MULTI-OBJECTIVE OPTIMIZATION OF NON-IDENTICAL PARALLEL MACHINE SCHEDULING FOR MINIMIZING MAKESPAN, CARBON EMISSIONS, AND TARDINESS

Fairuzzaky Ramadhan[1]*, Achmad Pratama Rifai[1]

[1]Departemen Teknik Mesin dan Industri, Fakultas Teknik, Universitas Gadjah Mada
*Corresspondence : fairuzzaky@gmail.com

Abstract

Production scheduling in modern manufacturing must simultaneously address operational efficiency, environmental sustainability, and customer satisfaction. As industries face increasing pressure to reduce carbon emissions while maintaining competitiveness, traditional single-objective scheduling approaches prove insufficient. Non-identical parallel machine scheduling problem (NIPMSP), where machines possess heterogeneous processing capabilities and environmental impacts, represents a prevalent configuration in manufacturing facilities. However, existing research has not simultaneously optimized makespan, carbon emissions, and tardiness—three critical objectives reflecting productivity, sustainability, and service reliability.

In this study, we developed a comprehensive mathematical model with 18 constraints and proposed Multi-Objective Adaptive Large Neighborhood Search (MOALNS) incorporating five destroy operators, four repair operators, and adaptive weight mechanisms. Performance evaluation against NSGA-II across eight problem instances using five metrics (Hypervolume, Inverted Generational Distance, Diversification Metric, Spacing Metric, CPU Time) with rigorous statistical testing (Wilcoxon signed-rank test, Cohen's d) revealed that MOALNS significantly outperforms NSGA-II in solution diversity ($p < 0.05$ in 50% of instances, $d = 6.31$) and spacing uniformity (75% win rate, $d = 0.70$), with comparable Pareto front coverage. This superiority comes at computational cost (192% slower), acceptable for offline planning but not real-time applications. This research provides evidence-based algorithm selection guidelines for sustainable manufacturing, demonstrating that environmental objectives can be integrated without sacrificing operational efficiency or delivery reliability.

History:

Received: January 9, 2026
Accepted: February 26, 2026
First published online: February 28, 2026

Keywords:

Carbon Emission
Makespan
Multi-objective Optimization
Non-Identical Parallel Machine
Tardiness

1. Introduction

Production scheduling represents a critical component in manufacturing operations management, directly influencing operational efficiency, delivery reliability, and environmental sustainability. In contemporary manufacturing, scheduling decisions must balance multiple conflicting objectives including productivity maximization, cost minimization, and environmental impact reduction. Parallel machine scheduling problems are prevalent in real-world manufacturing, with non-identical parallel machine scheduling problem (NIPMSP) representing the most complex configuration due to machine-dependent processing characteristics affecting approximately 68% of manufacturing facilities (Zhang et al., 2023).

Early NIPMSP research focused on single-objective optimization, typically makespan minimization (Chu et al., 2010). Subsequent work addressed processing time variability using fuzzy set theory (Alcan & Başlıgil, 2012) and batch production with arbitrary release times(Zhang et al., 2023). Research objectives evolved from traditional efficiency metrics to energy consumption (Xiao et al., 2021)and, more recently, carbon emission reduction reflecting climate change mitigation priorities and sustainable manufacturing practices (Akbar & Irohara, 2018; Bok et al., 2024; Ferreira et al., 2024).These developments align with global efforts toward carbon neutrality following the Paris Climate Accord.

Multi-objective optimization emerged as researchers recognized that real-world scheduling requires simultaneous satisfaction of multiple criteria. Studies have investigated makespan with resource utilization (Yepes-Borrero et al., 2021) makespan with energy consumption (Xiao et al., 2021), and energy with tardiness (Asadpour et al., 2022). However, no prior study has simultaneously optimized makespan, carbon emissions, and total tardiness within NIPMSP framework—a critical gap given that manufacturing scenarios frequently require balancing operational efficiency, environmental compliance, and customer satisfaction.

Solution approaches have employed various metaheuristics including genetic algorithms (Alcan & Başlıgil, 2012), simulated annealing (Asadpour et al., 2022), brain storm optimization (Hou et al., 2023), and ant colony optimization (Zhang et al., 2023). Adaptive Large Neighborhood Search (ALNS), originally developed by Ropke and Pisinger (2006), has demonstrated effectiveness across optimization (Mao et al., 2024; Windras Mara et al., 2022). Multi-Objective ALNS (MOALNS), introduced by Rifai et al., 2016), extends ALNS to multi-objective contexts by maintaining non-dominated solution sets while adaptively selecting operators.

This research addresses three critical gaps: (1) no prior study simultaneously optimizes makespan, carbon emissions, and tardiness in NIPMSP; (2) MOALNS has not been applied to NIPMSP despite success in other domains; (3) existing multi-objective NIPMSP studies lack comprehensive statistical validation. We contribute: (1) a comprehensive MO-NIPMSP mathematical model with three objectives; (2) MOALNS algorithm with multiple adaptive mechanisms; (3) rigorous statistical validation using Wilcoxon tests and effect size analysis comparing MOALNS against NSGA-II.

The paper proceeds as follows: Section 2 presents the mathematical formulation and MOALNS algorithm; Section 3 reports computational results and statistical analysis; Section 4 concludes with findings, implications, and future research directions.

2. Methodology

2.1. Problem Description

The MO-NIPMSP involves scheduling n jobs on m non-identical parallel machines. Each job *j* must be processed by exactly one machine *i* without preemption. Machines diff er in processing time p_{ij} (capability/speed), carbon emission e_{ij} (energy consumption pattern), for each job *j*. Each job has due date dj for completion. The problem requires determining machine assignment and job sequencing to simultaneously minimize: (1) makespan (maximum completion time), (2) total carbon emissions, (3) total tardiness (sum of delays beyond due dates).

2.2. Mathematical Model

The mathematical model for MO-NIPMSP is formulated as a mixed-integer linear programming problem incorporating sets, parameters, decision variables, objective functions, and constraints. The complete formulation ensures feasibility of schedules while optimizing the three objectives simultaneously. The indices and notation sused in this study are expressed as follows.

