Non-Identical Parallel Machine Scheduling Problem with Worker Fatigue Consideration

Achmad Pratama Rifai[1]*, Suma Samala Jiwangga[2], Andri Nasution[3]

[1,2] Department of Mechanical and Industrial Engineering, Faculty of Engineering, Universitas Gadjah Mada, Special Region of Yogyakarta, 55284, Indonesia

[3] Department of Industrial Engineering, Faculty of Engineering, Universitas Sumatera Utara, Medan, Indonesia

*Corresponding author
E-mail: achmad.p.rifai@ugm.ac.id[1], suma.samala.jiwangga@mail.ugm.ac.id[2], andri.nasution@usu.ac.id[3]

Abstract

Machine scheduling critically influences production efficiency. This study focuses on the Non-Identical Parallel Machine Scheduling Problem (NIPMSP) and explicitly incorporates human factors, namely worker fatigue, into an exact optimization framework. A mixed-integer nonlinear programming (MINLP) model where fatigue accumulation and recovery dynamically affect processing times. The model is verified and validated using secondary datasets and manual calculations, and solved with Gurobi across multiple configurations (2–3 machines; 6–20 jobs). Results show that the model consistently produces feasible solutions; higher fatigue rates increase makespan, whereas higher recovery rates reduce it. Sensitivity analyses on the fatigue rate, recovery rate, and percentage of jobs influenced by fatigue further confirm these trends. Incorporating worker fatigue enables more realistic makespan minimization and provides actionable managerial insights to balance throughput and workforce well-being.

Keywords: Machine scheduling, NIPMSP, worker fatigue, mathematical modelling, makespan

1. Introduction

Machine scheduling plays a fundamental role in manufacturing systems by determining the allocation and sequencing of jobs on available resources to achieve operational objectives such as reducing makespan, improving resource utilization, and increasing production efficiency. As manufacturing environments become more complex, scheduling decisions must account for variations in machine capabilities, workflow structures, and production requirements [1,2]. Among the various scheduling configurations, parallel machine systems are widely adopted to increase production capacity and flexibility. In practice, however, machines often differ in terms of processing capabilities and performance, resulting in machine-dependent processing times for the same job. This configuration is referred to as the Non-Identical Parallel Machine Scheduling Problem (NIPMSP), where the primary challenge lies in assigning and sequencing jobs across heterogeneous machines in a manner that optimizes a predefined performance measure, such as makespan minimization.

According to Balin [3], one of the primary objectives in machine scheduling problems is to minimize the make span (Cmax), which represents the total time required to complete all scheduled jobs. In addition, worker factors that operate the machine play a significant role in scheduling decisions. The human factor is crucial and must be taken into account during the scheduling process. One of the key aspects of worker-related factors that can affect processing time is worker fatigue. Although the impact of fatigue on worker performance is widely recognized, its integration into formal scheduling optimization models remains highly limited.

This study addresses the Non-Identical Parallel Machine Scheduling Problem (NIPMSP) with the objective of minimizing the make span while incorporating worker fatigue as a consideration. Fatigue, in this context, generally refers to the continuous consumption of energy resulting from doing work continuously, which can cause a series of changes in personal physiology and psychology and lead to a decrease in operational ability [4]. Several studies on fatigue have shown that fatigue levels tend to increase exponentially over time [5-6].

Considering the existing issues and the absence of a mathematical model that adequately captures the described problem, this research aims to address the Non-Identical Parallel Machine Scheduling Problem

In press

by incorporating worker fatigue as a critical factor. An exact method will be employed to solve this problem, focusing on the development of a mathematical optimization model. The proposed model will be verified through computational experiments using secondary data and solved using the Gurobi Optimizer. Additionally, sensitivity analysis is conducted to evaluate the impact of fatigue and recovery rates on the overall scheduling performance. The results of this study provide insight into more realistic scheduling strategies that better reflect the human dimension of manufacturing operations and offer practical guidance for decision-makers seeking to balance efficiency and worker well-being. Despite its practical importance, worker fatigue has rarely been integrated into exact NIPMSP models, creating a gap that this study addresses. Recent advances in unrelated/non-identical parallel machine scheduling also address machine availability and eligibility constraints, controllable processing times and setups, as well as evolutionary approaches for large instances, underscoring the continued need for realistic human-centric models [7]; [8]; [9].

2. Literature Review

The problem of scheduling machinery on a job and the operator of a particular machine in a production process in the manufacturing industry is an important thing in a company's production management [10]. This problem becomes important due to its direct impact on productivity, operational cost, and resource utilization. Effective scheduling will determine exactly which jobs are assigned to which machines and at what times, ensuring that operations are completed efficiently. Among the various types of scheduling problems, the Parallel Machine Scheduling Problem (PMSP) has received significant attention, particularly its non-identical variant, which is called the Non-Identical Parallel Machine Scheduling Problem due to its greater realism and complexity.

Parallel machines refer to a type of manufacturing process in which multiple machines operate simultaneously to perform a set of tasks or processes [11]. The term non-identical machines refers to non-identical machines is the characteristics of the machines that are different from one another, which causes differences in processing time between machines. The complexity of NIPMSP increases as it must consider not only job allocation but also the differences in machine capabilities, making it an NP-hard problem in many cases [12].

Several studies have proposed optimization models and heuristics to solve NIPMSP. Balin [3] and Noman et al. [10] focused on minimizing the makespan using exact methods, while others, such as Santoro & Junqueira [7] introduced heuristics to handle larger problem instances. Additional constraints such as release time [13], flexible processing time [8], and dependent setup time [9] have been introduced to capture the complexity of real-world production systems.

Traditionally, research on scheduling problems has focused heavily on machine-related parameters. However, in practical applications, human resources, particularly workers, play a crucial role in job execution. Multi-skilled worker scheduling, for instance, has gained interest due to its flexibility and cost-efficiency. Liu & Yang [14] emphasized the importance of assigning workers with varying skill levels to machines, showing that worker capabilities significantly influence processing time and overall system efficiency. Moreover, the “learning effect” has been introduced in some models to reflect how workers become more efficient over time [9]. While learning effects improve processing time as workers gain experience, few studies have addressed the inverse phenomenon: worker fatigue. A recent agent-based simulation for hybrid flow shops explicitly integrates multi-skilled workers and fatigue factors, coupling simulation with optimization to capture dynamic human performance [15].

Worker fatigue refers to the decline in mental and physical performance due to prolonged work without adequate rest. It manifests as increased processing time, reduced accuracy, and higher error rates. Fatigue modeling has been applied in limited contexts, such as single-machine systems [16] or hybrid flow shops with agent-based simulation [15]. However, these studies often treat fatigue qualitatively or in simplified forms. Fatigue can be mathematically modeled using exponential growth functions to represent fatigue accumulation and exponential decay for recovery [5],[17]. Ferjani et al. [18] introduced a concept called "processing time deterioration," where processing time increases as fatigue builds up. Despite its relevance, the integration of such fatigue functions into complex production scheduling models like NIPMSP remains limited.

Recent studies have demonstrated significant progress in solving parallel machine scheduling problems through the incorporation of increasingly realistic production constraints and advanced optimization techniques. Noman et al. [10] developed a tabu search algorithm enhanced with lower-bound information and exact solutions to minimize makespan in non-identical parallel machine environments. Zhang et al. [9] proposed a combinatorial evolutionary algorithm that simultaneously considers sequence-dependent setups, machine-dependent setup times, limited worker resources, and learning effects, highlighting the growing interest in integrating human-related factors into scheduling decisions. Wang et al. [8]

In Press

investigated unrelated parallel machine scheduling with controllable processing times and setup activities, demonstrating the impact of processing-time flexibility on scheduling performance. More recently, Santoro and Junqueira [7] introduced optimization models that incorporate machine availability and eligibility constraints, providing a more realistic representation of industrial production systems. Furthermore, Liu et al. [15] integrated multi-skilled workers and fatigue effects into a hybrid flow shop environment using agent-based simulation and optimization, emphasizing the importance of human factors in production planning. These developments indicate a clear trend toward more realistic scheduling models that simultaneously consider machine heterogeneity, operational constraints, and workforce-related characteristics.

