

Article

Not peer-reviewed version

A Policy-Based Rough Optimization with Large Neighborhood Search for Carbon-Aware Flexible Job Shop Scheduling with Tardiness Penalty

[Saurabh Sanjay Singh](#)* and [Deepak Gupta](#)

Posted Date: 15 April 2026

doi: 10.20944/preprints202604.1052.v1

Keywords: sustainable manufacturing; green flexible job-shop scheduling; reinforcement learning; large neighborhood search; carbon-conscious scheduling



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

A Policy-Based Rough Optimization with Large Neighborhood Search for Carbon-Aware Flexible Job Shop Scheduling with Tardiness Penalty

Saurabh Sanjay Singh * and Deepak Gupta

Department of Industrial, Systems, and Manufacturing Engineering, College of Engineering, Wichita State University, Wichita, KS 67260, USA; sxsingh28@shockers.wichita.edu

Abstract

Sustainable manufacturing increasingly requires production schedules that balance environmental responsibility with delivery reliability. In flexible job shop environments, this challenge is especially difficult because machine assignment and sequencing decisions affect both the carbon footprint of production and the risk of missing job due dates. Motivated by this trade-off, this paper studies the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T), a flexible job shop formulation in which total carbon emissions and total tardiness penalty are treated as the two primary objectives, while energy consumption and makespan are retained as supporting performance indicators. To solve this problem, we propose a Policy-based Rough Optimization with Large Neighborhood Search (Pro-LNS) framework that combines Proximal Policy Optimization for fast, policy-guided construction of feasible schedules with an adaptive large neighborhood search procedure for targeted refinement. The two phases are aligned through a normalized scalarized objective that balances carbon emissions and tardiness penalty while preserving all precedence, eligibility, and machine-capacity constraints. Computational experiments on benchmark instances spanning small, medium, and large workcenter categories show that Pro-LNS produces high-quality schedules with strong due-date performance and controlled carbon emissions. Under equal objective weighting, the method achieves a median optimality gap of 6.12% relative to the exact formulation, with all reported instances remaining within 14%, while requiring only 4.08 seconds on average and at most 10.51 seconds. These results indicate that Pro-LNS is an effective and computationally practical approach for carbon-aware, tardiness-sensitive flexible job shop scheduling.

Keywords: sustainable manufacturing; green flexible job-shop scheduling; reinforcement learning; large neighborhood search; carbon-conscious scheduling

1. Introduction

Manufacturing is changing in a fundamental way. Firms are no longer evaluated only by how efficiently they produce goods, but also by how responsibly they use resources and how well they align with broader sustainability goals. In the past, operational performance and environmental responsibility were often treated as separate concerns. Today, they are increasingly seen as part of the same strategic challenge. This shift is being driven by both internal and external pressures. On one side, manufacturers now view sustainability as a source of innovation, operational improvement, competitive advantage, and long-term profitability [1,2]. On the other side, stricter regulations, changing customer expectations, and global sustainability standards are pushing firms toward cleaner and more accountable production systems [3,4]. These changes are further strengthened by the global push toward net-zero emissions and the broader vision of Industry 5.0, which emphasizes sustainable, resilient, and human-centered manufacturing [5,6]. As a result, sustainability is no longer a secondary issue in manufacturing. It has become central to how production systems are designed, managed, and evaluated [7].

Within this broader transformation, production scheduling takes on a much more important role. Scheduling is no longer only about deciding the order of jobs or the use of machines. It is also a point where firms must balance operational efficiency with environmental responsibility and service performance. Among production scheduling problems, the Flexible Job Shop Scheduling Problem (FJSP) is especially important because it reflects the reality of modern manufacturing systems, where there are alternative machines, routing flexibility, and complex sequencing decisions. In this setting, each job consists of a sequence of operations, and each operation can often be processed on more than one machine. This flexibility creates opportunities to improve system performance, but it also makes the scheduling task much more difficult. The scheduler must decide not only the sequence of operations, but also which machine should process each operation, often under multiple and competing objectives.

Much of the traditional literature on flexible job shop scheduling has mainly focused on productivity-oriented objectives such as makespan, production cost, and machine utilization. Tardiness has also received attention when due-date performance is important [8,9]. However, sustainability-related concerns have historically received much less emphasis [10]. This limited focus is becoming increasingly inadequate. As manufacturers face stronger pressure to decarbonize their operations, it is no longer enough to evaluate schedules only in terms of speed or resource use. Scheduling decisions can also shape the environmental footprint of production, especially when different machines have different processing characteristics and different carbon intensities. Because of this, there is a growing need for scheduling models that explicitly include environmental outcomes instead of treating them as indirect or secondary effects.

At the same time, environmental performance cannot be considered in isolation from delivery performance. In many real manufacturing settings, a schedule is judged not only by how efficiently the shop floor operates, but also by whether customer orders are completed on time. Late jobs can lead to penalties, create disruptions in downstream operations, and weaken customer trust. For this reason, tardiness is a highly meaningful performance measure in practice. From a managerial point of view, the real challenge is not simply to reduce emissions, and it is not simply to avoid delays. The challenge is to make scheduling decisions that balance both concerns in a clear and disciplined way.

This trade-off becomes even more important in flexible job shop environments. The same machine flexibility that helps improve operations can also create clear differences in both environmental impact and delivery performance across different processing routes. Choosing one machine instead of another can change not only the processing time, but also the carbon footprint of the final schedule. In the same way, a routing or sequencing decision that reduces emissions may increase the chance that some jobs are completed after their due dates. So, the scheduling problem is not only a combinatorial problem. It is also a practical decision problem in which manufacturers must balance two responsibilities that are now central to modern production: environmental responsibility and delivery reliability [11].