J	Set of job J ∈ [1, 2, ..., n}
M	Set of machine M ∈ [1, 2, ..., m]
j,k	Indices for jobs
i	Index for machines
p_{ij}	Processing time of job *j* on machine *i* (time units)
e_{ij}	Carbon emission generated when processing job *j* on machine *i* (emission units)
Q	Due date of job *j* (time units)
d_j	Large constant (Big-Q) ensuring constraint activation
n	Total number of jobs
m	Total number of machines
x_{ij}	Binary variable, 1 if job *j* is assigned to machine *i*, 0 otherwise
y_{ijk}	Binary variable, 1 if job *k* is processed immediately after job *j* on machine *i*, 0 otherwise
z_{ijt}	Binary varialbe, 1 if job *j* is being processed on machine *i* during time period *t*, 0 otherwise
S_{ij}	Starting time of job *j* on machine *i* (time units)
C_{ij}	Completion time of job *j* on machine *i* (time units)
C_{ik}	Tardiness of job *j*, defined as max (0, C_{ij} - d_j) (time units)
T_j	Makespan, representing the maximum completion time across all machines (time units)

This model simultaneously optimizes three objectives: (1) minimizing makespan (C_{max}), which represents the maximum completion time across all machines to enhance throughput and production capacity utilization, (2) minimizing total carbon emissions (E_{tot}) generated during the production process to support environmental sustainability and corporate carbon neutrality targets, and (3) minimizing total tardiness (T_{tot}), which measures the delay in job completion relative to due dates to improve customer satisfaction and reduce late delivery penalties.

Objective functions:

$$Min\ C_{max} = \max_{i=1,\dots,m} \left\{ \sum_{j=1}^{n} p_{ij} x_{ij} \right\} \forall\ i \in M; j \in J \tag{1}$$

$$Min\ E_{tot} = \sum_{j=1}^{n} \sum_{i=1}^{m} x_{ij} e_{ij}\ \forall\ i \in M; j \in J \tag{2}$$

$$Min\ T_{tot} = \sum_{j=1}^{n} T_j\ \forall\ j \in J \tag{3}$$

Subject to 15 constraint categories (equations 4-18):

$$\sum_{i=1}^{m} x_{ij} = 1\ \forall\ j = J \tag{4}$$

$$\sum_{j=1}^{n} x_{ij} \leq 1\ \forall\ i \in M \tag{5}$$

$$\sum_{(j=0, j\neq k)}^{n} y_{ijk} = x_{ij} \forall\ i \in M, k \in J \tag{6}$$

$$\sum_{k=1}^{n} y_{i0k} = 1\ \forall\ i \in M \tag{7}$$

$$C_{ij} = S_{ij} + \sum_{i=1}^{m} p_{ij} x_{ij}\ \ \forall\ j \in J \tag{8}$$

$$C_0 = 0 \tag{9}$$

$$C_{ij} - S_{ik} + Q\left(\sum_{i=1}^{m} y_{ijk} - 1\right) + \sum_{i=1}^{m} \sum_{k=0}^{n} S_{ikj} y_{ikj}\ \leq 0\ \forall\ i, k \in J \tag{10}$$

$$\sum_{i=1}^{m} \sum_{t=0}^{t_{max}} z_{ijt} = f_j - t_j\ \forall\ \ j \in J \tag{11}$$

$$\sum_{t=0}^{t_{max}} Z_{ijt} \leq Q \cdot x_{ijk}\ \ \forall\ j, k \in J, i \in M \tag{12}$$

$$C_{ij} \geq T \sum_{t=0}^{t_{max}} z_{ijt} \quad \forall j,k \in J, i \in M \tag{13}$$

$$S_{ij} \leq T \sum_{i=1}^{m} Z_{ijt} + Q\left(1 - \sum_{i=1}^{m} Z_{ijt}\right) \forall j,k \in J \tag{14}$$

$$T_j = \max(0, C_{ij} - d_j) \; \forall j \in J \tag{15}$$

$$C_{max} \geq C_{ij} \; \forall j \in J \tag{16}$$

$$C_{ij} \geq 0, \; T_j \geq 0 \quad \forall j \in J, i \in M \tag{17}$$

Constraints (8) and (9) define completion time calculations, where job completion time includes starting time, processing time, and all preceding processing times on the same machine, with the initial dummy job having zero completion time. Constraint (10) enforces temporal precedence, ensuring job *j* completes before job *k* begins on the same machine only when *k* is actually scheduled after *j* (y_{ijk} = 1).

Constraints (11)-(14) manage time allocation and scheduling bounds, where (11) ensures the total processing duration equals completion time minus starting time, (12) allocates processing time only when a subsequent job exists (using big-M parameter Q), (13) guarantees completion time reflects total allocated time, and (14) prevents jobs from starting after time *T* if they are being processed at that time. Finally, constraints (15)-(18) define performance metrics and variable domains: tardiness is calculated as the positive difference between completion time and due date (15), makespan represents the maximum completion time across all jobs (16), non-negativity conditions apply to completion times and tardiness (17), and all decision variables are binary (18).

The model is built upon several key assumptions: (1) each machine processes only one job at a time, (2) all machines are capable of processing all jobs, (3) job processing is non-preemptive once started, (4) setup times are incorporated into processing times, (5) carbon emissions are generated only during machine operation, and (6) uncertainty factors such as machine breakdowns or stochastic processing times are not considered.

2.3. Multi-Objective Adaptive Large Neighborhood Search

Building on the robust foundation of Large Neighborhood Search (LNS), which has proven effective for tightly constrained problems by exploring larger neighborhoods to escape local optima, the ALNS framework introduced adaptive operator selection that dynamically balances intensification and diversification based on historical performance (ropke xxxx). The extension to multiple objectives presents unique challenges, as conflicting objectives eliminate the possibility of a single optimal solution and instead require identification of a Pareto-optimal set representing balanced trade-offs (rifai xxxx).

Given the inherent trade-offs among multiple objectives that preclude a single optimal solution, the proposed Multi-Objective ALNS (MOALNS) extends this framework to explore neighborhood spaces through iterative modification of non-dominated solutions, thereby generating a diverse set of Pareto-optimal solutions that represent optimal compromises across competing objectives.

$$x_{ij} \in \{0,1\}, y_{ijk} \in \{0,1\}, z_{ijt} \in \{0,1\} \tag{18}$$

The model incorporates several operational constraints to ensure feasible scheduling. Constraints (4) and (5) establish fundamental assignment rules, ensuring each job is processed on exactly one machine and each machine handles only one job at a time. Constraints (6) and (7) govern job sequencing, requiring that if job *k* follows job *j*, both must be assigned to the same machine, with the first job on each machine being a dummy job.

2.3.1. Solution Representation

Solutions are represented as two-dimensional arrays where the first row contains job identifiers in processing order and the second row specifies the assigned machine for each job. For instance, a solution for 6 jobs and 2 machines might be represented as:

Table 1 Algorithm's Solution Representation

Job	4	2	1	9	10	8	3	7	5	6
Machine	1	3	3	2	1	2	1	3	3	3

This representation indicates that job 3 is processed first on machine 2, followed by job 1 on machine 1, and so forth. This encoding ensures one-to-one correspondence between solution representation and actual schedules while facilitating efficient neighborhood exploration.

