2) The number of workers must be greater than or equal to the number of workstations, allowing flexibility in worker assignments.
3) Each job belongs to a specific type and each type has a defined nominal processing time at each workstation.
4) Worker's availability is modeled using a Markov chain, transitioning between productive and non-productive states, affecting final processing times of associated job tasks.
5) Jobs must follow a strict sequence through all workstations, with each job processed at only one workstation at a time.

B. Notations

Parameters:

- W: Set of workers.
- I: Set of workstations.
- J: Set of jobs with a predefined sequence for operation, indexed by j.
- T_j: Type of job j.
- p_{ij}: Nominal processing time for job j on workstation i, based on its type T_j.

- a_{wt}: Binary parameter, taking value 1 if worker w is available at time t, 0 otherwise.
- S: Set of scenarios.

Decision Variables:

- x_{wi}: Binary variable, taking value 1 if worker w is assigned to workstation i, 0 otherwise.
- s_{ij}: Start time of job j on workstation i.
- c_{ij}: Completion time of job j on workstation i.

C. Deterministic Optimization Model

With above-mentioned notations and decision variables, the corresponding deterministic formulation of the DRAP can be written as:

$$\text{Minimize } C_{\max} = \max_{i\in I, j\in J}(c_{ij}) \quad (1)$$

$$\text{Subjects to } \sum_{i\in I} x_{wi} \le 1, \quad \forall w \in W \quad (2)$$

$$\sum_{w\in W} x_{wi} = 1, \quad \forall i \in I \quad (3)$$

$$p_{ij} = \sum_{w\in W} \sum_{t=s_{ij}}^{c_{ij}} a_{wt} \cdot x_{wi}, \quad \forall i \in I, j \in J \quad (4)$$

$$s_{i,j+1} \ge c_{ij} + 1, \quad \forall i \in I, j \in J \quad (5)$$

$$s_{i+1,j} \ge c_{ij} + 1, \quad \forall i \in I, j \in J \quad (6)$$

$$x_{wi} \in \{0,1\}, \quad \forall i \in I, w \in W \quad (7)$$

$$s_{ij}, c_{ij} \ge 0, \quad \forall i \in I, j \in J \quad (8)$$

where the objective (1) of the problem aims to minimize the makespan of a given set of jobs. Constraints (2) ensure that each worker can be assigned to at most one workstation. Constraints (3) guarantee that each workstation should be assigned exactly one worker. Constraints (4) specify the start time and completion time of a job j processed on a workstation i assisted by a work w to ensure defined processing time. Constraints (5) indicate the sequential order of the jobs on each workstation. Constraints (6) specify the sequential order of different workstations for each job. Constraints (7)–(8) define the feasible domain of decision variables.

It is worthy to remark that constraints (4) are nonlinear. We propose a linearization by first discretizing the time space and then introduce a new boolean variable x_{ijt} taking value 1 if job j is occupied on workstation i at time t. Then, the above constraints (4) can be linearized as follows:

$$\sum_{t\in T} x_{ijt} a_{wt} \ge p_{ij} x_{wi}, \quad \forall i \in I, j \in J, w \in W \quad (9)$$

$$\sum_{t\in T} x_{ijt} a_{wt} \le p_{ij} + M(1 - x_{wi}), \forall i \in I, j \in J, w \in W \quad (10)$$

$$s_{ij} \le t x_{ijt} + M(1 - x_{ijt}), \quad \forall i \in I, j \in J, t \in T \quad (11)$$

$$c_{ij} \ge t x_{ijt}, \quad \forall i \in I, j \in J, t \in T \quad (12)$$

$$\sum_{t\in T} x_{ijt} = c_{ij} - s_{ij} + 1, \quad \forall i \in I, j \in J \quad (13)$$

where Constraints (9) and (10) ensure that total available times of a worker assigned on the workstation is able to satisfy the defined nominal processing time of a processed job. Constraints (11) and (12) define the start time and completion time of the processed job on the corresponding workstation. Constraints (13) corresponds to the actual processing time incorporating the worker's availability.