In this context, focusing on carbon emissions and tardiness is not just a convenient modeling choice. It is a clear and well-justified way to represent the two most important decision outcomes. On the environmental side, carbon emissions are more suitable than energy consumption as the main sustainability objective because energy use by itself is not the final concern. The real concern is the environmental harm caused by that energy use. Carbon emissions capture that harm more directly and connect more closely to decarbonization goals, emission reporting, and net-zero targets that now influence manufacturing strategy. For scheduling decisions, this makes carbon a more meaningful end objective than energy alone. Energy consumption is still important, but mainly because it is one of the main sources through which carbon impact is created [12,13]. On the operational side, tardiness is more meaningful than traditional efficiency measures such as makespan when production is driven by due dates. Makespan shows how early the full schedule is completed, but it does not show whether individual jobs are completed on time. A schedule can look efficient overall and still perform badly if important jobs miss their due dates. Tardiness captures this problem directly. It reflects the service-related failures that matter in practice, including penalties, customer dissatisfaction, and loss of

trust. In due-date-based manufacturing settings, tardiness is therefore not just one more performance measure. It is the measure that most clearly reflects delivery performance from the customer's point of view [14,15].

Taken together, carbon emissions and tardiness should not be seen as two random objectives added to broaden the model. They represent the two main outcome-level criteria that define the real challenge of sustainable manufacturing scheduling. Carbon represents the environmental impact that firms are under growing pressure to reduce. Tardiness represents the delivery failure that firms cannot afford to ignore. Other traditional FJSP and sustainability measures, such as energy consumption, makespan, and machine utilization, are still useful, but in this setting they are better treated as supporting or diagnostic indicators. They do not capture the final decision consequences as directly as carbon emissions and tardiness. For this reason, a scheduling model built around carbon emissions and tardiness provides a clearer, stronger, and more managerially relevant problem definition than a broader model that combines many overlapping objectives [16,17].

Despite the growing importance of sustainable scheduling, the literature still leaves room for a more focused treatment of this specific trade-off. Many existing studies address sustainability through broad multi-objective models that combine several environmental and operational criteria at the same time. Although these studies are valuable, such broad formulations can make it difficult to clearly understand the direct interaction between environmental responsibility and delivery performance. In practice, however, this interaction is often the most immediate and most challenging issue for manufacturers. Firms need schedules that are cleaner, but they also need schedules that remain dependable from the customer's point of view. This creates a strong motivation to study a flexible job shop scheduling formulation in which carbon emissions are treated as the primary environmental objective and tardiness is treated as the primary operational objective.

Motivated by this gap, this paper studies a carbon-aware flexible job shop scheduling setting with a particular focus on tardiness-sensitive production environments, where machine assignment and sequencing decisions are made with explicit attention to environmental impact and delivery performance. To reflect this focus, the problem is defined as the *Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T)*.

The remainder of this paper is organized as follows. Section 2 reviews the related literature on flexible job shop scheduling, sustainable and environmentally aware scheduling, and relevant solution approaches. Section 3 presents the problem definition, details the proposed solution approach, and describes the experimental setup. Section 4 reports and discusses the results and analysis. Finally, Section 5 concludes the paper and outlines directions for future research.

2. Literature Review

The methodological development of flexible job shop scheduling has progressively moved toward more adaptive and hybrid solution strategies as the problem has become richer in both structure and objectives. Early research primarily emphasized production-oriented criteria such as makespan, tardiness, and machine utilization, and addressed the resulting combinatorial complexity through mixed integer programming, genetic algorithms, ant colony optimization, and NSGA-II-type frameworks. These studies established that effective flexible job shop scheduling depends not only on broad exploration, but also on the coordinated treatment of machine assignment and operation sequencing within a tightly coupled search space [18,19]. As the literature developed, this foundation was extended through fuzzy processing settings, simulation-based evaluation, transportation-aware scheduling, and local improvement mechanisms, all of which reinforced the value of structured search and hybrid solution design in increasingly realistic shop-floor environments [20,21].

This methodological progression became even more consequential once environmental considerations were incorporated into the scheduling model. When alternative machines differ not only in processing capability but also in energy use or emissions behavior, machine assignment decisions generate environmental trade-offs at the same time that they generate timing trade-offs. This shift has led

to formulations that jointly consider combinations of makespan, carbon emissions, energy consumption, customer satisfaction, and transport-related effects, thereby expanding both the dimensionality of the objective space and the structural difficulty of the underlying search problem [22,23]. It has also encouraged more integrated scheduling models in which processing and material-handling decisions are considered together, particularly in settings with automated guided vehicles and energy-aware transport interactions [24,25].

Accordingly, environmentally aware flexible job shop scheduling has not merely added further objectives to the classical problem. It has altered the algorithmic demands of the problem itself. Lower-emission or lower-energy machine choices may increase completion times, while decisions intended to reduce idle energy or respond to time-of-use electricity pricing can reshape the sequencing logic of the schedule [26,27]. Dynamic low-carbon settings with job insertions and transfers intensify this interaction by requiring algorithms to respond simultaneously to environmental and operational disruptions [28,29]. Additional considerations such as variable machine speeds, peak power constraints, and machine-state transitions deepen this coupling still further and render static optimization logic increasingly inadequate [30,31]. Idle power effects and underutilized machine states are particularly important in this regard, since they indicate that environmental performance is shaped not only by active processing decisions, but also by how effectively the schedule governs inactive resource behavior [24,32].