Recent studies published in 2025 further demonstrate the growing interest in incorporating human-related and heterogeneous resource considerations into scheduling optimization. Li et al. [19] investigated the Flexible Job-Shop Scheduling Problem with Heterogeneous Workers (FJSPWH) and proposed a parallel cooperative co-evolutionary algorithm that explicitly integrates worker heterogeneity into scheduling decisions. Similarly, Liu et al. [20] addressed a multi-objective flexible flow shop scheduling problem using an improved NSGA-II algorithm, highlighting the increasing adoption of advanced optimization techniques for complex production environments. These developments indicate a clear research trend toward human-centered scheduling models that account for worker-related variability and resource heterogeneity. Nevertheless, the explicit integration of worker fatigue dynamics into exact optimization models for Non-Identical Parallel Machine Scheduling Problems (NIPMSPs) remains limited, thereby motivating the development of the proposed fatigue-aware MINLP framework.

Most existing NIPMSP models focus on machine-based constraints and performance optimization, with limited attention to human factors. Although some studies have considered resource allocation involving multi-skilled workers and learning effects, the deterioration of worker performance due to fatigue has not been fully explored, especially in the context of exact optimization models. This opens a significant research gap for modeling human-centric constraints in scheduling. This study seeks to address that gap by integrating a nonlinear fatigue function into an exact mathematical model for NIPMSP. By doing so, it aims to produce more realistic scheduling outputs that account for both machine heterogeneity and worker well-being.

3. Problem Formulation

The scheduling environment considered in this study consists of a set of jobs processed on multiple non-identical parallel machines. Each machine is operated by a dedicated worker, and processing times vary across machine-worker combinations due to differences in machine characteristics and worker performance. The objective is to determine the optimal job assignment and sequencing decisions while accounting for worker fatigue and recovery dynamics. This scheduling environment serves as the basis for the proposed fatigue-aware NIPMSP formulation.

This research focuses on designing a Non-Identical Parallel Machine Scheduling Problem (NIPMSP) model by incorporating the worker fatigue factor. In this model, the scheduling problem involves a set of N jobs that must be completed using M machines capable of operating in parallel. Each job j is assigned to one of the M machines and may have a different processing time depending on the machine assigned (Pij). Accordingly, a dataset containing the processing times for each job on each machine is required. The processing time data used in this study were adapted from Ferreira et al. [22], which provides benchmark processing-time data for production scheduling applications. To accommodate the objectives of the present study, the original dataset was modified by incorporating worker fatigue and recovery considerations while preserving the non-identical processing time characteristics among machine-worker combinations. Furthermore, it is assumed that each machine is operated by a dedicated worker (one worker per machine).

The scheduling environment considered in this study consists of a set of jobs that must be processed on multiple non-identical parallel machines, where each machine is operated by a dedicated worker. Due to differences in machine characteristics and worker performance, the processing time of a given job may vary across machine–worker combinations. Consequently, assigning all jobs to the fastest machine–worker pair may lead to excessive workload concentration, increased idle time on other resources, and ultimately a larger make span. Therefore, an effective scheduling strategy must balance job assignments among available machine–worker pairs while accounting for their heterogeneous processing capabilities. In addition to machine heterogeneity, worker performance may also vary over time due to fatigue accumulation during production activities. To capture this phenomenon, the worker fatigue mechanism incorporated in the present study is adapted from the

In Press

model proposed by Liu et al. [15], in which fatigue progression is represented through an exponential function influenced by job processing time and recovery duration.

3.1. Worker Fatigue and Recovery Model

Worker fatigue has been widely recognized as an important human factor affecting operational performance, as it can reduce productivity, increase processing times, and negatively influence decision-making and work quality. In Liu et al. [15]It is shown how fatigue increases over time, and a formula is given to determine the level of fatigue after a short rest.

The following is a model that describes worker fatigue after carrying out work in the study, adapted from Liu et al. [15].

$$F(t_i) = F(t_{i-1}) + (1 - F(t_{i-1}))(1 - e^{-\lambda(\Delta t)}) \quad (1)$$

To describe the fatigue after rest, the following equation is used.

$$F(t_i) = F(t_{i-1})e^{-\mu(\Delta t)} \quad (2)$$

In Equation (1) $F(t_i)$ represents the level of accumulated fatigue at time t_i, while λ denotes the rate at which fatigue builds up in the worker. A fatigue value of F=0 indicates no fatigue, whereas *F=1* represents the maximum fatigue level. Similarly, in Equation (2), $F(t_i)$ is the rest accumulation but it refers to the remaining fatigue after a rest period of duration Δt, with μ representing the recovery rate. To reformulate Equation (2) for the purpose of calculating the required rest duration to reach a certain fatigue level, the equation can be rearranged accordingly and become equation (3) as follows:

$$\Delta t = \frac{ln\left(F(t_{i-1})/F(t_i)\right)}{\mu} \quad (3)$$

Worker fatigue increases processing time, leading to longer job completion. So, there will be some modified processing time using the following equation.

$$T'_{jm} = T_{jm}\left(1 + \delta\lambda\left(ln(1 + F(t_{i-1}))\right)\right) \quad (4)$$

Equation (4) represents the modified processing time, and δ is a parameter that describes the magnitude of the effect of fatigue on processing time. The calculation of δ can be done using equation (5) below.

$$\delta = \frac{\%influenced\ on\ job}{\lambda ln(2)} \quad (5)$$

% Influenced on job means how much the job influences the worker when fatigue hits; it can also be interpreted as how demanding the work is. Liu et al. [15] stated it in three categories, namely the job has no influence at all (0% influenced on job), making δ=0 and the process time does not change, moderately influenced (30%), and severely influenced (70%)

When a worker is assigned to complete a job, it is necessary to first determine the actual process time, then after completing the job, the fatigue experienced by the worker will be recalculated using and if the fatigue level exceeds the specified limit based on Saha et al. [21] that if the level of operational load of workers exceeds 33% of maximum oxygen absorption, workers will not be able to continue working due to fatigue and will need to rest. The values of fatigue rate (λ) and recovery rate (μ) are taken from the results of a pilot study by Jaber et al. [17], which obtained results that were divided into three, namely slow, moderate, and fast r lambda, tthey're0.01 0.03, and 0.05, while for the μ are 0.03, 0.05, and 0.07

3.2. Mathematical Model

Subsequently, the mathematical model was extended by incorporating a worker fatigue mechanism adapted from Liu et al. [15], and fatigue parameters from Jaber et al. [17], which were integrated with NIPMSP. In line with recent exact and hybrid formulations for unrelated parallel machines that incorporate availability/eligibility constraints and controllable processing times [7- 8], An MINLP model is formulated to endogenize fatigue-driven processing-time deterioration. The following are the sets of indices, parameters, and decision variables used in the model development to determine the objective function value, as detailed in Table 1.

Table 1. Notations and Sets, Parameters, and Decision Variables

Notations and set	Description
M	Set of machines and workers, indexed by ($i \in \{0, 1, 2, \ldots, m\}$)
J	Set of jobs, indexed by ($j, k \in \{0,1,2, \ldots, n\}$).
Parameters	**Description**
P_{ij}	Processing time required to complete job j on machine

In Press

$F_{initial}$	Initial fatigue level of a worker before processing the first assigned job.	
F_{max}	Maximum fatigue level achieved before being required to take a break	
$F_{after_rest_target}$	Target fatigue level after taking a break	
λ	Fatigue rate parameter of the worker	
μ	Recovery rate parameter of the worker	
δ	Influenced factor parameter	
F_{abs_max}	The maximum fatigue value that can be achieved	1
N_{fs}	Number of breakpoints for linearization	$ln(1+F(t_{i-1}))$
N_y	Number of breakpoints for linearization	$e^{-\lambda(\Delta t)}$
$ln(fs_k)$	Parameter value of ln(fs) for linearization	
$expY_k$	Parameter value of exp(Y) for linearization	
$BigM_{time}$	Time-based constraint	10000
$BigM_{fatigue}$	Fatigue-based constraint	1

Decision Variables	Description
x_{ij}	Binary variable, 1 if job *j* is done by worker *i*
S_{ij}	Start time of job *j* by worker i
y_{ikj}	Binary variable, 1 if job *k* is done before job *j* on worker *i*
$first_{ij}$	Binary variable, 1 if job *j* is the first job done on worker *i*
FS_{ij}	Fatigue level experienced by worker *i* at the start of job *j*
P'_{ij}	Actual processing time of job *j* on machine *i*
FE_{ij}	Fatigue level experienced by worker *i* at the end of job *j*
UR_{ij}	Binary variable, 1 if rest required
TR_{ij}	Duration of rest required by worker *i* after job j
FA_{ij}	Fatigue level experienced by worker *i* after rest (if any) taken after job *j*
C_{ij}	Completion time of job *j* on worker *i*(before adding rest)
EC_{ij}	Effective completion time of job *j* on worker *i* (after adding rest)

Auxiliary Variables	Description
$appx_lnFS_{ij}$	Variable used to approximate the value of $ln(1+F_{(ti)})$
Y_{varij}	Variable representing $\lambda(\Delta t)$
$appx_expY_{ij}$	Variable used to approximate the value of e^{-Y}
Q_{FS_expYij}	Variable representing the result of $FA_{ij} \times appx_expY_{ij}$
$rest_needed_{ij}$	Binary variable equal to 1 if the fatigue level after work exceeds the specified threshold and is evaluated after task *j* for worker *i*
FAY_{ikj}	Auxiliary variable used to facilitate the calculation of FS_{ij}

Based on the assumptions and parameters described above, the problem is formulated as the following mixed-integer nonlinear programming (MINLP) model.

$$MinC_{max} = max\left\{\sum_{j=1}^{n} P'_{ij}x_{ij}\right\}, \quad \forall i \in M \qquad \text{Minimize make span (before adding rest)} \tag{6}$$

$$MinC_{max} = max\left\{\sum_{j=1}^{n} P'_{ij}x_{ij} + TR_{ij}\right\}, \quad \forall i \in M \qquad \text{Minimize make span (after adding rest)} \tag{7}$$

There will be several groupings of constraints used in this model.