D. Stochastic Model Formulation

Recall that S is the set of scenarios, p_s is the probability of scenario $s \in S$ to occur and $\sum_{s\in S} p_s = 1$. Therefore, the stochastic DRAP can be formulated as the following Mixed Integer Program:

$$\text{Minimize} \quad \sum_{s\in S} p_s C^s_{\max}$$

Subjects to:

$$C^s_{\max} \ge c^s_{ij}, \quad \forall i \in I, j \in J, s \in S$$

$$\sum_{i\in I} x_{wi} \le 1, \quad \forall w \in W$$

$$\sum_{w\in W} x_{wi} = 1, \quad \forall i \in I$$

$$\sum_{t\in T} x^s_{ijt} a^s_{wt} \ge p_{ij} x_{wi}, \quad \forall i \in I, j \in J, w \in W, s \in S$$

$$\sum_{t\in T} x^s_{ijt} a^s_{wt} \le p_{ij} + M(1 - x_{wi}), \quad \forall i \in I, j \in J, w \in W, s \in S$$

$$s^s_{ij} \le t x^s_{ijt} + M(1 - x^s_{ijt}), \quad \forall i \in I, j \in J, t \in T, s \in S$$

$$c^s_{ij} \ge t x^s_{ijt}, \quad \forall i \in I, j \in J, t \in T, s \in S$$

$$\sum_{t\in T} x^s_{ijt} = c^s_{ij} - s^s_{ij} + 1, \quad \forall i \in I, j \in J, s \in S$$

$$s^s_{i,j+1} \ge c^s_{ij} + 1, \quad \forall i \in I, j \in J, s \in S$$

$$s^s_{i+1,j} \ge c^s_{ij} + 1, \quad \forall i \in I, j \in J, s \in S$$

$$x_{wi} \in \{0,1\}, \quad \forall i \in I, w \in W$$

$$x^s_{ijt} \in \{0,1\}, \quad \forall i \in I, j \in J, t \in T, s \in S$$

$$s^s_{ij}, c^s_{ij} \ge 0, \quad \forall i \in I, j \in J, s \in S$$

IV. Numerical Experiments and Discussion

In this section, we present the numerical experiments to evaluate the performance of the proposed stochastic optimization model and to analyze the impact of the worker's availability profile on production performance.

The mathematical models are solved using CPLEX 20.1 and Python 3.8. Various problem settings are considered to assess the impact of different parameters, particularly focusing on manipulating the probability transition matrices of the worker profiles.

A. Algorithm Setting

Four solution methods are applied in the numerical analysis:

- **Method 1: Scenario-Based Model**
  Each scenario is solved individually with deterministic formulation for scenario-specific worker assignments.
- **Method 2: Stochastic Programming**

We solve the stochastic optimization model introduced in section III-D. The worker assignment is optimized across all scenarios using expectation-based decision-making.

- **Method 3: Deterministic Model Using Stable Markov States**
  Worker assignment is computed from the deterministic formulation, with stable Markov states as input.
- **Method 4: Worst Assignment Model**
  The worst assignment is made by matching stable Markov states to total workstation processing times. Job sequencing on workstations remains optimal.

B. Case Study

In this section, the effectiveness of the proposed algorithm is demonstrated and validated through numerical experiments. A series of test instances are randomly generated based on references [1]. The relevant parameters are summarized in Table I.

The test instances involve a flow shop consisting of several processing workstations. The table below provides key parameters, such as the number of jobs, number of workers, processing times, and break probabilities for workers in their Markov chains.

TABLE I
SIMULATION PARAMETERS FOR TEST INSTANCES

Parameter	Description
Number of Workers	4
Number of Workstations	4
Number of Jobs	6, 12
Number of Scenarios in Each Test Case	10
Job Processing Time Range	$U(4,8)$, $U(10,20)$
Break Probability of the Worker	0, 0.05, 0.1, 0.15, 0.2

TABLE II
RESULTS OF ALL TESTED SCENARIOS.