These developments help explain why the literature has relied so heavily on adaptive metaheuristics and hybrid search procedures in environmentally conscious flexible job shop settings. Hierarchical integrated scheduling has been used to jointly minimize cost, energy consumption, and makespan in multi-objective production environments [11]. Mixed-integer multi-objective formulations combined with sparrow search have been applied to settings that explicitly balance makespan and carbon emissions [22,23]. Decomposition-based search has supported simultaneous optimization of makespan, energy use, emissions, and workload balance [29,33]. Memetic and knowledge-driven evolutionary frameworks have been used to improve both diversity and intensification across multi-objective flexible job shop variants [34,35]. Other work has explored shuffled frog-leaping, improved buffalo optimization, PSO-GA hybrids, and hybrid discrete imperial competition in order to manage trade-offs among efficiency, workload, and energy-related objectives [36,37]. Similar motivations underlie the use of population-based hybrid heuristics in settings where productivity and environmental criteria must be balanced without sacrificing computational tractability [38,39].

What emerges from this body of work is a clear methodological pattern. As the objective space becomes more coupled and environmentally aware, the literature moves away from rigid or single-mechanism algorithms and toward methods that combine broad search capacity with stronger problem-specific guidance. This same pattern helps explain the growing role of reinforcement learning in scheduling. Reinforcement learning (RL) has become increasingly attractive because it can learn decision policies directly through interaction with the scheduling environment, rather than relying entirely on fixed dispatching rules or manually designed priority logic. This is especially valuable in flexible job shop settings, where the effect of one decision depends strongly on the evolving state of the shop and the downstream structure of the schedule. Recent studies show that RL-based schedulers can outperform classical dispatching rules and many metaheuristics in dynamic job shop environments, particularly when the state space is large and the decision process is sequential and highly context dependent [40,41].

Within this broader RL movement, several methodological streams have emerged. Deep Q-network (DQN) approaches have shown strong performance across multiple scheduling settings and have demonstrated that learned value-based guidance can improve decision quality beyond traditional rule-based baselines [42,43]. Graph-based deep RL (DRL) has further strengthened this direction by improving the representation of scheduling states with varying topology and scale [40,44]. Proximal Policy Optimization (PPO)-based and PPO-DQN hybrid frameworks have added another important layer by showing that policy-gradient-based learning can achieve strong generalization and computational performance in environments with dynamic job arrivals, distributed decision structures,

and large action spaces [44,45]. The literature has therefore gradually moved from asking whether learning can assist scheduling to asking which learning architecture is most appropriate for structurally difficult scheduling environments.

Within that context, PPO is especially relevant for complex flexible job shop applications. Because PPO combines policy-gradient learning with clipped policy updates, it offers a useful balance between policy improvement and training stability, which is particularly important in scheduling environments where decisions are sequential, rewards are highly nonconvex, and early actions influence the quality of the downstream schedule [46,47]. This is especially significant in flexible job shop problems because machine assignment and sequencing decisions are not isolated actions. They propagate through precedence relationships, machine availability, and due-date performance. Compared with other value-based approaches, PPO has therefore become especially attractive in high-dimensional scheduling settings that require both adaptive policy learning and reliable convergence behavior over repeated training episodes [41,45].

At the same time, the scheduling literature also suggests that learned guidance is strongest when it is informed by structural features of the schedule itself. One relevant direction in this regard is the use of critical-path-related information as a supporting signal within the learning process. In flexible job shop environments, critical path structure helps identify operations whose assignment or delay is likely to have disproportionate downstream consequences. When incorporated carefully, such information can strengthen policy learning by making early-stage scheduling decisions more sensitive to the parts of the solution structure that matter most [40,48]. It is most effective, however, as a supporting source of structural awareness rather than as a replacement for broader policy learning.

Even with these advances, the literature also indicates that learning alone is rarely sufficient for the most demanding flexible job shop variants. Several recent studies have integrated RL into broader search architectures, but often around narrower objective sets such as makespan and energy, or around workload proxies rather than explicitly carbon-aware and tardiness-sensitive formulations. A Q-learning-guided MOEA/D framework improved the balance between makespan and total machine energy consumption through adaptive neighborhoods and parameter control [49]. A deep Q-network-based memetic algorithm combined learning guidance with local search and multi-AGV coordination for makespan and energy optimization [48]. Other work has used fuzzy RL for dynamic allocation or actor-critic frameworks for energy-oriented scheduling, further illustrating the growing role of learning-based guidance in flexible job shop environments [50,51]. Taken together, these studies suggest that RL provides a strong adaptive foundation, particularly when incorporated into broader scheduling frameworks with richer structural and objective interactions.

In this context, neighborhood-based search becomes a natural complement to learning-based approaches. Once RL has guided the schedule into a promising region of the solution space, the next challenge is how to improve that schedule without discarding the structure already learned. Large neighborhood search (LNS) is especially valuable for this purpose because it can destroy and reconstruct meaningful parts of a schedule rather than relying solely on very small local moves. This gives it sufficient flexibility to escape weak local structure while still preserving useful incumbent information [52,53]. In flexible job shop scheduling, this is especially important because improvement often requires coordinated changes across both sequencing and machine assignment. Earlier neighborhood-based, tabu-assisted, and local-search-intensive methods already pointed in this direction, even before learning-based methods became prominent [20,21].