1. Job assignment constraints

$$\sum_{i=0}^{m} x_{ij} = 1 \qquad \forall j \in J \tag{8}$$

$$C_{max} \geq EC_{ij} \qquad \forall i \in M, j \in J \tag{9}$$

The constraints (10) and (11) ensure that each job must be allocated to exactly one machine and worker and define that the make span must be the completion time on all machines and workers.

2. Fatigue initiation constraints

$$\sum_{j=0}^{n} first_{ij} \leq 1 \qquad \forall i \in M \tag{10}$$

$$first_{ij} \leq x_{ij} \qquad \forall i \in M \tag{11}$$

$$x_{ij} = first_{ij} + \sum_{k=0,k\neq j}^{n} y_{ikj} \qquad \forall i \in M;\ j \in J \tag{12}$$

$$S_{ij} \leq BigM_{time}(1 - first_{ij}) \qquad \forall i \in M;\ j \in J \tag{13}$$

In Press

$$S_{ij} \geq EC_{ik} - BigM_{time}(1 - y_{ikj}) \quad \forall i \in M;\ k,j \in J \tag{14}$$

$$y_{ikj} \leq x_{ik} \quad \forall i \in M;\ k,j \in J \tag{15}$$

$$y_{ikj} \leq x_{ij} \quad \forall i \in M;\ k,j \in J \tag{16}$$

$$y_{ijj} = 0 \quad \forall i \in M;\ j \in J \tag{17}$$

$$FAY_{ikj} \leq F_{abs_max} y_{ikj} \quad \forall i \in M;\ k,j \in J \tag{18}$$

$$FAY_{ikj} \leq FA_{ik} \quad \forall i \in M;\ k,j \in J \tag{19}$$

$$FAY_{\ ikj} \leq FA_{ik} - F_{abs_max}(1 - y_{ikj}) \quad \forall i \in M;\ k,j \in J \tag{20}$$

$$FAY_{ikj} \geq 0 \quad \forall i \in M;\ k,j \in J \tag{21}$$

$$FS_{ij} = F_{initial}\, first_{ij} + \sum_{k=0,k\neq j}^{n} FAY_{ikj} \quad \forall i \in M;\ j \in J \tag{22}$$

$$FS_{ij} \leq F_{abs_max} x_{ij} \quad \forall i \in M;\ j \in J \tag{23}$$

The constraints (12) and (13) ensure that there is only one job that can be the first scheduled job, and if a job is the first job for machine *i*, it must be allocated to machine and worker *i*. Constraint (14) was then designed so that if job *j* is allocated to machine and worker *i*, the job can be the first job or have another job as a predecessor.

In the constraints (15) and (16), if a job becomes the first scheduled job, the start time must be 0, and if not, then the start time must be greater than or equal to the effective completion time of the previous job. Constraints (17) and (18) mean that, if, $y_{ikj} = 1$ indicating that job *j* is processed immediately after job *k*, then $x_{ik} = 1$ and $x_{ij} = 1$. Consequently, both jobs *j* and *k* must be assigned to the same machine worker pair. Constraint (19) ensures that a job cannot be preceded by itself.

The following constraints define the fatigue level of worker *i* at the start of job *j*. Constraints (20) to (23) will ensure that the value of fatigue at start is the same as fatigue initial (0) if job *j* is the first scheduled job; otherwise, the value is the fatigue after the rest of the previous job.

3. Actual processing time constraints

$$\omega_{ijk}^{fs} \geq 0 \quad \forall i \in M, j \in J, k \in \{1, 2, \ldots, N_{fs} - 1\} \tag{24}$$

$$\sum_{k=0}^{N_{fs}-1} \omega_{ijk}^{fs} = x_{ij} \quad \forall i \in M, j \in J \tag{25}$$

$$FS_{ij} = \sum_{k=0}^{N_{fs}-1} \omega_{ijk}^{fs} fs_k \quad \forall i \in M, j \in J \tag{26}$$

$$appx_lnFS_{ij} = \sum_{k=0}^{N_{fs}-1} \omega_{ijk}^{fs} ln(fs_k) \quad \forall i \in M, j \in J \tag{27}$$

$$P'_{ij} \geq \left(P_{ij} + P_{ij}\, \delta\lambda\, appx_lnFS_{ij}\right) - BigM_{time}(1 - x_{ij}) \quad \forall i \in M, j \in J \tag{28}$$

$$P'_{ij} \leq \left(P_{ij} + P_{ij}\, \delta\, \lambda appx_lnFS_{ij}\right) - BigM_{time}(1 - x_{ij}) \quad \forall i \in M, j \in J \tag{29}$$

In carrying out work, processing time will change according to the fatigue felt by workers in previous work. Constraints (24) to (27) will be some sort of approximation of linearized form from the reference journal to calculate the actual processing time. Constraints (28) and (29) define the actual processing time by incorporating the effects of worker fatigue.

4. Completion time and effective completion time calculations constraints

$$C_{ij} = S_{ij} + \sum_{i=0}^{m} P'_{ij} \quad \forall j \in J \tag{30}$$

$$EC_{ij} = C_{ij} + \sum_{i=0}^{m} TR_{ij} \quad \forall j \in J \tag{31}$$

Constraints (30) and (31) ensure that the completion time of a job is calculated from the start time plus the actual processing time before any breaks, and then for the effective completion time, it is the sum of the completion time and the break time.

5. Machine and workers overlap prevention constraints

$$S_{ij} \geq EC_{ik} - BigM_{time}(1 - y_{ikj}) - BigM_{time}(2 - x_{ik} - x_{ij}) \quad \forall j,k \in J;\ i \in M \tag{32}$$

$$S_{ik} \leq EC_{ij} - BigM_{time}(1 - y_{ikj}) - BigM_{time}(2 - x_{ik} - x_{ij}) \quad \forall j,k \in J;\ i \in M \tag{33}$$

In this section, constraints (32) and (33) will prevent those two jobs from being done on the same machine and by the same worker at the same time

6. Fatigue after job constraints

$$Y_{varij} = \lambda P'_{ij} \quad \forall j \in J;\ i \in M \tag{34}$$

$$\omega_{ijk}^{expY} \geq 0 \quad \forall j \in J;\ i \in M \tag{35}$$

$$\sum_{k=0}^{N_{fs}-1} \omega_{ijk}^{expY} = x_{ij} \quad \forall j \in J;\ i \in M \tag{36}$$

$$Y_{varij} = \sum_{k=0}^{N_{fs}-1} \omega_{ijk}^{expY} expY_k \quad \forall j \in J;\ i \in M \tag{37}$$

$$appx_expY_{ij} = \sum_{k=0}^{N_y-1} \omega_{ijk}^{expY} expY_k \quad \forall j \in J;\ i \in M \tag{38}$$

$$Q_{FS_expYij} = FS_{ij}\ appx_expY_{ij} \quad \forall j \in J;\ i \in M \tag{39}$$

$$FE_{ij} \geq \left(1 - appx_e xpY_{ij} + Q_{Fs_expYij}\right) - BigM_{fatigue}(1 - x_{ij}) \quad \forall j \in J;\ i \in M \tag{40}$$

$$FE_{ij} \leq \left(1 - appx_e xpY_{ij} + Q_{Fs_expYij}\right) - BigM_{fatigue}(1 - x_{ij}) \quad \forall j \in J;\ i \in M \tag{41}$$

$$FE_{ij} \leq F_{abs_max} x_{ij} \quad \forall i \in M;\ j \in J \tag{42}$$

$$FE_{ij} \geq 0 \quad \forall i \in M;\ j \in J \tag{43}$$

In press

Similar to section 3, an approximation for linearization is used. Constraints (34) through (39) are a linearization approximation from the reference journal to calculate fatigue after a job. After those constraints (40) and (41), the formula to calculate fatigue after a job is given, then constraints (42) and (43) will ensure that it does not appear when job j is not allocated on the machine and worker i, and must not have a negative value.