Group	Case	Method 1			Method 2 (Gap)	Method 3	Method 4
		Min	Max	Average			
short_short	0	-	-	53	53	61	53
	1	53.7	53.7	53.7	55.8 (5.69%)	64.18	55.7
	2	55.5	55.5	55.5	58 (10.06%)	67.69	58.0
	3	58.7	58.7	58.7	62.6 (17.79%)	71.61	62.5
	4	61.5	61.5	61.5	67.1 (22.04%)	75.90	67.1
	5	56.6	56.6	56.6	58.4 (9.17%)	67.73	64
short_long	0	-	-	145	145	145	145
	1	148	152	149.7	152.6 (5.83%)	147.3	152.6
	2	154	159	156.5	162.4 (12.39%)	155.4	162.2
	3	160	170	164.1	171.9 (19.38%)	164.5	171.5
	4	166	183	174.6	183.1 (26.89%)	174.5	182.6
	5	157	170	160.9	172.5 (19.79%)	158.46	164.3
long_short	0	-	-	91	91	91	91
	1	92	95	93.4	96.5 (7.22%)	104.2	96.5
	2	96	100	97.5	102.3 (13.67%)	109.9	102.3
	3	99	105	102.2	108.6 (20.67%)	116.3	108.6
	4	102	113	108.7	115.9 (28.78%)	123.4	115.6
	5	97	103	98.9	101.7 (13%)	107.7	110.5

Table II presents the experiments results obtained from three designed groups:

- `short_short`: 6 jobs with processing times $U(4,8)$.
- `short_long`: 6 jobs with processing times $U(10,20)$.
- `long_short`: 12 jobs with processing time $U(4,8)$.

The case numbering is as follows:

- **Case 0:** no breaks for all the workers (workers are robots).
- **Cases 1–4:** all the workers have the same breaking probability which is $(0.05 \times k)$, $k = 1, .., 4$.
- **Case 5:** all the four worker profiles are used for the production.

1) Key Observations:

a) Impact of Breaks (Case 0 vs. Cases 1–4): All groups achieve the best performance in Case 0 when the robot operates without breaks. In Cases 1–4, as the break probability increases, all objective values (Method 1, Method 2, etc.) tend to increase. The gap associated with Method 2 grows, indicating higher deviation as the chance of breaks increases.

b) Influence of Job Processing Time and Number of Jobs: Comparing `short_short` and `short_long` (both with 6 jobs), longer processing times in `short_long` lead to substantially higher objective values. For instance, in Case 0, the objective value (makespan) increases from 53 (short_short) to 145 (short_long). Although `long_short` has 12 jobs (double the number compared to the other two groups), its objective values remain lower than those in `short_long` due to the short processing times.

c) Worker Profiles (Cases 1–4 vs. Case 5):

- **Single Worker Profiles (Cases 1–4):** As the break probability increases from Case 1 to Case 4, there is a consistent deterioration in performance. The highest objective values and gaps are seen in Case 4, reflecting the negative impact of the worst-performing worker profile.
- **Mixed Worker Profile (Case 5):** The mixed worker profile generally yields objective values that are better than those from the worst assignment single profile (Case 4). The gap for Case 5 is often lower than that of Case 3-4, suggesting that mixing worker profiles can mitigate the negative effects of high break probabilities.

d) Performance of the different resolution methods: The analysis shows that the scenario-based solution (Method 1) consistently delivers the best performance across all cases, which is expected since the job assignments and scheduling are optimized for each scenario individually.

Moreover, in mixed worker profile Case 5, the stochastic optimal strategy (Method 2) often outperforms the worst assignment approach (Method 4)—the latter representing a situation where no optimization is performed—thereby demonstrating the practical utility of incorporating uncertainty into the optimization process even with a computational time limit. The fact that the stochastic solution is better even under a strict computational gap proves that our approach adds value by handling uncertainty more effectively, especially in complex, mixed-profile scenarios.

Additionally, the `short_short` test group consistently achieves better processing times than the `short_long` test group. This result suggests that the optimization tool can more easily reconcile job processing times with worker's productive states when the processing times are shorter or more homogeneous. In other words, when the processing times are compact (`short_short`), the scheduler has greater flexibility and efficiency in aligning the tasks with the worker's optimal productivity periods. This insight could guide future problem formulations or preprocessing steps to favor more uniform processing time distributions, potentially enhancing overall scheduling performance.

Furthermore, the Method 3 frequently provides solutions that deviate significantly from the stochastic optimum (Method 2). This gap indicates that the method used to estimate the actual processing time for jobs in this formulation may be inaccurate. Since the expected strategy does not capture the nuances of real-world variability as well as the other approach, it falls short in providing near-optimal results. This discrepancy highlights the need for a more refined estimation method or alternative approaches to better model the actual processing times, thereby improving the quality of the expected optimal strategy.