The literature therefore points toward a coherent methodological conclusion. Reinforcement learning contributes adaptive global guidance and the capacity to learn meaningful scheduling patterns from repeated interaction. Neighborhood-based search contributes the intensification and structural repair needed to improve those policy-guided solutions once they have been constructed. This division of labor is now visible across job shop scheduling, flow shop optimization, production-maintenance planning, and related combinatorial domains such as routing and packing, where learned decision guidance is followed by adaptive local or neighborhood-based improvement [54,55]. Similar logic

appears in estimation-of-distribution methods with learned operator control and in swarm-based methods strengthened by Q-learning-guided search decisions [56,57]. Broader reviews of scheduling and energy-efficient production support the same interpretation, identifying learned adaptability and search-based intensification as increasingly complementary rather than competing mechanisms [58,59].

Seen through this progression, *A Policy-Based Rough Optimization with Large Neighborhood Search* (Pro-LNS) emerges naturally as an extension of the current literature rather than as a departure from it. The name reflects the underlying logic of the method. The policy-based rough optimization stage is intended to generate an initial schedule that is fast, adaptive, and structurally informed, without requiring the policy component to solve the full combinatorial problem in a single pass. The large neighborhood search stage then takes that policy-guided schedule and refines it through targeted destruction and reconstruction, thereby enabling deeper improvement once a promising search region has been identified. In this sense, “rough optimization” refers to purposeful and informed early-stage guidance, while “large neighborhood search” captures the systematic refinement that follows.

This also aligns closely with the methodological demands of carbon-aware flexible job shop scheduling with tardiness penalty. Such a setting requires a method that can respond to machine heterogeneity, environmental consequences, and tardiness-sensitive production behavior without collapsing into either rigid rule-based logic or purely local search. A policy-based method grounded in PPO offers a strong learning component because of its stability and sequential decision strength. Supporting structural cues such as critical-path information can sharpen that guidance. Large neighborhood search then provides the refinement necessary to improve beyond the initial policy output. Taken together, this methodological combination helps explain why Pro-LNS is well suited to carbon-aware flexible job shop scheduling with tardiness penalty and why it aligns with the literature’s broader movement toward hybrid, adaptive, and structurally informed scheduling methods.

3. Methodology

3.1. Problem Definition

The Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T) extends the classical Flexible Job Shop Scheduling Problem (FJSP) by incorporating environmental and delivery-performance considerations into the scheduling process. In the classical FJSP, n jobs with machine-dependent operation sequences are processed on m flexibly assigned machines. The CAFJSP-T augments this setting by accounting for machine-specific carbon emissions during processing and by penalizing tardiness relative to job due dates.

Under this formulation, each job consists of a sequence of operations subject to precedence constraints, and each operation can be processed only by a subset of eligible machines with machine-dependent processing times. The scheduler must determine both the machine assignment and the processing sequence for all operations while satisfying the standard feasibility requirements of the FJSP, including operation precedence, machine capacity limitations, and operation-machine eligibility restrictions. Unlike the classical formulation, however, these scheduling decisions are evaluated not only in terms of feasibility and production efficiency but also in terms of their environmental and service-level consequences.

Accordingly, the CAFJSP-T is modeled using two primary objectives: total carbon emissions and total tardiness penalty. These two components capture the environmental and due-date-related dimensions of schedule quality, respectively. In addition, total energy consumption and makespan are retained as secondary performance metrics and are reported to provide supplementary insight into overall schedule efficiency and resource utilization.

Problem Assumptions

1. Machines are continuously available throughout the scheduling horizon, with no downtime due to breakdowns or maintenance.
2. Operations are processed without interruption once started, that is, processing is non-preemptive.

3. Processing-time-dependent carbon emissions remain constant for each machine during active operation.
4. All jobs are available for processing at time zero, and job due dates are predetermined and fixed.
5. Tardiness penalties are deterministic and time-invariant over the scheduling horizon.
6. The study considers a deterministic and steady-state production environment, without uncertain processing times, machine failures, sequence-dependent setup times, dynamic job arrivals, or worker-related constraints.
7. Carbon-emission estimation is based on steady-state machine processing conditions and does not incorporate transient operating states, time-varying carbon intensity, or machine warm-up and cool-down effects.

Notations

Sets and Indices:

- $J = \{1, \dots, n\}$: Set of jobs
- $M = \{1, \dots, m\}$: Set of machines
- $\mathcal{O}_j = \{1, \dots, O_j\}$: Operations of job $j \in J$
- $\mathcal{O} = \{(j, o) \mid j \in J, o \in \mathcal{O}_j\}$: Set of all operations
- $M_{j,o} \subseteq M$: Eligible machines for operation (j, o)

Parameters:

- $p_{j,o,m}$: Processing time (mins) of operation (j, o) on machine $m \in M_{j,o}$
- P_m^{proc} : Power consumption of machine m during processing (kW)
- CI_m : Carbon intensity of machine m (kg CO₂/kWh)
- d_j : Due date for job j
- π : Tardiness penalty rate
- $B_1^{(c)}$: Baseline carbon value for instance category c
- $B_2^{(c)}$: Baseline tardiness penalty value for instance category c
- w_1, w_2 : Scalarization weights, where $w_1 + w_2 = 1$
- B : A sufficiently large positive constant

Derived Quantities:

- $e_{j,o,m}^{\text{carbon}} = \frac{p_{j,o,m} \cdot P_m^{\text{proc}} \cdot CI_m}{60}$: Carbon emissions (kg CO₂)

Decision Variables:

- $x_{j,o,m} \in \{0, 1\}$: 1 if operation (j, o) is assigned to machine m
- $S_{j,o} \geq 0$: Start time of operation (j, o)
- $C_j \geq 0$: Completion time of job j
- $D_j \geq 0$: Tardiness of job j
- $C_{\max} \geq 0$: Makespan
- $y_{j_1, o_1, j_2, o_2, m} \in \{0, 1\}$: Sequencing variable for two operations sharing machine m