7. Rest decision constraints

$$FE_{ij} \geq (F_{max} + \varepsilon)UR_{ij} - BigM_{fatigue}(1 - UR_{ij}) \quad \forall j \in J;\ i \in M \quad (44)$$

$$FE_{ij} \leq F_{max} + BigM_{fatigue}UR_{ij} \quad \forall j \in J;\ i \in M \quad (45)$$

$$UR_{ij} \leq x_{ij} \quad \forall i \in M;\ j,k \in J \quad (46)$$

Constraints (44) to (46) will ensure the decision of rest. if the fatigue value found is greater than or equal to the threshold value, the UR_{ij} value will be 1.

8. Rest duration constraints

$$FE_{ij} \geq (F_{after_rest} + \varepsilon)rest_needed_{ij} - BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (47)$$

$$FE_{ij} \leq F_{after_rest} - BigM_{fatigue}rest_needed_{ij} \quad \forall j \in J;\ i \in M \quad (48)$$

$$\mu TR_{ij} \geq (ln(FE_{ij}) - ln(FE_{after_rest})) - BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (49)$$

$$\mu TR_{ij} \leq (ln(FE_{ij}) - ln(FE_{after_rest})) + BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (50)$$

$$TR_{ij} \geq 0 \quad \forall i \in M;\ k,j \in J \quad (51)$$

Constraints (47) and (48) will ensure that the rest time will be valuable if $UR_{ij} = 1$ and the fatigue after work is greater thanF_{after_rest}. Constraints (49) – (51) determine the rest time required for worker recovery.

9. Fatigue calculations after rest (if any) constraints

$$FA_{ij} \leq F_{after_rest} + BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (52)$$

$$FA_{ij} \geq F_{after_rest} - BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (53)$$

$$FA_{ij} \leq FE_{ij} + BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (54)$$

$$FA_{ij} \geq FE_{ij} - BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (55)$$

$$FA_{ij} \geq 0 \quad \forall j \in J;\ i \in M \quad (56)$$

$$FA_{ij} \leq F_{after_rest} + BigM_{fatigue}(1 - rest_needed_{ij}) \quad \forall j \in J;\ i \in M \quad (57)$$

All the constraints in this section will specify that if $UR_{ij} = 1$, Then the fatigue after rest will have a value of F_{after_rest}. If $UR_{ij} = 0$, Then the value will be the same as fatigue after work. Constraint (52) ensures that the resulting value remains non-negative

4. Experiment and Results

This section presents the computational experiments conducted to evaluate the performance of the proposed mathematical model. The experiments include model verification, validation through manual calculations, numerical tests on various problem sizes, sensitivity analysis for fatigue-related parameters, and managerial insights derived from the results.

To test the developed model, hypothetical data were used, based on processing time matrices commonly adopted in previous NIPMSP studies. Each job $j \in J$ is assigned a base processing time P_{ij} for each machine and worker $i \in M$, forming a non-identical matrix. The experiment was conducted on several problem instances varying in size, including 2 to 3 machines and workers, and 6 to 20 jobs. Fatigue-related parameters were then set based on physiological studies and prior research [15]-[17] with the following configurations in Table 2.

4.1. Validation and verification

The model was implemented using Python and solved using the Gurobi Optimizer. Verification was performed by ensuring the model ran without infeasibilities or logic errors for all test instances. To validate the accuracy of the model, a small instance with 2 machines and 6 jobs was tested manually. An

In press

example of the processing time matrix and fatigue parameters is presented in Table 3.

Table 2. Parameter Values

Parameter	Value
Fatigue Rate (λ)	0.01, 0.03, 0.05
Recovery Rate (μ)	0.03, 0.05, 0.07
% Influenced (δ)	0%, 30%, 70%

Table 3. Processing Time Dataset

Job	Processing Time	
	Machine and Worker 1	Machine and Worker 2
Job 1	13	10
Job 2	14	12
Job 3	21	27
Job 4	9	10
Job 5	11	12
Job 6	15	13

The computational process was conducted for two scenarios: one without considering worker fatigue and one with considering fatigue. In the visualization of the scheduling results, indexing begins at 0; thus, index 0 corresponds to job 1, index 1 to job 2, and so forth. In the model that excludes worker fatigue, the resulting make span is 41. Under this scenario, Machine 1 is assigned jobs 4, 3, and 2, while Machine 2 handles jobs 5, 1, and 0. The scheduling output for the non-fatigue model is illustrated in Figure 1. However, this schedule is considered less realistic, as it does not reflect the effects of worker fatigue on processing time and job assignment.

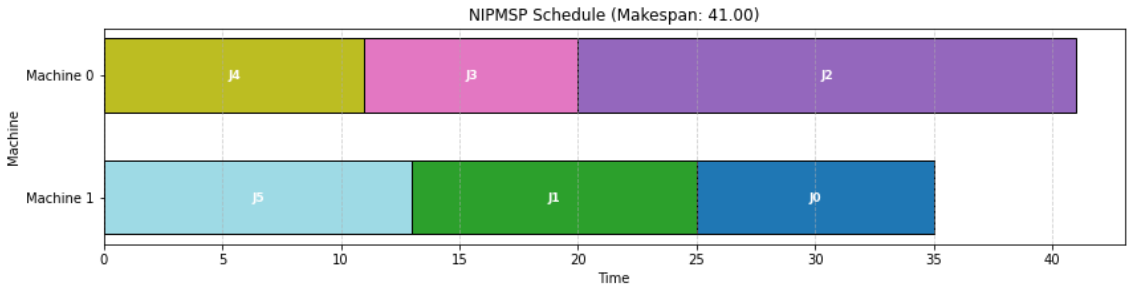

Figure 1. Optimal schedule obtained under the baseline scenario without worker fatigue.

The second computational experiment incorporates the consideration of worker fatigue, resulting in different scheduling outcomes. These differences are reflected in the objective function values presented in Table 4. and the decision variables are detailed in Table 5. The results indicate that the model functions correctly, producing a feasible and error-free solution. Additionally, the scheduling output is visualized to illustrate the computation results, as shown in Figure 2.

The computational results presented in Table 4. indicate a make span value of 76.63 when rest time at the end of the work period is considered for the first worker. This solution yields a 0% optimality gap, confirming that the model successfully identified an optimal solution. A clear difference emerges between the fatigue-aware and non-fatigue models. Without considering fatigue, the make span is 41. When fatigue is included, the make span increases to 76.63 with end-of-shift rest time, and to 44.75 when rest at the end is excluded (based on the completion time of Job 4 on worker 1). This increase is attributed to the extended processing times caused by accumulated fatigue, which impacts job completion. Comparing the results, the make span increases by approximately 9.41% from 41 to 44.75 when fatigue is considered without end-of-shift rest. These results serve as the foundation for the manual calculation phase that follows.