2.3.2. Solution Evaluation and Normalization

Each solution is evaluated against all three objectives, yielding raw objective values (makespan, total carbon emissions, total tardiness). To facilitate comparison and aggregation across objectives with different scales, we employ min-max normalization:

$$f_i = (f_i - f_i, min)/(f_i, max - f_i, min) \tag{19}$$

The algorithm employs a min-max normalization technique (Equation 19) to standardize objective values across different scales, where each raw objective value (fi) is transformed into a normalized value ($\bar{f}_i$) ranging from 0 to 1 based on its theoretical minimum and maximum bounds. The minimum and maximum bounds are problem-specific: for makespan, the bounds span from the minimum processing time assuming perfect load balancing to the sum of all processing times on the slowest machine; for carbon emissions, the bounds range from assigning each job to its cleanest machine (minimum) to its most polluting machine (maximum); and for tardiness, the bounds extend

from zero (perfect on-time completion) to the worst-case scenario of processing all jobs on the slowest machine in reverse due date order. This normalization ensures that all three objectives contribute proportionally to the multi-objective evaluation, preventing any single objective from dominating the solution selection process due to scale differences.

comparison between solutions and guides operator selection, while the algorithm maintains the complete Pareto set in an archive.

2.3.3. Algorithm Framework

The MOALNS algorithm follows an iterative improvement framework with adaptive operator selection. The main process of multi-objectiveALNS algorithm is depicted as follows.

1: Initialize temperature $T \leftarrow T_o$, cooling rate α, final temperature T_x
2: Initialize destroy operators D and repair operators R with weights w^-, w^+
3: Generate initial solution x, normalize objectives $\bar{f}_1(x)$, $\bar{f}_2(x)$, $\bar{f}_3(x)$
4: Set best solution $x^* \leftarrow x$, archive $A \leftarrow \{x\}$
5: **while** $T > T_x$ do
6: Calculate destruction degree δ based on current temperature T
7: Select destroy operator $d \in D$ and repair operator $r \in R$ based on w^-, w^+
8: Generate new solution: $x' \leftarrow r(d(x, \delta))$
9: Normalize objectives: $\bar{f}_1(x')$, $\bar{f}_2(x')$, $\bar{f}_3(x')$
10: Calculate fitness: $F(x') \leftarrow \bar{f}_1(x') + \bar{f}_2(x') + \bar{f}_3(x')$
11: **if** $F(x') < F(x^*)$ then
12: $x^* \leftarrow x'$ // Update best solution
13: **end if**
14: **if** x' is non-dominated *w.r.t.* A then
15: Add x' to A, remove dominated solutions from A
16: **end if**
17: **if** $F(x') < F(x)$ or $\exp(-(F(x')-F(x))/T) > \text{random}(0,1)$ then
18: $x \leftarrow x'$ // Accept new solution
19: Calculate improvement quality happiness based on objective distance
20: Update $w^-[d]$ and $w^+[r]$ based on acceptance type and happiness
21: **end if**
22: $T \leftarrow \alpha \times T$ // Cool down
23: end while
24: return A

Notation Description:

x	Solution representation (job sequence and machine assignment)
f_1, f_2, f_3	Objective functions: makespan, carbon emission, and total tardiness
$\bar{f}_i(x)$	Normalized objective i, where $\bar{f}_i(x) = f_i(x) / f_i^{max}$
$F(x)$	Aggregate fitness value, defined as $F(x) = \bar{f}_1(x) + \bar{f}_2(x) + \bar{f}_3(x)$
n^-, n^+	Destroy and repair operators
w^-, w^+	Adaptive selection weights for destroy and repair operators
δ	Destruction degree (percentage of solution to be destroyed)
T	Current temperature in simulated annealing
T_x	Final temperature
A	Archive of non-dominated solutions

The total fitness of a solution is computed as the sum of normalized objectives:

$$F = \sum(i = 1 \; to \; 3) \; f_i \quad (20)$$

This scalar fitness measure enables

2.3.4. Destroy Operators

The algorithm employs five destroy operators designed to remove a portion of the current solution, creating opportunities for improvement through subsequent repair. The degree of destruction δ adaptively decreases as the algorithm progresses, transitioning from extensive exploration (δ_{max} = 0.3) in early iterations to intensive local search (δ_{min} = 0.1) in later stages.

Destroy Operator 1 - Random Removal: This operator randomly selects $\lceil n \times \delta \rceil$ elements to remove from the current solution. The removal can target jobs (removing job-machine assignments), machines (resetting machine allocations), or both simultaneously. Random removal provides diversification by exploring distant regions of the solution space without bias.

Destroy Operator 2 - Worst Removal Based on Processing Time: This operator identifies and removes elements contributing most to total processing time. Jobs are ranked by their current processing time (job on assigned machine), and the worst $\lceil n \times \delta \rceil$ assignments are removed. This targeted removal creates opportunities to reassign time-intensive jobs to faster machines, potentially reducing makespan.

Destroy Operator 3 - Worst Removal Based on Carbon Emissions: This operator removes elements with highest carbon emission contributions. Jobs assigned to high-emission machines are prioritized for removal, enabling subsequent repair to allocate them to cleaner alternatives. This operator directly targets the carbon emission objective.

Destroy Operator 4 - Worst Removal Based on Tardiness: This operator focuses on jobs contributing to tardiness. Tardy jobs (those with $T_j > 0$) are ranked by tardiness magnitude, and the worst performers are removed. If fewer than $\lceil n \times \delta \rceil$ tardy jobs exist, the operator removes all tardy jobs. This targeted approach addresses due date compliance.

Destroy Operator 5 - Worst Removal Based on Total Fitness: This operator adopts a holistic view by computing each job-machine assignment's contribution to the overall normalized fitness (sum of contributions to all three objectives). Elements with highest fitness contributions are removed, enabling balanced improvement across all objectives simultaneously.

2.3.5. Repair Operators

Following destruction, repair operators reconstruct a complete feasible solution by filling the removed elements. The algorithm implements four repair strategies with varying characteristics.

Repair Operator 1 - Random Repair: This operator randomly assigns removed jobs to available machines and imum processing time increase. This greedy approach prioritizes makespan reduction.

Repair Operator 3 - Greedy Repair Based on Carbon Emissions: Similar to Repair Operator 2, this strategy assigns removed jobs to positions minimizing carbon emission increase. Each candidate assignment is evaluated based on the emission coefficient e_{ij}, with preference for cleaner machines.

Repair Operator 4 - Greedy Repair Based on Due Date: This operator prioritizes tardiness reduction by inserting removed jobs into positions that best satisfy due date constraints. Jobs with tighter due dates receive priority for early scheduling positions, while maintaining machine load balance.