Mathematical Formulation

$$\min Z = w_1 \left(\frac{\sum_{(j,o) \in \mathcal{O}} \sum_{m \in M_{j,o}} x_{j,o,m} e_{j,o,m}^{\text{carbon}}}{B_1} \right) + w_2 \left(\frac{\pi \sum_{j \in J} D_j}{B_2} \right) \quad (1)$$

$$\sum_{m \in M_{j,o}} x_{j,o,m} = 1 \quad \forall (j, o) \in \mathcal{O} \quad (2)$$

$$S_{j,o+1} \geq S_{j,o} + \sum_{m \in M_{j,o}} x_{j,o,m} p_{j,o,m} \quad \forall j \in J, o = 1, \dots, O_j - 1 \quad (3)$$

$$C_j = S_{j,O_j} + \sum_{m \in M_{j,O_j}} x_{j,O_j,m} p_{j,O_j,m} \quad \forall j \in J \quad (4)$$

$$D_j \geq C_j - d_j \quad \forall j \in J \quad (5)$$

$$D_j \geq 0 \quad \forall j \in J \quad (6)$$

$$S_{j_1,o_1} + p_{j_1,o_1,m} \leq S_{j_2,o_2} + B(1 - y_{j_1,o_1,j_2,o_2,m}) \\ \forall (j_1,o_1) < (j_2,o_2), m \in M_{j_1,o_1} \cap M_{j_2,o_2} \quad (7)$$

$$S_{j_2,o_2} + p_{j_2,o_2,m} \leq S_{j_1,o_1} + B y_{j_1,o_1,j_2,o_2,m} \\ \forall (j_1,o_1) < (j_2,o_2), m \in M_{j_1,o_1} \cap M_{j_2,o_2} \quad (8)$$

$$C_{\max} \geq C_j \quad \forall j \in J \quad (9)$$

Equation (1) represents the objective function, where the first term denotes total carbon emissions and the second term denotes total tardiness penalty. The objective-scaling and weighting structure associated with B_1 , B_2 , w_1 , and w_2 is described in Subsubsection 3.1.1. Equation (2) ensures that every operation is assigned to exactly one eligible machine. Equation (3) enforces precedence among consecutive operations of the same job. Equation (4) defines the completion time of each job based on its final operation. Equations (5) and (6) define non-negative job tardiness relative to due dates. Equations (7) and (8) are the disjunctive machine-capacity constraints that prevent overlapping operations on the same machine by imposing a binary processing order. Equation (9) defines the makespan as the maximum job completion time.

3.1.1. Objective Scaling and Weighting

The optimization model combines two objectives, total carbon emissions and total tardiness penalty, into a single scalarized objective. Since these two objective components are measured in different units and may differ substantially in numerical magnitude, direct aggregation through a weighted sum can lead to scale-driven dominance of one term over the other. In such cases, the resulting objective value may be influenced more by unit magnitude than by the intended decision preference.

To address this issue, each objective is normalized using a corresponding baseline value. Let $f_1(x)$ denote total carbon emissions and $f_2(x)$ denote total tardiness penalty for a feasible schedule x . In the present formulation,

$$f_1(x) = \sum_{(j,o) \in \mathcal{O}} \sum_{m \in M_{j,o}} x_{j,o,m} e_{j,o,m}^{\text{carbon}}, \quad f_2(x) = \pi \sum_{j \in J} D_j,$$

where $f_1(x)$ represents total carbon emissions and $f_2(x)$ represents total tardiness penalty. Let $B_1 > 0$ and $B_2 > 0$ denote reference baseline values for these two objectives, respectively. The normalized objective components are then defined as

$$\tilde{f}_1(x) = \frac{f_1(x)}{B_1}, \quad \tilde{f}_2(x) = \frac{f_2(x)}{B_2}.$$

This transformation converts both objectives into dimensionless quantities, since each term is divided by a reference value expressed in the same unit. As a result, the two components become numerically comparable and can be aggregated without one objective dominating purely due to its physical unit or absolute scale.

The scalarized objective is therefore written as

$$Z = w_1 \tilde{f}_1(x) + w_2 \tilde{f}_2(x)$$

or equivalently

$$Z = w_1 \frac{f_1(x)}{B_1} + w_2 \frac{f_2(x)}{B_2},$$

where $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$. The parameters w_1 and w_2 are scalarization weights that determine the relative importance assigned to carbon emissions and tardiness penalty, respectively.

Accordingly, the normalization parameters B_1 and B_2 control comparability of scale, while the scalarization weights w_1 and w_2 encode decision preference between the two normalized objective components.

3.2. Policy-Based Rough Optimization with Large Neighborhood Search (Pro-LNS)

The Policy-based Rough Optimization with Large Neighborhood Search (Pro-LNS) framework is a two-phase methodology developed for the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T). It combines the global decision-making capability of reinforcement learning (RL) with the refinement strength of adaptive large neighborhood search (LNS), consistent with a broader direction in combinatorial optimization that integrates learning-based guidance with neighborhood-based improvement to balance exploration and intensification. Throughout both phases, all CAFJSP-T constraints, including operation precedence, machine eligibility, and machine capacity, are strictly maintained during schedule construction, modification, and repair.

In the first phase, the CAFJSP-T is represented as a Markov decision process (MDP) and solved through a learned scheduling policy. At each decision step, the policy selects a ready operation and assigns it to an eligible machine, incrementally constructing a complete feasible schedule. The reward structure follows the weighted-sum scalarization introduced in the problem formulation, so that learning reflects the joint influence of carbon emissions and tardiness penalty.