Table 4. Objective Function of Model with Fatigue

Instance	Job	Machine and worker ID	Computational time	Gap (%)	Make span
XYZ	6	2	61.29s	0%	76.63

Table 5. Decision Variable of the Model with Fatigue

Job	Machine & Worker ID	Base Time	Fatigue Start	Actual Time	Fatigue end	Rest Duration	Fatigue after rest	Completion time
1	1	12	0	12	0.30	No	0.30	12
5	1	13	0.30	14.49	0.55	No	0.55	26.49
0	1	10	0.55	11.89	0.68	No	0.68	38.38
2	0	21	0	21	0.47	No	0.47	21
4	0	11	0.47	12.83	0.64	No	0.64	33.83
3	0	9	0.64	10.92	0.74	31.89	0.15	44.75
				Rest				76.63

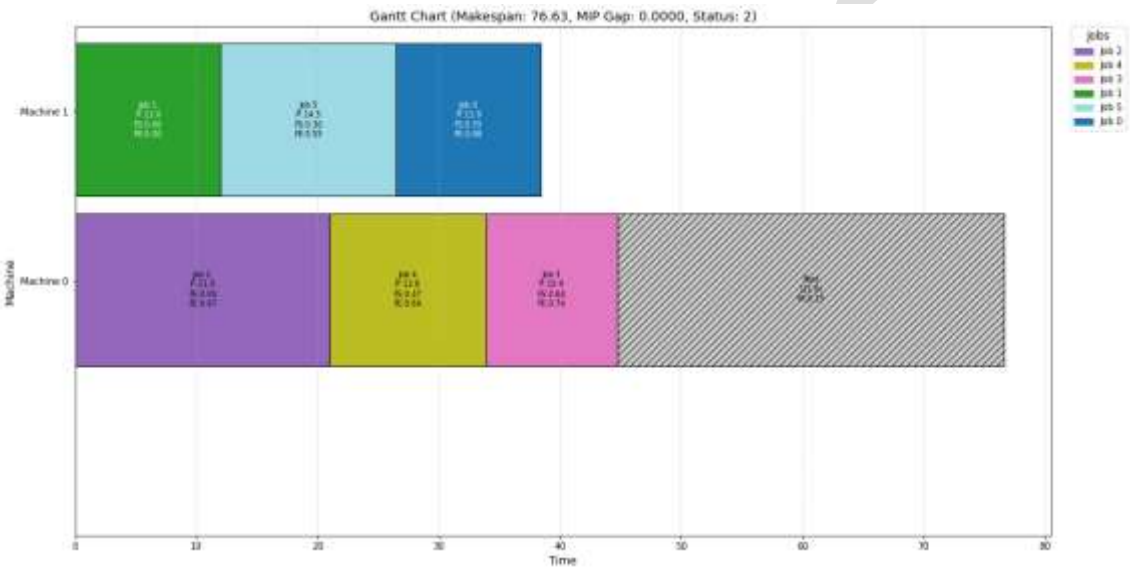

Figure 2. Optimal schedule considering worker fatigue effects.

After obtaining the correct computational results from the Gurobi optimizer program, manual calculations are conducted to validate the accuracy of the decision variable outputs, particularly those related to fatigue. These manual validations utilize the original fatigue accumulation and recovery formulas proposed by Liu et al. [15], along with parameter values sourced from de Souza Ferreira et al. [22]. The purpose is to confirm that the approximation model accurately reflects the original formula. The validation process involves step-by-step tracing of the job sequence generated by Gurobi to recalculate fatigue levels manually and ensure consistency with the model's output. The results are presented in Table 6.

After computing the worker fatigue values using the original model, the results were found to be consistent with those obtained from the Gurobi model, matching up to two decimal places. Any minor discrepancies are likely due to rounding errors introduced during the approximation process. This close agreement demonstrates that the mathematical model effectively captures fatigue dynamics with a high degree of accuracy. Additionally, the make span calculated manually by summing the individual job completion times matches the objective function value produced by the computational model. For a comparison of the values obtained between calculations using the model and manual calculations, see Table 7.

Table 6. Manual Fatigue Calculation

Job	Base Time	F after job	Rest	F after rest	Processing time next job
			Machine/Worker 1		
2	21	0.47	No	0.47	12.83
4	11	0.64	No	0.64	10.92

Job	Base Time	F after job	Rest	F after rest	Processing time next job
3	9	0.74	31.89	0.15	-
Machine/Worker 2					
1	12	0.30	No	0.30	14.49
5	13	0.55	No	0.55	11.89
0	10	0.68	No	0.68	-

Table 7. Comparison of Gurobi vs. Manual Fatigue Calculation

Job	Machine & Worker ID	Fatigue Start		Fatigue End	
		Model	Manual	Model	Manual
2	0	0	0	0.4674	0.4674
4	0	0.4674	0.4674	0.6375	0.6375
3	0	0.6375	0.6375	0.7387	0.7387
1	1	0	0	0.3023	0.3023
5	1	0.3023	0.3023	0.5482	0.5482
0	1	0.5482	0.5482	0.6838	0.6837

4.2. Numerical experiments

After verification and validation, with the values obtained not violating the established constraint function, the model then performs a series of further tests to complete the numerical study case with more varied machine configurations. At this stage, testing was carried out on 8 different datasets. The result can be seen in Table 8.

Based on the results of the numerical study, all problem instances yielded feasible solutions. The total computation time is influenced by several factors, including the complexity of the problem instance defined by the number of machines, workers, jobs, and the variability in processing times and the number of breakpoints used in approximation. As the problem size increases, the time required to find a feasible or optimal solution also increases.

Several instances reached the 4-hour computational time limit without achieving proven optimality, highlighting the computational complexity introduced by fatigue accumulation and recovery modelling. Future studies should investigate decomposition methods, heuristics, or metaheuristics to improve scalability for larger industrial-sized instances

4.3. Sensitivity analysis

A sensitivity analysis is conducted to evaluate the impact of parameter variations on the objective function values. In this study, the parameters that will be used for sensitivity analysis are fatigue rate (λ), recovery rate (μ), and %influenced (δ).

Table 8. Results of Numerical Study

Instance	No. of Machine & Worker	Job	Computational Time	Optimality Gap (%)	Make span
1	2	6	180.28 s	0	80.69
2	2	12	14400.18 s	69.44%	172.06
3	2	12	14400.29 s	64.75%	179.51
4	3	16	14400.38 s	83.86%	167.30
5	2	16	14400.37 s	82.49%	262.67
6	3	20	14400.59 s	84.02%	211.95
7	3	20	14400.63 s	86.78%	227.01

Table 9. Fatigue (λ) Rate Value Range

Level	Description	Value of λ	
		λ_1	λ_2
1	Slow to moderate	0.01	0.03
2	Slow to fast	0.01	0.05
3	Moderate to fast	0.03	0.05

Table 9. presents the range of λ values used in the sensitivity analysis to examine how changes in the worker fatigue rate affect the model's objective function. In this analysis, only the fatigue rate (λ) is varied, while other parameters, namely the recovery

In press

rate (μ) and the percentage of jobs affected by fatigue (%influenced) are held constant at moderate levels, as recommended in the study by Jaber et al. [17].

Table 10. Model Results with Fatigue Rate (λ) Parameter Changes

Level	Objective Value 1	Objective Value 2	Computational Time 1	Computational Time 2
1	42.79	76.63	8.48s	61.29s
2	42.79	77.03	8.48s	73.79
3	76.63	77.03	61.29s	73.79

Note:
Objective Value 1 = make span computed from completion times before any end-of-shift rest;
Objective Value 2 = make span computed from effective completion times after including any required end-of-shift rest.