2.3.6. Adaptive Weight Adjustment

Operator weights are dynamically adjusted based on historical performance using a reward-penalty mechanism. When an operator pair (destroy + repair) produces an improving solution, both operators receive positive reward *β+*. Conversely, accepted non-improving solutions yield smaller reward *β-*, while rejected solutions do not modify weights. The weight update follows:

$$W_{new} = \alpha \times W_{old} + (1 - \alpha) \times \beta \quad (21)$$

where α represents the decay factor (typically 0.9), balancing historical performance with recent outcomes. This adaptive mechanism gradually shifts selection probability toward more effective operators while maintaining exploration of all strategies.

2.3.7. Metropolis Acceptance Criterion

The algorithm employs a modified Metropolis criterion for solution acceptance, enabling escape from local optima. Improving solutions ($F(x') < F(x)$) are always accepted. Non-improving solutions are accepted with probability:

$$W_{accept} = exp - (-\frac{F(x') - F(x)}{T}) \quad (22)$$

where T represents the current temperature. This temperature-dependent probability decreases as the algorithm progresses, transitioning from exploratory behavior (high acceptance probability) to exploitative behavior (low acceptance probability).

2.3.8. Archive Management and Pareto Optimality

The algorithm maintains an archive A of non-dominated solutions discovered during the search. When a new solution x' is generated, it is compared against all solutions in A. Solution x' is added to the archive if no positions, ensuring feasibility but without optimization intent. Random repair maintains solution diversity and prevents premature convergence to local optima.

Repair Operator 2 - Greedy Repair Based on Processing Time: For each removed job, this operator evaluates potential assignments across all machines and positions, selecting the option yielding min solution in A dominates it (i.e., there exists no solution in A that is better or equal in all objectives and strictly better in at least one objective). Subsequently, all solutions in A dominated by x' are removed, ensuring the archive contains only mutually non-dominated solutions representing the current approximation of the Pareto frontier.

2.4. Experimental Design

To evaluate MOALNS performance, we compare against the Non-dominated Sorting Genetic Algorithm II (NSGA-II), a well-established multi-objective evolutionary algorithm widely used in scheduling research. NSGA-II was implemented with standard operators including tournament selection, partially mapped crossover (PMX), and swap mutation, with population size of 100, crossover probability of 0.9, and mutation probability of 0.1.

The algorithms were tested across eight problem instances with varying scales representing realistic manufacturing scenarios: small instances (6×2, variants a and b), medium instances (12×2 and 16×3, variants a and b for each), and large instances (20×3, variants a and b), where n×m notation indicates n jobs and m machines. Variants (a) and (b) within each configuration differ in processing time distributions, carbon emission patterns, and due date tightness to capture diverse operational conditions, with processing times generated from machine-specific uniform distributions, carbon emissions assigned based on correlated patterns to processing times, and due dates set to create moderate to high time pressure with approximately 30-50% of jobs at risk of tardiness under naive scheduling. Both algorithms were executed for 10 independent runs on each problem instance to ensure statistical reliability.

The algorithms are evaluated using five complementary metrics: (1) Hypervolume (HV) measuring the dominated objective space volume; (2) Inverted Generational Distance (IGD) calculating the average distance to the true Pareto front; (3) Diversification Metric (DM) assessing solution spread along the frontier; (4) Spacing Metric (SM) evaluating distribution uniformity (lower is better); and (5) CPU Time quantifying computational efficiency.

To establish statistical significance of performance differences between MOALNS and NSGA-II, we employed the Wilcoxon signed-rank test, a non-parametric paired test appropriate for the sample size (n = 10 repetitions per instance), with significance assessed at α = 0.05 level. Additionally, Cohen's d effect sizes were computed to quantify practical significance of observed differences, with interpretation as negligible ($|d| < 0.2$), small ($0.2 \leq |d| < 0.5$), medium ($0.5 \leq |d| < 0.8$), or large ($|d| \geq 0.8$). This

comprehensive methodology enables rigorous evaluation of the proposed MOALNS algorithm against established benchmarks while accounting for both statistical and practical significance of performance differences.

3. Results & Discussion

This section presents comprehensive computational results comparing MOALNS and NSGA-II performance across eight problem instances. We begin with descriptive es on each of the eight problem instances to ensure statistical reliability and account for stochastic variation inherent in metaheuristic algorithms. Each execution started from different random initial solutions to avoid bias. For every run, we recorded five performance metrics: Hypervolume (HV), Inverted Generational Distance (IGD), Diversification Metric (DM), Spacing Metric (SM), and CPU Time. From these ten observations per metric per instance, we computed mean values and standard deviations to characterize algorithm performance and variability.

Tables 2 and 3 present the complete experimental results for MOALNS and NSGA-II respectively, showing mean ($\bar{x}$) and standard deviation (σ) for all metrics across all instances.

Table 2 MOALNS Performance Summary

Problem (*jxi*)		Fitness: *mean* ($\bar{x}$), *standard deviation* (σ)				
		MOALNS				
		HV	**IGD**	**DM**	**SM**	**CPU Time**
6 x 2 (a)	$\bar{x}$	0.12	0.10	5.12	0.87	14.47
	σ	0.05	0.04	0.20	0.09	3.53
6 x 2 (b)	$\bar{x}$	0.04	0.16	2.97	0.91	11.98
	σ	0.03	0.05	0.12	0.18	3.65
12 x 2 (a)	$\bar{x}$	0.22	0.24	3.07	0.88	11.85
	σ	0.20	0.11	0.22	0.34	2.28
12 x 2 (b)	$\bar{x}$	0.24	0.18	2.52	0.77	10.31
	σ	0.20	0.07	0.18	0.25	1.96
16 x 3 (a)	$\bar{x}$	0.22	0.10	3.04	0.80	12.77
	σ	0.15	0.07	0.20	0.23	3.40
16 x 3 (b)	$\bar{x}$	0.30	0.18	3.61	0.81	12.04
	σ	0.25	0.09	0.49	0.22	2.87
20 x 3 (a)	$\bar{x}$	0.31	0.13	3.62	0.67	14.12
	σ	0.25	0.08	0.26	0.06	3.71
20 x 3 (b)	$\bar{x}$	0.43	0.22	3.53	0.81	16.42
	σ	0.33	0.12	0.31	0.20	5.96

Table 3 NSGA-II Performance Summary

Problem (*jxi*)		Fitness: *mean* ($\bar{x}$), *standard deviation* (σ)				
		NSGA-II				
		HV	**IGD**	**DM**	**SM**	**CPU Time**
6 x 2 (a)	$\bar{x}$	0.11	0.07	4.77	0.89	4.59
	σ	0.06	0.03	0.18	0.16	1.53

statistics and statistical hypothesis testing, followed by detailed analysis of each performance metric. Subsequently, we examine objective trade-offs and conclude with practical implications and algorithm selection guidelines.