In the second phase, the RL-generated schedule is refined through an adaptive LNS procedure. A subset of operations is removed according to a criticality criterion derived from the same scalarized objective, and the removed operations are then greedily reinserted into positions that yield the greatest improvement in the composite objective value. Through repeated removal and reinsertion, the LNS phase explores a large neighborhood around the initial solution and accepts repairs that improve schedule quality.

Pro-LNS thus consists of a policy-guided construction phase followed by a neighborhood-based refinement phase, with both stages aligned to the carbon-aware and tardiness-sensitive objective of the CAFJSP-T. A detailed description of both phases is provided next, followed by an overview of the overall scheduling framework.

3.2.1. Phase I: MDP-Based Reinforcement Learning

The Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T) is modeled as a finite-horizon Markov decision process (MDP)

$$(\mathcal{S}, \mathcal{A}, T, r, H), \quad H = |\mathcal{O}|.$$

Here, H denotes the total number of operations and therefore defines the decision horizon. A policy π_θ sequentially constructs a complete feasible schedule while satisfying the CAFJSP-T constraints, including operation precedence, machine eligibility, and machine capacity.

State space:

At decision step t , the state

$$s_t = (f_t, m_t, e_t, c_t)$$

is formed by concatenating

$$f_t \in \{0, 1\}^{|\mathcal{O}|}, \quad m_t \in \mathbb{R}^{|\mathcal{M}|}, \quad e_t \in [0, 1]^{|\mathcal{O}| \times M_{\max}}, \quad c_t \in [0, 1]^{|\mathcal{O}| \times 2},$$

into a single feature vector. The state components are defined as follows:

- **Ready-flag vector:** Indicates which operations are currently eligible for dispatch:

$$f_{t,(j,o)} = \begin{cases} 1, & \text{if all predecessors of } (j,o) \text{ have been completed by step } t, \\ 0, & \text{otherwise.} \end{cases}$$

- **Machine ready-time vector:** Records the earliest time at which each machine becomes available:

$$m_{t,m} = \begin{cases} \max\{C_{j',o'} \mid (j',o') \text{ assigned to machine } m\}, & \text{if machine } m \text{ has processed} \\ & \text{at least one operation,} \\ 0, & \text{otherwise.} \end{cases}$$

- **Normalized earliest-completion-time matrix:** Estimates the completion time of each eligible operation on each eligible machine:

$$e_{t,(j,o),k} = \frac{1}{H} \left(\max\{m_{t,m_k}, C_{j,o-1}\} + p_{j,o,m_k} \right),$$

where $m_k \in M_{j,o}$ denotes the k -th eligible machine for operation (j,o) .

- **Critical-path metrics:** Encodes downstream workload and due-date slack. Let $\text{succ_time}_{j,o}$ denote the precomputed lower bound on the remaining processing time of job j from operation o onward, obtained as the sum of minimum eligible processing times of the remaining operations. Let

$$\tau_t = \min_{m \in M} m_{t,m}$$

denote the earliest machine-available time at decision step t . Then

$$c_{t,(j,o),1} = \frac{\text{succ_time}_{j,o}}{H}, \quad c_{t,(j,o),2} = \frac{\max\{0, d_j - \tau_t - \text{succ_time}_{j,o}\}}{H}.$$

Action space:

At state s_t , the agent selects an action

$$a_t = (j, o, m),$$

where operation (j,o) must be ready for processing and machine m must belong to the eligible machine set $M_{j,o}$. Thus,

$$(j,o) \in \{(j',o') \mid f_{t,(j',o')} = 1\}, \quad m \in M_{j,o}.$$

This construction guarantees that only feasible operation-machine assignments are considered.

Transition function:

Once action $a_t = (j,o,m)$ is selected, operation (j,o) is assigned to machine m at its earliest feasible start time:

$$S_{j,o} = \max\{m_{t,m}, C_{j,o-1}\}, \quad C_{j,o} = S_{j,o} + p_{j,o,m}.$$

The system state is then updated according to the new completion time:

$$m_{t+1,m} = C_{j,o},$$

$$f_{t+1,(j,o)} = 0, \quad f_{t+1,(j,o+1)} = 1 \quad \text{if operation } (j, o + 1) \text{ exists.}$$

In this way, the transition function preserves feasibility while advancing the partial schedule by one operation assignment.

Reward function:

The immediate reward is designed to reflect the baseline-scaled bi-objective structure of the CAFJSP-T formulation. For action $a_t = (j, o, m)$, the reward is defined as

$$r(s_t, a_t) = - \left[w_1 \left(\frac{e_{j,o,m}^{\text{carbon}}}{B_1} \right) + w_2 \left(\frac{\pi \Delta D_t}{B_2} \right) \right],$$

where $e_{j,o,m}^{\text{carbon}}$ is the carbon emission incurred by assigning operation (j, o) to machine m , and ΔD_t denotes the increment in cumulative tardiness at decision step t . The parameters B_1 and B_2 are the baseline values used to normalize the carbon-emission and tardiness-penalty components, while w_1 and w_2 are the corresponding scalarization weights.

In implementation, the tardiness contribution is activated when the selected action changes the completion status of a job and therefore affects its realized tardiness relative to the due date. This reward structure provides a stepwise approximation of the baseline-scaled objective defined in the mathematical formulation.