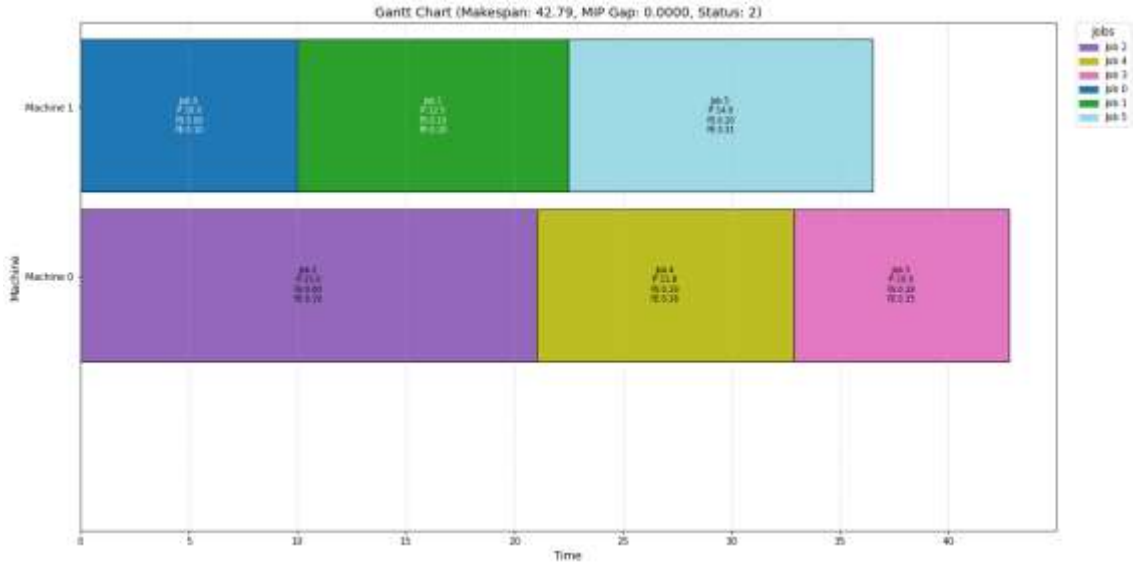

Figure 3. Optimal schedule obtained with a fatigue rate of λ = 0.01

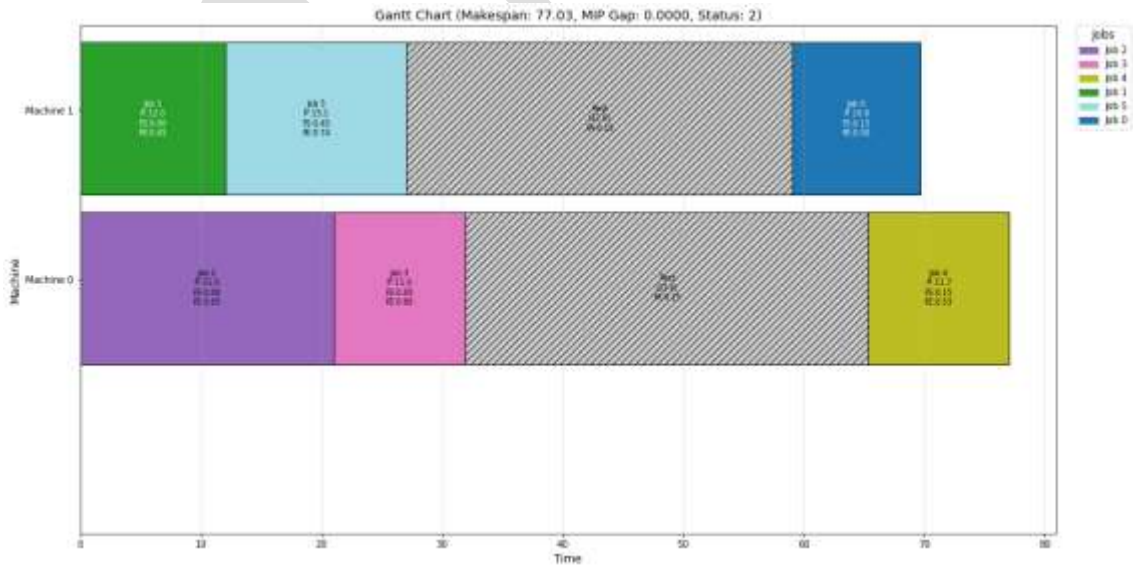

Figure 4. Optimal schedule obtained with a fatigue rate of λ = 0.05.

Based on Table 10, which presents the differences in objective function values, it can be concluded that higher worker fatigue rates lead to increased objective function values. This outcome is attributed to the quicker accumulation of fatigue, which causes workers to reach the maximum fatigue threshold

sooner and necessitate more frequent rest periods. The significant gap observed between λ=0.01 and the higher fatigue rates is due to the slower fatigue accumulation at λ=0.01, which in turn allows the schedule to be completed without requiring any rest. Figures 3, 4, and 5 present the scheduling results for fatigue rates of λ=0.03, λ=0.01, and λ=0.05, respectively.

The influence of the recovery rate parameter (μ) on the objective function values is investigated using the parameter settings presented in Table 11. And the same as before, the other parameters will be set to moderate points.

Table 11. Recovery Rate (μ) Value Range

Level	Description	Value μ	
		μ_1	μ_2
1	Slow to moderate	0.03	0.05
2	Slow to fast	0.03	0.07
3	Moderate to fast	0.05	0.07

Table 12. Model Result with Recovery Rate (μ) Parameter Changes

Level	Objective Value 1	Objective Value 2	Computational Time 1	Computational Time 2
1	97.89	76.63	120.73s	61.29s
2	97.89	67.52	120.73s	89.54s
3	76.63	67.52	61.29s	89.54s

Note:
Objective Value 1 = make span (before end-of-shift rest);
Objective Value 2 = make span (after end-of-shift rest, if required).

Based on Table 12, it can be seen that the greater the recovery rate value, the lower the objective value produced. This is related to the shorter rest time required. Figure 2 is a visualization of μ= 0.05, Figure 5 is a visualization of μ=0.03, and Figure 6 is a visualization of μ = 0.07.

The last parameter to be tested is the percentage of how much fatigue influences the increase in process time. The parameter values can be seen in Table 13. and the other parameters are set to moderate values as before.

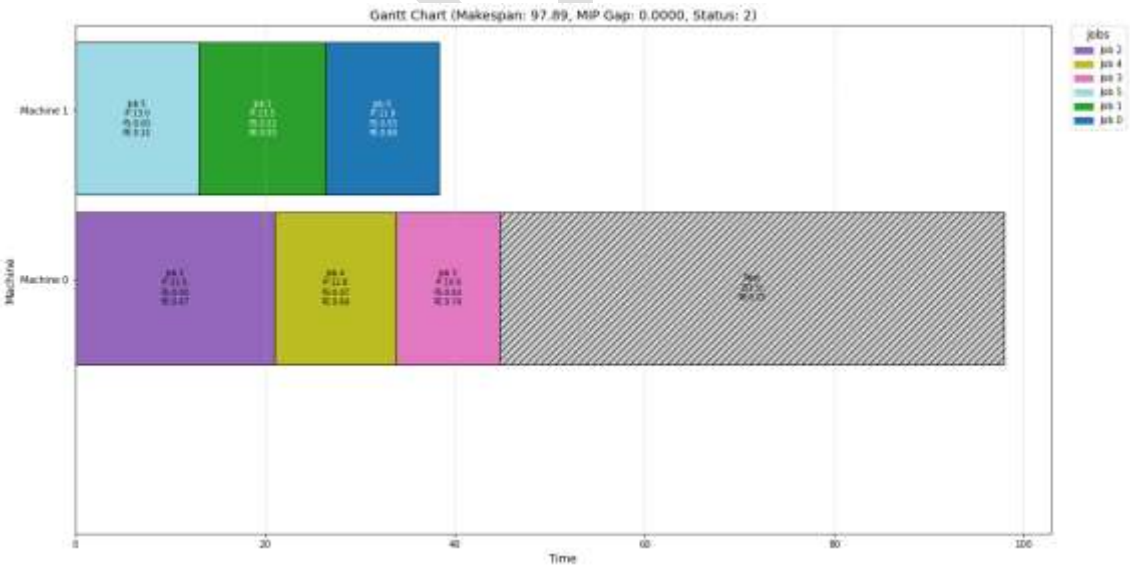

Figure 5. Optimal schedule obtained with a recovery rate of μ = 0.03.

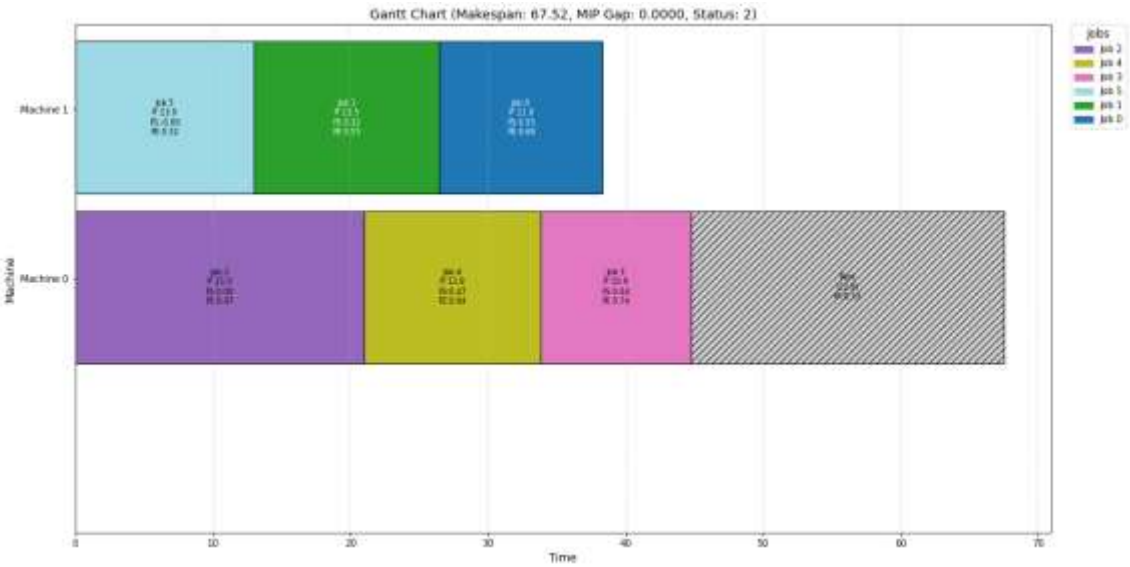

Figure 6. Optimal schedule obtained with a recovery rate of μ = 0.07.