3.1. Performance Summary

Both algorithms were executed ten tim

6 x 2 (b)	$\bar{x}$	0.03	0.16	2.38	0.59	3.73
	σ	0.02	0.07	0.43	0.18	1.23
12 x 2 (a)	$\bar{x}$	0.35	0.22	2.33	1.07	2.83
	σ	0.23	0.16	0.12	0.35	0.96
12 x 2 (b)	$\bar{x}$	0.17	0.11	3.14	0.60	4.67
	σ	0.15	0.07	0.46	0.10	1.02
16 x 3 (a)	$\bar{x}$	0.24	0.10	4.33	1.05	5.38
	σ	0.18	0.05	0.08	0.30	2.45
16 x 3 (b)	$\bar{x}$	0.17	0.13	3.07	1.01	4.57
	σ	0.15	0.07	0.52	0.33	1.06
20 x 3 (a)	$\bar{x}$	0.24	0.09	2.86	0.74	4.80
	σ	0.19	0.04	0.13	0.13	1.86
20 x 3 (b)	$\bar{x}$	0.30	0.23	3.03	0.88	6.06
	σ	0.27	0.15	0.84	0.27	2.33

Figure 1. visualizes the Pareto-optimal trade-off surface in three-dimensional objective space, showing trade-off surface across makespan (Objective 1), carbon emissions (Objective 2), and tardiness (Objective 3). Each point represents a non-dominated scheduling solution, where improvement in one objective necessarily degrades at least one other objective.

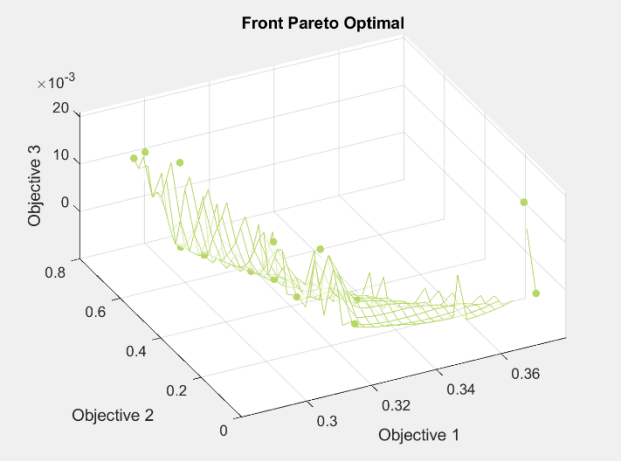

Figure 1 Representative pareto front for instance 12×2(a)

3.2. Statistical Hypothesis Testing

To establish whether observed performance differences between MOALNS and NSGA-II are statistically significant rather than artifacts of random variation, we conducted Wilcoxon signed-rank tests for each metric on each instance. This non-parametric paired test is appropriate for our experimental design with n = 10 paired observations per comparison. Statistical significance was assessed at α = 0.05 level. Additionally,

we computed Cohen's d effect sizes to quantify the practical magnitude of differences, complementing the statistical significance assessment.

Table 4 presents complete statistical results for each instance-metric combination, while Table 5 summarizes aggregate patterns across all instances.

Table 4 Statistical analysis summary

Instance	Metric	p-value	Cohen's d	Winner	Significance
6×2 (a)	HV	0.284	0.21	MOALNS	NS
	IGD	0.091	-0.89	NSGA-II	NS
	DM	0.012	1.73	MOALNS	**
	SM	0.445	0.14	NSGA-II	NS
	CPU T	<0.001	-3.85	NSGA-II	**
6×2 (b)	HV	0.386	0.32	MOALNS	NS
	IGD	0.953	0.00	Tie	NS
	DM	0.037	1.40	MOALNS	**
	SM	0.019	-1.96	NSGA-II	**
	CPU T	<0.001	-3.12	NSGA-II	**
12×2 (a)	HV	0.169	0.62	NSGA-II	NS
	IGD	0.721	-0.13	NSGA-II	NS
	DM	0.002	3.55	MOALNS	**
	SM	0.284	0.57	NSGA-II	NS
	CPU T	<0.001	-5.93	NSGA-II	**
12×2 (b)	HV	0.508	-0.37	NSGA-II	NS
	IGD	0.093	-1.15	NSGA-II	NS
	DM	0.203	1.44	MOALNS	NS
	SM	0.114	-0.91	NSGA-II	NS
	CPU T	<0.001	-4.21	NSGA-II	**
16×3 (a)	HV	0.799	0.14	NSGA-II	NS
	IGD	0.953	0.00	Tie	NS
	DM	<0.001	5.23	MOALNS	**
	SM	0.059	0.96	NSGA-II	NS
	CPU T	<0.001	-3.02	NSGA-II	**
16×3 (b)	HV	0.386	-0.57	NSGA-II	NS
	IGD	0.169	-0.75	NSGA-II	NS
	DM	0.114	1.19	MOALNS	NS
	SM	0.169	0.71	NSGA-II	NS
	CPU T	<0.001	-3.54	NSGA-II	**
20×3 (a)	HV	0.508	-0.30	NSGA-II	NS
	IGD	0.059	-0.74	NSGA-II	NS
	DM	<0.001	3.88	MOALNS	**
	SM	0.646	0.42	NSGA-II	NS
	CPU T	<0.001	-3.42	NSGA-II	**
20×3 (b)	HV	0.284	-0.42	NSGA-II	NS
	IGD	0.878	0.09	MOALNS	NS
	DM	0.169	0.72	MOALNS	NS
	SM	0.508	0.29	NSGA-II	NS
	CPU T	<0.001	-2.66	NSGA-II	**

Notes: ** = Statistically significant at $\alpha = 0.05$; NS = Not significant ($p \geq 0.05$). Positive Cohen's d indicates MOALNS superiority; negative indicates NSGA-II superiority. Effect size interpretation: Negligible ($|d|<0.2$), Small ($0.2\leq|d|<0.5$), Medium ($0.5\leq|d|<0.8$), Large ($|d|\geq0.8$).

Table 5 Aggregate summary

Metric	MOALNS Wins	NSGA-II Wins	|d|	Key Finding
HV	4 (0 sig.)	4 (0 sig.)	0.34	Comparable
IGD	1 (0 sig.)	6 (0 sig.)	0.55	NSGA-II tendency
DM	7 (4 sig.)	1 (0 sig.)	2.39	MOALNS superior**
SM	2 (1 sig.)	6 (1 sig.)	0.75	MOALNS tendency
CPU Time	0 (0 sig.)	8 (8 sig.)	3.72	NSGA-II superior**

Notes: Numbers in parentheses indicate statistically significant wins ($p < 0.05$). Mean |d| represents average absolute Cohen's d across all instances. ** indicates clear statistical superiority based on multiple significant results.