Learning objective:

Proximal Policy Optimization is used to learn the policy parameters θ by maximizing the expected cumulative return:

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{H-1} r(s_t, a_t) \right].$$

Since the reward is defined as the negative of the weighted normalized cost incurred during schedule construction, maximizing $J(\theta)$ is equivalent to minimizing the baseline-scaled combination of carbon emissions and tardiness penalty over the full schedule.

3.2.2. Phase II: Adaptive Large Neighborhood Search (LNS)

Building on the RL-generated schedule σ^0 , Phase II applies an adaptive large neighborhood search to further improve the schedule with respect to the same scalarized objective used in the CAFJSP-T formulation:

$$J(\sigma) = w_1 \left(\frac{\text{TotalCarbonEmission}(\sigma)}{B_1} \right) + w_2 \left(\frac{\text{TotalTardinessPenalty}(\sigma)}{B_2} \right).$$

Adaptive Removal:

At iteration t , let k_t denote the number of operations to remove. The removal procedure is defined as follows:

1. **Marginal-impact scoring:** For each scheduled operation (j, o) , estimate its contribution to the scalarized objective by evaluating the changes in carbon emissions and tardiness penalty associated with removing and reinserting that operation. The combined score is computed as

$$\text{score}_{j,o} = w_1 \left(\frac{\Delta \text{Carbon}_{j,o}}{B_1} \right) + w_2 \left(\frac{\Delta \text{TardinessPenalty}_{j,o}}{B_2} \right).$$

2. **Removal:** Remove the k_t operations with the highest score $\text{score}_{j,o}$, producing a partial schedule in which the most disruptive operations are unscheduled.

3. **Adaptive tuning:** Define the destroy-size bounds dynamically as

$$k_{\min} = \max\{2, \lceil \rho_{\min} |\mathcal{O}| \rceil\}, \quad k_{\max} = \min\{|\mathcal{O}| - 1, \lceil \rho_{\max} |\mathcal{O}| \rceil\},$$

where $0 < \rho_{\min} < \rho_{\max} < 1$ are predefined ratios. If reinserting the removed operations yields an improved schedule, set

$$k_{t+1} = \max(k_{\min}, k_t - 1);$$

otherwise set

$$k_{t+1} = \min(k_{\max}, k_t + 1).$$

Greedy Reinsertion:

The removed operations are reinserted one at a time while preserving feasibility:

1. **Precedence constraint:** Operation (j, o) is considered for reinsertion only after its predecessor $(j, o - 1)$, if any, has already been reinserted.
2. **Feasible start times:** For each eligible machine $m \in M_{j,o}$, compute

$$s_{j,o}(m) = \max\{\text{ready_time}(m), C_{j,o-1}\}, \quad c_{j,o}(m) = s_{j,o}(m) + p_{j,o,m}.$$

3. **Objective-based choice:** For each eligible machine, evaluate $\Delta J(m)$, the increase in the scalarized objective if (j, o) is inserted on machine m . The operation is then assigned to the machine

$$m^* = \arg \min_{m \in M_{j,o}} \Delta J(m).$$

Acceptance and Adaptation:

Let σ^{t+1} denote the schedule obtained after reinsertion. If

$$J(\sigma^{t+1}) < J(\sigma^t),$$

then σ^{t+1} is accepted as the new incumbent schedule and the removal size is reduced according to

$$k_{t+1} = \max(k_{\min}, k_t - 1).$$

Otherwise, the candidate schedule is rejected and the removal size is increased according to

$$k_{t+1} = \min(k_{\max}, k_t + 1).$$

This mechanism balances intensification and diversification during the search.

Termination:

The remove–reinsert–accept procedure continues until convergence is observed, defined as the absence of improvement in the scalarized objective over a prescribed number of iterations. The final schedule σ^* therefore represents an LNS-refined improvement over the RL-generated initial solution with respect to the baseline-scaled combination of carbon emissions and tardiness penalty.

3.2.3. Policy-Based Rough Optimization with Large Neighborhood Search

The complete Pro-LNS procedure is depicted in detail in Algorithm 1.

Algorithm 1 Policy-based Rough Optimization Neighborhood Search (Pro-LNS)**Initialization:**

Load job data $\mathcal{J}, \mathcal{O}_j, \mathcal{M}_{j,o}$ and machine parameters
 Initialize PPO policy π_θ with padded observation space
 Set adaptive LNS bounds k_{\min}, k_{\max} and initial $k \leftarrow k_{\text{init}}$
 Set stagnation threshold $\tau \leftarrow \lambda \cdot |\mathcal{O}|$
 Initialize empty schedule $\sigma^0 \leftarrow \emptyset$

Phase 1: MDP-based Construction

Objective: Maximize the expected return $J(\theta) = \sum_{t=0}^{H-1} r(s_t, a_t)$, where

$$r(s_t, a_t) = - \left[w_1 \left(\frac{e_{j^*, o^*, m^*}^{\text{carbon}}}{B_1} \right) + w_2 \left(\frac{\pi \Delta D_t}{B_2} \right) \right]$$

with $e_{j^*, o^*, m^*}^{\text{carbon}} = \frac{p_{j^*, o^*, m^*}^{\text{Proc}} C I_{m^*}}{60}$ and ΔD_t denoting the increment in tardiness penalty at step t

while $\exists (j, o) \notin \sigma^t$, enforcing: **do**

- Machine eligibility: $m^* \in \mathcal{M}_{j^*, o^*}$
- Precedence: $(j^*, o^* - 1)$ must be scheduled before (j^*, o^*)
- Capacity: $S_{j^*, o^*} \geq \max(m_{t, m^*}, C_{j^*, o^* - 1})$

State Observation:

Ready ops $\mathcal{R} \leftarrow \{(j, o) \notin \sigma^t \mid o = 1 \vee (j, o - 1) \in \sigma^t\}$

Extract features:

- Operation flags $f_t \in \{0, 1\}^{|\mathcal{O}|}$
- Machine ready times $m_t \in \mathbb{R}^M$
- Normalized ECT matrix $e_t \in [0, 1]^{|\mathcal{O}| \times M_{\max}}$
- Critical-path metrics $c_t \in [0, 1]^{|\mathcal{O}| \times 2}$

Note: All time-based features are normalized by H .