Table 13. % Influenced Value Range

Level	Description	Value % *influenced*	
		$influenced_1$	$influenced_2$
1	No influence to moderate	0%	30%
2	No influence to severe	0%	70%
3	Moderate to severe	30%	70%

Table 14. Model Result with %influenced Parameter Changes

Level	Objective Value 1	Objective Value 2	Computational Time 1	Computational Time 2
1	72.03	76.63	57.86s	61.29s
2	72.03	81.57	57.86s	98.43s
3	76.63	81.57	61.29s	98.43s

Note:
Objective Value 1 = make span (before end-of-shift rest);
Objective Value 2 = make span (after end-of-shift rest, if required).

Table 14. presents the results of the sensitivity analysis for the %influenced parameter, showing a clear trend of increasing objective function values as the percentage of fatigue-affected jobs rises. This pattern reflects the impact of a higher proportion of jobs contributing to processing time deterioration; as more jobs are influenced by fatigue, overall processing durations increase, thereby raising the make span. Visual representations of the scheduling outcomes are provided in Figures 3, 8, and 9, corresponding to %influenced values of 0.3, 0.0, and 0.7, respectively. These trends are consistent with recent findings that processing-time controllability, setup/availability effects, and worker-related constraints can materially shift the Pareto trade-offs and elevate $C_{\max}$ in unrelated parallel-machine environments [7]; [8]; [10]; [9].

4.4. Managerial Insight

The integration of worker fatigue into the Non-Identical Parallel Machine Scheduling Problem provides valuable managerial insights for industries that rely heavily on human-operated machinery. Traditional scheduling approaches often neglect the physical and cognitive limitations of workers, assuming that processing times remain constant throughout the production process. However, this study demonstrates that fatigue significantly alters processing efficiency over time, especially in high-volume and repetitive task environments. By explicitly modeling fatigue accumulation and recovery, managers can obtain more realistic estimations of production timelines, which are crucial for accurate delivery forecasting, inventory planning, and resource allocation. This ensures that production plans are not only efficient on paper but also feasible in practice, aligning operational targets with human performance capacity. Furthermore, aligning fatigue-aware scheduling with sustainability-oriented decision support echoes

In press

recent hybrid-flow-shop studies on carbon-aware planning and model integration into operational analytics [22].

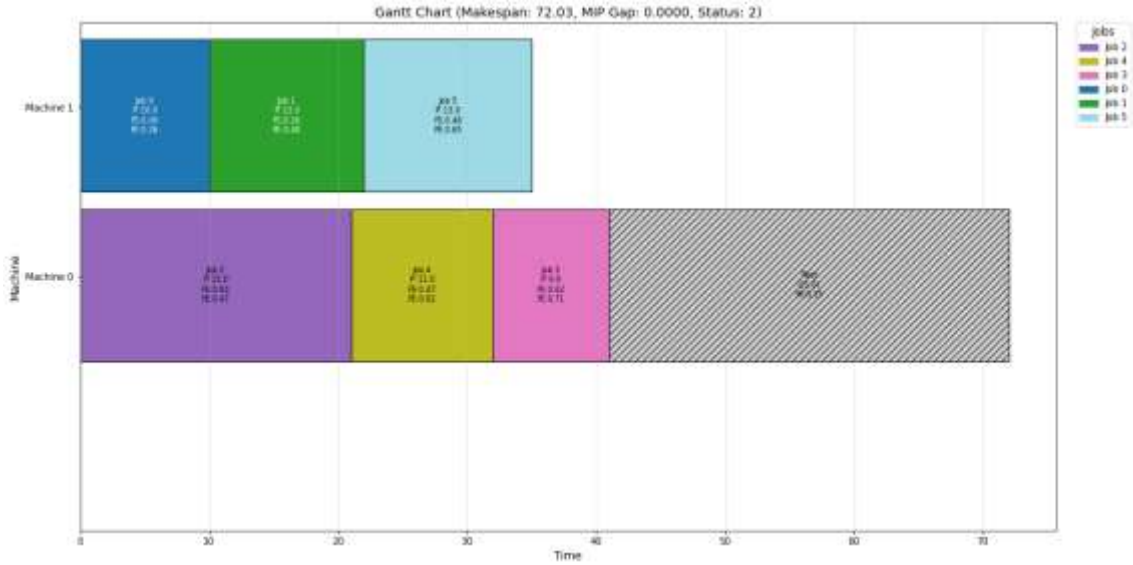

Figure 7. Optimal schedule when no jobs are affected by worker fatigue (% influenced = 0).

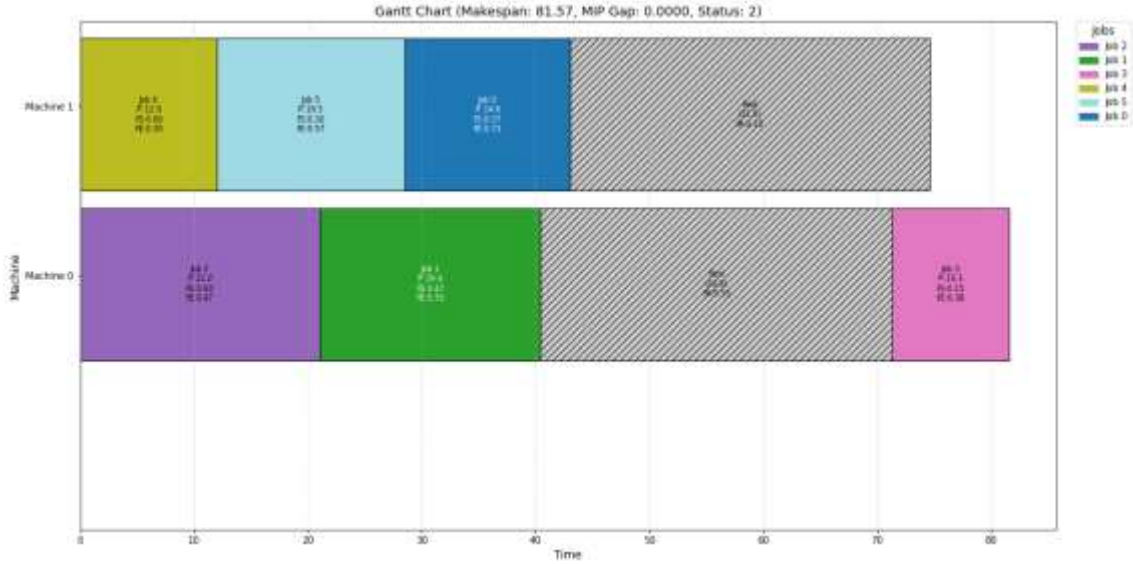

Figure 8. Optimal schedule when 70% of jobs are affected by worker fatigue (% influenced = 0.7).

In this study, the parameter values and job data were obtained for the purpose of building and testing the model. Factors such as the type of work, work environment, tools used, to the demographics and physical conditions of workers will certainly produce different values of λ and μ. If a company or policymaker wants to implement a model, they cannot simply use parameters from literature or previous research. They need to calibrate the existing parameters.

Before executing the proposed fatigue-aware scheduling model, several preparatory steps must be taken to ensure the model is correctly initialized and capable of generating valid, interpretable results. These steps are critical, particularly due to the presence of nonlinear fatigue dynamics and the need for linearization, which requires additional parameter tuning and data preprocessing. Each step is intended to prepare both the structural components of the model and the contextual assumptions required to reflect realistic production conditions.

The first essential step is data preparation, which involves constructing the base processing time matrix that defines how long each job takes when processed on each machine, without fatigue effects. This data is ideally taken when workers are in prime condition (beginning of the shift) to avoid fatigue bias. This matrix must reflect the non-identical nature of machines and workers. In addition, the

model requires specific fatigue-related parameters: the fatigue rate λ, recovery rate μ, and the percentage of jobs influenced by fatigue (%influenced). These values can be determined by conducting a pilot study, which will observe a small group of workers on certain tasks to measure the decline in performance over time. Each job must also be assigned an influence level, which determines whether and how much the job contributes to fatigue accumulation.

After data collection, the next step is implementation and optimization by entering processing time data and parameters into the model. The final step is strategic decision-making by policymakers. This decision-making can be in the form of sensitivity analysis for policies, for example, by changing the λ parameter to determine the impact of a program aimed at reducing fatigue, policies for worker rotation, and company production capacity planning. Through the results obtained from this study, the mathematical model that has been designed is expected to be a valuable contribution to practitioners and academics in the field of work scheduling.