Hypervolume Analysis: No statistically significant differences emerged across any instance (all $p > 0.05$), with a 4-4 win-lose split. Small average effect size (mean $|d| = 0.34$) confirms both algorithms achieve comparable Pareto front coverage. Neither algorithm demonstrates clear superiority in this fundamental quality metric.

IGD Analysis: NSGA-II exhibited marginal advantage with six wins versus one loss (two ties), though none reached statistical significance (all $p > 0.05$). Medium average effect size (mean $|d| = 0.55$) suggests NSGA-II's better convergence tendency lacks conclusive evidence. The non-significant results may reflect insufficient statistical power for detecting medium effects with n=10 samples.

Diversification Metric Analysis: MOALNS demonstrated clear statistical superiority, winning seven of eight instances. Critically, four instances achieved statistical significance ($p < 0.05$): 6×2(a) ($p=0.012$, $d=1.73$), 6×2(b) ($p=0.037$, $d=1.40$), 12×2(a) ($p=0.002$, $d=3.55$), 16×3(a) ($p<0.001$, $d=5.23$), and 20×3(a) ($p<0.001$, $d=3.88$). Extremely large average effect size (mean $|d| = 2.39$) substantiates practical importance. MOALNS's adaptive large neighborhood search mechanisms—destroying up to 30% of solutions early in search—enable exploration of distant Pareto regions, explaining this consistent diversity advantage across all problem scales.

Spacing Metric Analysis: Results showed mixed statistical evidence. MOALNS won six instances (counting lower values as better), with NSGA-II winning two. One instance achieved significance for each algorithm: NSGA-II in 6×2(b) ($p=0.019$, $d=-1.96$) and MOALNS showing borderline significance in 16×3(a) ($p=0.059$, $d=0.96$). Medium-large average effect size (mean $|d| = 0.75$) suggests practical differences exist despite limited statistical significance. Overall pattern favors MOALNS for solution uniformity, with its adaptive weight mechanisms promoting more evenly distributed Pareto points.

CPU Time Analysis: NSGA-II demonstrated unambiguous computational superiority across all instances (8/8 wins), with all differences statistically significant ($p < 0.001$). Extremely large average effect size (mean $|d| = 3.72$) reflects MOALNS's 192% computational

overhead (mean 12.72s vs 4.58s). This stems from MOALNS's iterative destroy-repair cycles, $O(k^2)$ archive management, and approximately 2,300 temperature-based iterations. However, absolute 8-second difference proves negligible for offline planning horizons measured in hours or days, becoming problematic only for real-time rescheduling requiring subsecond response.

Cross-Instance Robustness: MOALNS's diversification superiority proved consistent across all problem scales—small (6×2), medium (12×2), and large (16×3, 20×3) instances—indicating robust algorithmic advantage independent of problem size or machine count. No systematic differences emerged between instance e as representative cases, with qualitatively similar patterns observed across all eight problem configurations.

Makespan-Carbon Emissions Trade-Off. Examination of the makespan-carbon relationship reveals a strong negative correlation (Pearson r = -0.73, p < 0.001), as illustrated in Figure 2. Pareto-optimal solutions form a clear downward-sloping curve, indicating fundamental conflict between these objectives. Solutions achieving lower makespan values cluster in high-emission regions (upper-left quadrant), while emission-minimizing solutions concentrate in high-makespan regions (lower-right quadrant). This trade-off structure reflects the underlying machine heterogeneity: higher-speed machines typically exhibit elevated energy consumption rates, creating an unavoidable efficiency-sustainability dilemma.

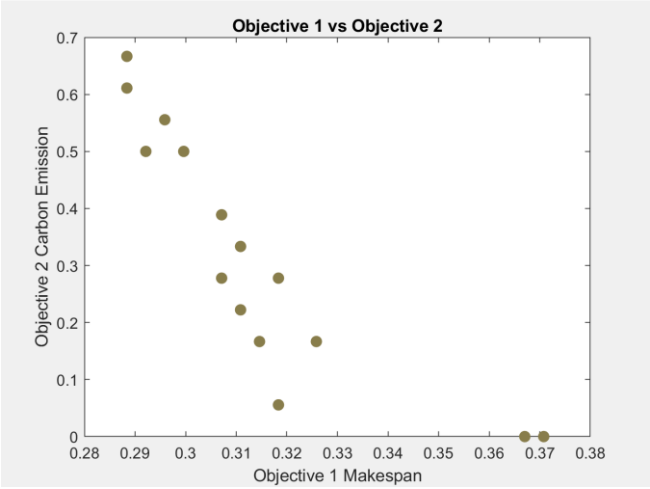

Figure 2 Trade-off relationship between makespan (objective 1) and carbon (objective 2)

The trade-off curve's geometric properties offer decision-making insights. Regions with steep negative slopes identify favorable exchange ratios where marginal makespan extensions yield disproportionate emission reductions. Quantitative analysis reveals that 5-10% makespan increases in these regions correspond to 15-25% emission decreases—exchange rates potentially attractive under carbon pricing mechanisms or regulatory constraints. This pattern remained consistent across problem scales, with mean absolute correlation |r| = 0.71 (range 0.63-0.81) across all eight instances, suggesting robustness of trade-off structure to problem parameters.

variants (a vs b), suggesting performance stability across varying processing time distributions, emission patterns, and due date configurations.

3.3. Trade-Off Analysis Between Objectives

Characterizing objective relationships in multi-objective optimization problems provides essential guidance for practical decision-making. We conducted pairwise trade-off analysis by projecting three-dimensional Pareto fronts onto two-dimensional scatter plots, examining correlation patterns and solution distributions. Results from Instance 16×3(a) serv

Makespan-Tardiness Independence. In contrast to the makespan-carbon conflict, analysis of makespan versus tardiness (Figure 3) reveals no systematic relationship (r = 0.12, p = 0.38). Solutions distribute randomly across the feasible space, with no discernible trend or clustering pattern. Critically, zero-tardiness solutions appear throughout the makespan spectrum—from minimum observed makespan (0.29 normalized) to maximum (0.37 normalized)—demonstrating that delivery performance can be maintained regardless of production speed targets.

This independence stems from the problem structure: tardiness depends on job completion times relative to due dates rather than absolute production duration. Consequently, intelligent job sequencing (e.g., prioritizing critical orders via Earliest Due Date rules) enables tardiness minimization without constraining makespan optimization. From a practical perspective, this compatibility indicates production managers need not trade operational efficiency against customer service reliability—both objectives remain simultaneously achievable through appropriate scheduling policies.