2) Management insight: The optimal performance is consistently achieved in Case 0, that is the situation where all the workers are robots. Higher break probabilities (Cases 1–4) lead to worse performance and greater variability, as evidenced by increasing gap percentages. Longer processing times, as observed in `short_long` tests, result in significantly larger values of makespan, even when the number of jobs is the same. Conversely, a higher number of jobs with short processing times (`long_short`) can maintain relatively lower objective values. There is a clear trade-off between job processing time and the number of jobs. Optimizing processing times is also essential and maybe potential aspect to improving system performance, potentially even more so than reducing the number of jobs.

V. Conclusion

In this paper, we have addressed a human-centric Dual-Resource Allocation Problem (DRAP) in a flow shop setting with stochastic worker behavior, extending the foundational work of [1]. Unlike existing methodologies that often simplify human resource dynamics (e.g., assuming deterministic behavior or homogeneous availability), our approach explicitly incorporates stochastic worker availability profiles in a stochastic mathematical model, bridging a critical gap in practical applicability. Our experiments demonstrate that the integrated SAA methodology outperforms deterministic approaches in stochastic settings, balancing realism and computational viability.

For future work, developing faster decomposition or heuristic algorithms tailored to our expanded problem formulation will be essential, particularly for scaling to industrial-sized instances. Further evaluation of practical aspects, such as diverse availability patterns and dynamic skill-fatigue interactions, will enhance realism. Integrating additional human factors (e.g., skill decay, fatigue dynamics) could refine the models, though this will necessitate advanced techniques to manage complexity. By grounding advancements in the gaps identified here, future research can build on both our constraints and the broader state-of-the-art to advance human-centric production systems.

References

[1] N. Bouaziz, B. Bettayeb, M. Sahnoun, and A. Yassine, "Incorporating uncertain human behavior in production scheduling for enhanced productivity in industry 5.0 context," *International Journal of Production Economics*, p. 109311, 2024.

[2] N. Katiraee, M. Calzavara, S. Finco, D. Battini, and O. Battaïa, "Consideration of workers' differences in production systems modelling and design: State of the art and directions for future research," *International Journal of Production Research*, vol. 59, no. 11, pp. 3237–3268, 2021.

[3] T. Mraihi, O. B. Driss, and H. B. El-Haouzi, "Distributed permutation flow shop scheduling problem with worker flexibility: Review, trends and model proposition," *Expert Systems with Applications*, vol. 238, p. 121947, 2024.

[4] Y. Liu, J. Fan, L. Zhao, W. Shen, and C. Zhang, "Integration of deep reinforcement learning and multi-agent system for dynamic scheduling of re-entrant hybrid flow shop considering worker fatigue and skill levels," *Robotics and Computer-Integrated Manufacturing*, vol. 84, p. 102605, 2023.

[5] Y. Liu, W. Shen, C. Zhang, and X. Sun, "Agent-based simulation and optimization of hybrid flow shop considering multi-skilled workers and fatigue factors," *Robotics and Computer-Integrated Manufacturing*, vol. 80, p. 102478, 2023.

[6] M. Fekri, M. Heydari, and M. M. Mazdeh, "Bi-objective optimization of flexible flow shop scheduling problem with multi-skilled human resources," *Engineering Applications of Artificial Intelligence*, vol. 133, p. 108094, 2024.

[7] C.-H. Lin, K.-J. Wang, A. A. Tadesse, and B. H. Woldegiorgis, "Human-robot collaboration empowered by hidden semi-markov model for operator behaviour prediction in a smart assembly system," *Journal of Manufacturing Systems*, vol. 62, pp. 317–333, 2022.

[8] A. J. Kleywegt, A. Shapiro, and T. Homem-de Mello, "The sample average approximation method for stochastic discrete optimization," *SIAM Journal on optimization*, vol. 12, no. 2, pp. 479–502, 2002.

[9] V. Vijayakumar, F. Sgarbossa, W. P. Neumann, and A. Sobhani, "Framework for incorporating human factors into production and logistics systems," *International Journal of Production Research*, vol. 60, no. 2, pp. 402–419, 2022.

[10] C. Paige, G. P. Styan, and P. G. Wachter, "Computation of the stationary distribution of a markov chain," *Journal of Statistical Computation and Simulation*, vol. 4, no. 3, pp. 173–186, 1975.