5. Conclusion

Based on the research conducted, several conclusions can be drawn. The design of a mathematical model to determine the optimal make span value of the non-identical parallel machine scheduling (NIPMSP) case has been carried out, and the model has gone through the verification and validation stages. It was found that the model can produce feasible and ideal solutions. The model has gone through a series of numerical study tests with machine and worker configurations between 2-3 with 6-20 jobs. Gurobi Optimizer is used to run the model and produce an optimal solution with a cut-off time of 4 hours, and these models can effectively minimize the make span. The higher the worker's fatigue rate, the higher the make span produced. This is because workers will get tired quickly and need to rest immediately. The higher the worker's recovery rate, the faster the make span produced. This is because workers will need less time to rest and then be able to return to work. The higher the %-influenced value, the greater the make span produced. This is because changes in processing time will be greater, along with the increase in % influenced, which also makes the completion time longer.

Future research should (i) improve computational tractability via metaheuristics, (ii) relax the one-to-one assumption by allowing non-dedicated workers, (iii) model worker heterogeneity in fatigue and recovery parameters, and (iv) integrate the model into a real-time decision support system. These extensions would enable more realistic, data-driven scheduling while minimizing delays and supporting sustainable workforce management. Collectively, they position this study as a foundation for human-centered scheduling and practical deployment in labor-intensive industries.

Declarations of Interest Statement

Conflict of interest: The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the data and the paper are free of plagiarism.

Funding: No funding was obtained for this study.

Ethics approval and consent to participate: Not applicable.

Availability of data and material: The data that support the findings of this study are available from the corresponding author upon reasonable request.

Consent for publication We hereby provide consent for the publication of the manuscript detailed above, including any accompanying images or data contained within the manuscript.

Acknowledgements We would like to express our sincere thanks to all members of the production process and the system laboratory UGM who assisted during the data collection.

References

[1] A. Cardon, T. Galinho, and J.-P. Vacher, "Genetic algorithms using multi-objectives in a multi-agent system," *Robotics and Autonomous Systems*, vol. 33, no. 2–3, pp. 179–190, Nov. 2000, doi: 10.1016/S0921-8890(00)00088-9.

[2] C. Moon, M. Lee, Y. Seo, and Y. H. Lee, "Integrated machine tool selection and operation sequencing with capacity and precedence constraints using genetic algorithm," *Computers & Industrial Engineering*, vol. 43, no. 3, pp. 605–621, Sep. 2002, doi: 10.1016/S0360-8352(02)00129-8.

[3] S. Balin, "Non-identical parallel machine scheduling using genetic algorithm," *Expert Systems with Applications*, vol. 38, no. 6, pp. 6814–6821, Jun. 2011, doi: 10.1016/j.eswa.2010.12.064.

[4] L. S. Aaronson *et al.*, "Defining and Measuring Fatigue," *Image: the Journal of Nursing Scholarship*, vol. 31, no. 1, pp. 45–50, Mar. 1999, doi: 10.1111/j.1547-5069.1999.tb00420.x.

[5] S. E. Bechtold and D. L. Sumners, "Note—Optimal Work-Rest Scheduling with

In Press

Exponential Work-Rate Decay," *Management Science*, vol. 34, no. 4, pp. 547–552, Apr. 1988, doi: 10.1287/mnsc.34.4.547.

[6] S. Konz, "Work/rest: Part II – The scientific basis (knowledge base) for the guide," *International Journal of Industrial Ergonomics*, vol. 22, no. 1–2, pp. 73–99, Aug. 1998, doi: 10.1016/S0169-8141(97)00069-3.

[7] M. C. Santoro and L. Junqueira, "Unrelated parallel machine scheduling models with machine availability and eligibility constraints," *Computers & Industrial Engineering*, vol. 179, p. 109219, May 2023, doi: 10.1016/j.cie.2023.109219.

[8] S. Wang, R. Wu, F. Chu, and J. Yu, "Unrelated parallel machine scheduling problem with special controllable processing times and setups," *Computers & Operations Research*, vol. 148, p. 105990, Dec. 2022, doi: 10.1016/j.cor.2022.105990.

[9] L. Zhang, Q. Deng, R. Lin, G. Gong, and W. Han, "A combinatorial evolutionary algorithm for unrelated parallel machine scheduling problem with sequence and machine-dependent setup times, limited worker resources and learning effect," *Expert Systems with Applications*, vol. 175, p. 114843, Aug. 2021, doi: 10.1016/j.eswa.2021.114843.

[10] M. A. Noman, M. Alatefi, A. M. Al-Ahmari, and T. Ali, "Tabu Search Algorithm Based on Lower Bound and Exact Algorithm Solutions for Minimizing the Makespan in Non-Identical Parallel Machines Scheduling," *Mathematical Problems in Engineering*, vol. 2021, pp. 1–9, Dec. 2021, doi: 10.1155/2021/1856734.

[11] M. Afzalirad and J. Rezaeian, "Resource-constrained unrelated parallel machine scheduling problem with sequence dependent setup times, precedence constraints and machine eligibility restrictions," *Computers & Industrial Engineering*, vol. 98, pp. 40–52, Aug. 2016, doi: 10.1016/j.cie.2016.05.020.

[12] C.-Y. Cheng and L.-W. Huang, "Minimizing total earliness and tardiness through unrelated parallel machine scheduling using distributed release time control," *Journal of Manufacturing Systems*, vol. 42, pp. 1–10, Jan. 2017, doi: 10.1016/j.jmsy.2016.10.005.

[13] J. E. C. Arroyo and J. Y.-T. Leung, "Scheduling unrelated parallel batch processing machines with non-identical job sizes and unequal ready times," *Computers & Operations Research*, vol. 78, pp. 117–128, Feb. 2017, doi: 10.1016/j.cor.2016.08.015.

[14] M. Liu and X. Yang, "Bi-objective optimization for Scheduling and Multi-skilled Worker Assignments in the Hybrid Flow Shop," *IFAC-PapersOnLine*, vol. 52, no. 13, pp. 2128–2133, 2019, doi: 10.1016/j.ifacol.2019.11.520.

[15] 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, Apr. 2023, doi: 10.1016/j.rcim.2022.102478.

[16] I. E. Mouayni, G. Demesure, H. B.-E. Haouzi, P. Charpentier, and A. Siadat, "Jobs scheduling within Industry 4.0 with consideration of workers' fatigue and reliability using Greedy Randomized Adaptive Search Procedure," *IFAC-PapersOnLine*, vol. 52, no. 19, pp. 85–90, 2019, doi: 10.1016/j.ifacol.2019.12.114.

[17] M. Y. Jaber, Z. S. Givi, and W. P. Neumann, "Incorporating human fatigue and recovery into the learning–forgetting process," *Applied Mathematical Modelling*, vol. 37, no. 12–13, pp. 7287–7299, Jul. 2013, doi: 10.1016/j.apm.2013.02.028.

[18] A. Ferjani, A. Ammar, H. Pierreval, and S. Elkosantini, "A simulation-optimization based heuristic for the online assignment of multi-skilled workers subjected to fatigue in manufacturing systems," *Computers & Industrial Engineering*, vol. 112, pp. 663–674, Oct. 2017, doi: 10.1016/j.cie.2017.02.008.

[19] Z.-K. Li, N. Li, Q.-K. Pan, L. Gao, and W.-M. Li, "A parallel cooperative co-evolutionary algorithm for flexible job-shop scheduling with Workers' heterogeneity," *Expert Systems with Applications*, vol. 290, p. 128362, Sep. 2025, doi: 10.1016/j.eswa.2025.128362.

[20] X. Liu, X. Chen, Y. Zhao, X. Liu, C.-C. Wu, and W.-C. Lin, "A multi-objective flexible flow shop scheduling problem with an improved NSGA-II algorithm," *JIMO*, vol. 21, no. 5, pp. 4041–4062, 2025, doi: 10.3934/jimo. 2025042.

[21] P. N. Saha, S. R. Datta, P. K. Banerjee, and G. G. Narayane, "An acceptable workload for Indian workers," *Ergonomics*, vol. 22, no. 9, pp. 1059–1071, Sep. 1979, doi: 10.1080/00140137908924680.

[22] G. D. S. Ferreira, G. R. Mateus, and M. G. Ravetti, "Minimizing Carbon Emission in Hybrid Flow Shop Scheduling: A Comparative Analysis of Flow-based and Set Partitioning Formulations," *Procedia Computer Science*, vol. 232, pp. 2831–2840, 2024, doi: 10.1016/j.procs.2024.02.146.

In press