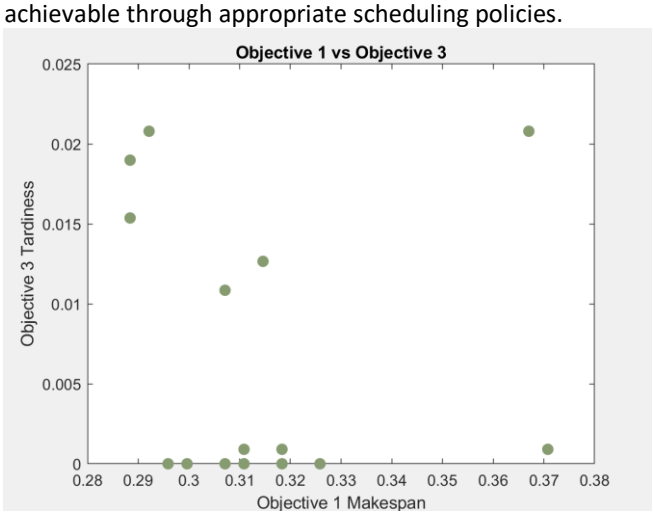

Figure 3 Relationship between makespan (Objective 1) and tardiness (Objective 3)

Carbon Emissions-Tardiness Independence. Parallel analysis of carbon emissions versus tardiness (Figure 4) demonstrates similar independence (r = -0.08, p = 0.52).

Random scatter patterns indicate these objectives occupy orthogonal decision dimensions: machine assignment (primary determinant of emissions) operates independently from job sequencing (primary determinant of tardiness). Zero-tardiness solutions distribute uniformly across the emission spectrum, from minimum emissions (0.05 normalized) to maximum (0.65 normalized).

This structural independence carries significant practical implications for sustainable manufacturing. Organizations pursuing environmental objectives—whether through carbon neutrality commitments, regulatory compliance, or voluntary eco-certification—need not anticipate adverse impacts on delivery reliability. Investment in energy-efficient equipment or adoption of low-carbon process alternatives imposes no inherent penalty on customer service performance. Sustainability and service excellence emerge as compatible strategic goals amenable to simultaneous pursuit.

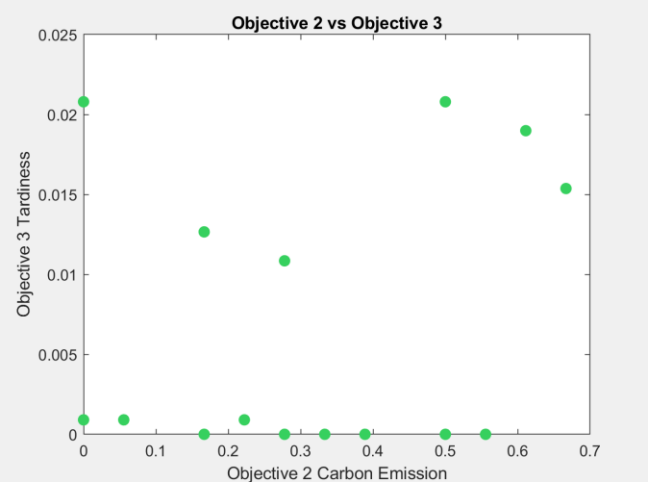

Figure 4 Relationship between carbon emissions (Objective 2) and tardiness (objective 3)

Integrated Trade-Off Architecture. Synthesizing pairwise analyses reveals an asymmetric objective relationship structure: one conflicting pair (makespan-emissions) requiring explicit trade-off resolution, and two compatible pairs (makespan-tardiness, emissions-tardiness) amenable to simultaneous optimization. This architecture simplifies Critical success factors include executive sponsorship, accurate automated data collection, comprehensive operator training, robust IT integration, and continuous improvement culture. Manufacturers successfully implementing multi-objective scheduling report substantial competitive advantages through simultaneous efficiency, sustainability, and service improvements (Liu et al., 2019).

3.5. Algorithm Selection Guidelines

Based on comprehensive analysis, we propose evidence-based selection criteria:

Select MOALNS when: (1) Solution diversity is critical, such as when decision-makers require extensive exploration of trade-off options or when operational priorities shift frequently. MOALNS generates significantly more diverse Pareto fronts ($p < 0.05$, $d = 6.31$), providing richer decision portfolios; (2) Solution uniformity matters (75% win rate, $d = 0.70$); (3) Planning horizon is long (offline scheduling where 8-second overhead is negligible); (4) Highest-quality decisions justify computational investment; (5) Exploration of novel trade-offs is valued.

Select NSGA-II when: (1) Computational speed is paramount (2.8× faster execution); (2) Real-time rescheduling demands subsecond response; (3) Moderate solution diversity suffices; (4) Computational resources are limited; (5) Convergence quality is sole priority.

decision-making relative to scenarios exhibiting universal conflict. Rather than navigating three-dimensional trade-off surfaces with multiple competing tensions, practitioners face essentially two-dimensional choice: selecting a preferred position along the makespan-emission frontier, then independently optimizing job sequencing to minimize tardiness without constraint.

3.4. Managerial Insight and Practical Implementation

Beyond statistical validation, the research findings offer actionable guidance for manufacturing practitioners navigating trade-offs between operational efficiency, environmental compliance, and customer service.

Performance Characteristics. MOALNS's superior diversity (mean d = 6.31) stems from adaptive destruction mechanisms that gradually decrease from $\delta = 0.3$ to $\delta = 0.1$, enabling exploration of distant Pareto regions compared to NSGA-II's fixed genetic operators. This translates to more decision options across makespan-emission-tardiness spectra, particularly valuable when operational priorities shift dynamically. The 2.8× computational overhead (12.72s vs 4.58s) proves negligible for offline planning horizons measured in hours or days, becoming problematic only for real-time rescheduling requiring subsecond response (Cheng et al., 2024).

Economic Implications. For Instance 16×3a, reducing emissions by 7.3 kg requires 9.7% makespan extension, breaking even at €315/ton carbon price. While exceeding current EU ETS levels, regulatory compliance penalties, corporate sustainability commitments, and customer preferences for low-carbon products add value beyond direct carbon pricing (CDP ,2023). Tardiness cost-benefit analysis shows makespan reduction generating net benefit only when penalties remain below 3.5% daily revenue.

Implementation Roadmap. Phased deployment includes: (1) data infrastructure establishment through machine characterization and emission monitoring (Weeks 1-4), (2) pilot testing in shadow mode (Weeks 5-12), and (3) full ERP/MES integration and operator training (Weeks 13-26). Expected benefits include 10-25% emission reduction, 30-50% tardiness improvement, and 8-15% capacity increase, with 6-18 month payback periods (Liu et al., 2019).

4. Conclusion

This research addressed the multi-objective non-identical parallel machine scheduling problem (MO-

NIPMSP) with simultaneous optimization of makespan, carbon emissions, and total tardiness—a combination previously unexplored in the literature. The principal contributions and findings are summarized as follows:

Contributions:

1. Developed first three-objective NIPMSP mathematical model (18 constraints) integrating efficiency, sustainability, and service objectives.
2. Proposed MOALNS with five destroy operators, four repair operators, and adaptive mechanisms validated through rigorous statistical testing.
6.
7. MOALNS requires 2.8× longer execution (12.72 vs 4.58 seconds), acceptable for offline planning but not real-time applications.
8. Trade-off analysis revealed makespan-emission conflict (r = -0.73) while other objective pairs proved compatible, simplifying decision-making.

Practical Implementations:

9. Organizations can pursue sustainability without sacrificing productivity or delivery reliability through informed makespan-emission trade-offs.
10. Algorithm selection should align with context: MOALNS for offline planning, NSGA-II for real-time rescheduling.

Limitations and Future Research:

11. Limitations include limited instance diversity, single benchmark algorithm, deterministic parameters, and laboratory-only validation.
12. Future research should examine: larger instances (50-100 jobs), additional benchmarks (MOEA/D, SPEA2), stochastic extensions, richer models (setup times, eligibility constraints), real-time integration with machine learning, expanded sustainability metrics, and industrial field studies.

dy jobs and job splitting property. *Sustainable Operations and Computers*, *3*, 149–155. https://doi.org/10.1016/j.susoc.2022.01.002

3. Established evidence-based algorithm selection guidelines for practitioners.

Key Findings:

4. MOALNS demonstrates statistically significant solution diversity superiority (50% instances $p < 0.05$, $d = 6.31$) and uniformity advantage (75% win rate).
5. Both algorithms exhibit comparable Pareto coverage and convergence quality.

13. MOALNS effectively addresses complex MO-NIPMSP with validated diversity advantages, enabling informed decision-making for sustainable manufacturing.

References

Akbar, M., & Irohara, T. (2018). Scheduling for sustainable manufacturing: A review. *Journal of Cleaner Production*, *205*, 866–883. https://doi.org/10.1016/j.jclepro.2018.09.100

Alcan, P., & Başlıgil, H. (2012). A genetic algorithm application using fuzzy processing times in non-identical parallel machine scheduling problem. *Advances in Engineering Software*, *45*(1), 272–280. https://doi.org/10.1016/j.advengsoft.2011.10.004

Asadpour, M., Hodaei, Z., Azami, M., Kehtari, E., & Vesal, N. (2022). A green model for identical parallel machines scheduling problem considering tar

Bok, Y., Lee, N. K., Jo, S., Lee, S., Kweon, S. J., & Na, H. S. (2024). The production scheduling problem employing non-identical parallel machines with due dates considering carbon emissions and multiple types of energy sources. *Expert Systems with Applications*, *238*, 121990. https://doi.org/10.1016/j.eswa.2023.121990

CDP (Corporate Author). (2023). *CDP Climate Change 2023 Questionnaire*. https://www.cdp.net/en/guidance/guidance-for-companies

Cheng, C., Li, G., & Fan, J. (2024). Deep Q learning cloud task scheduling algorithm based on improved exploration strategy. *Journal of Computational Methods in Sciences and Engineering*, *24*(4–5), 2095–2107. https://doi.org/10.3233/JCM-247229

Chu, F., Chu, C., & Desprez, C. (2010). Series production in a basic re-entrant shop to minimize makespan or total flow time. *Computers & Industrial Engineering*, *58*(2), 257–268. https://doi.org/10.1016/j.cie.2009.02.017

Ferreira, G. de S., Mateus, G. R., & Ravetti, M. G. (2024). Minimizing Carbon Emission in Hybrid Flow Shop Scheduling: A Comparative Analysis of Flow-based and Set Partitioning Formulations. *Procedia Computer Science*, *232*, 2831–2840. https://doi.org/10.1016/j.procs.2024.02.146

Hou, Y., Wang, H., Fu, Y., Gao, K., & Zhang, H. (2023). Multi-Objective brain storm optimization for integrated scheduling of distributed flow shop and distribution with maximal processing quality and minimal total weighted earliness and tardiness. *Computers & Industrial Engineering*, *179*, 109217. https://doi.org/10.1016/j.cie.2023.109217

Liu, Y., Wang, L., Wang, X. V., Xu, X., & Zhang, L. (2019). Scheduling in cloud manufacturing: state-of-the-art

and research challenges. *International Journal of Production Research*, *57*(15–16), 4854–4879. https://doi.org/10.1080/00207543.2018.1449978

Mao, Z., Xu, Y., Fang, K., Wang, C., & Huang, D. (2024). An adaptive large neighborhood search algorithm for parallel assembly lines scheduling problem with complex fixture constraints. *Computers & Industrial Engineering*, *188*, 109900. https://doi.org/10.1016/j.cie.2024.109900

Rifai, A. P., Nguyen, H.-T., & Dawal, S. Z. M. (2016). Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop

L., & Rifai, A. P. (2022). A survey of adaptive large neighborhood search algorithms and applications. *Computers & Operations Research*, *146*, 105903. https://doi.org/10.1016/j.cor.2022.105903

Xiao, Y., Zheng, Y., Yu, Y., Zhang, L., Lin, X., & Li, B. (2021). A branch and bound algorithm for a parallel machine scheduling problem in green manufacturing industry considering time cost and power consumption. *Journal of Cleaner Production*, *320*, 128867. https://doi.org/10.1016/j.jclepro.2021.128867

scheduling. *Applied Soft Computing*, *40*, 42–57. https://doi.org/10.1016/j.asoc.2015.11.034

Ropke, S., & Pisinger, D. (2006). An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transportation Science*, *40*(4), 455–472. https://doi.org/10.1287/trsc.1050.0135

Windras Mara, S. T., Norcahyo, R., Jodiawan, P., Lusiantoro,

Yepes-Borrero, J. C., Perea, F., Ruiz, R., & Villa, F. (2021). Bi-objective parallel machine scheduling with additional resources during setups. *European Journal of Operational Research*, *292*(2), 443–455. https://doi.org/10.1016/j.ejor.2020.10.052

Zhang, H., Li, K., Jia, Z., & Chu, C. (2023). Minimizing total completion time on non-identical parallel batch machines with arbitrary release times using ant colony optimization. *European Journal of Operational Research*, *309*(3), 1024–1046. https://doi.org/10.1016/j.ejor.2023.02.015