Action: Sample feasible $(j^*, o^*, m^*) \sim \pi_\theta(s_t)$ ($(j, o) \in \mathcal{R}, m^* \in \mathcal{M}_{j^*, o^*}$)

Schedule Update:

$S_{j^*, o^*} \leftarrow \max(m_{t, m^*}, C_{j^*, o^* - 1})$
 $C_{j^*, o^*} \leftarrow S_{j^*, o^*} + p_{j^*, o^*, m^*}$
 Update $\sigma^{t+1} \leftarrow \sigma^t \cup \{(j^*, o^*, m^*, S_{j^*, o^*}, C_{j^*, o^*})\}$
 Update $m_{t+1, m^*} \leftarrow C_{j^*, o^*}$

Phase 1 Output: $\sigma^0 \leftarrow \sigma^t$

Phase 2: Adaptive LNS Refinement

$\sigma^* \leftarrow \sigma^0, J^* \leftarrow J(\sigma^0)$

noImprovementCount $\leftarrow 0$

while noImprovementCount $< \tau$ **do**

Destroy:

For each $(j, o) \in \sigma^*$, compute:

$$\text{score}_{j,o} = w_1 \left(\frac{\Delta \text{Carbon}_{j,o}}{B_1} \right) + w_2 \left(\frac{\Delta \text{TardinessPenalty}_{j,o}}{B_2} \right)$$

Select $\mathcal{D} \leftarrow$ Top- k operations by $\text{score}_{j,o}$

Repair:

For each $(j, o) \in \mathcal{D}$ in precedence order:

- Wait until $(j, o - 1)$ is scheduled
- For each $m \in \mathcal{M}_{j,o}$:
 $S_{j,o}(m) \leftarrow \max(\text{ready_time}(m), C_{j,o-1})$
 $C_{j,o}(m) \leftarrow S_{j,o}(m) + p_{j,o,m}$
- Select $m^* = \arg \min_m \Delta J(m)$
- Insert (j, o) on m^* at $S_{j,o}(m^*)$

Evaluate:

Compute $J(\sigma')$

If $J(\sigma') < J^*$ **then**

$\sigma^* \leftarrow \sigma', J^* \leftarrow J(\sigma')$
 $k \leftarrow \max(k_{\min}, k - 1)$
 noImprovementCount $\leftarrow 0$

else

$k \leftarrow \min(k_{\max}, k + 1)$
 noImprovementCount \leftarrow noImprovementCount + 1

end if

Output:

Return σ^* together with $J(\sigma^*)$

3.2.4. RL Architecture and Training Protocol

This subsection describes the reinforcement learning component of the proposed Policy-based Rough Optimization with Large Neighborhood Search (Pro-LNS) framework. The RL agent is implemented using Proximal Policy Optimization (PPO) and is responsible for constructing an initial

feasible schedule for the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T).

RL Architecture.

The scheduling problem is modeled as a finite-horizon Markov decision process in which the policy sequentially assigns operations to eligible machines while respecting precedence and machine-capacity constraints. To accommodate heterogeneous instance sizes, all state representations are embedded into a fixed-dimensional observation space through neural padding. The largest training instance is used only to determine the required maximum dimensionality, allowing all training and benchmark instances to be processed without modifying the network architecture [60].

The policy is implemented in Stable-Baselines3 [61] using a Multi-Layer Perceptron (MLP) architecture. Both the policy and value networks use two hidden layers of size 256 with the Rectified Linear Unit (ReLU) activation. PPO optimization is based on the clipped surrogate objective with mean-squared-error value loss, entropy regularization, and per-batch advantage normalization.

Training Protocol.

To improve robustness across structurally diverse scheduling environments, PPO is trained on a pool of 20 synthetically generated large-scale CAFJSP-T instances rather than on a single fixed instance. These training instances are deliberately more demanding than the benchmark cases and contain 90–110 jobs, 54–66 machines, five operations per job, 3–62 eligible machines per operation, and processing times ranging from 5 to 35 minutes. This training strategy biases the policy toward structurally difficult scheduling conditions and supports generalization to smaller or less dense benchmark instances [62,63].

Training is conducted for a fixed budget of 500,000 timesteps across 8 parallel vectorized environments using a learning rate of 5×10^{-5} , entropy coefficient 0.001, mini-batch size 64, and 10 optimization epochs per update. The remaining PPO parameters follow the default Stable-Baselines3 configuration, including discount factor 0.99, Generalized Advantage Estimation (GAE) parameter 0.95, clipping range 0.2, value-function coefficient 0.5, and maximum gradient norm 0.5 [61]. The trained policy is saved once and reused for all benchmark evaluations without further fine-tuning.

Convergence is assessed empirically through the episodic reward trajectory recorded during training. As shown in Figure 1, returns stabilize in the later stages of training, supporting the use of the fixed training budget as a practical stopping criterion.

PPO Technical Details.

The principal PPO configuration is summarized below:

- **Policy type:** Multi-Layer Perceptron (MLP)
- **Hidden layers:** [256, 256]
- **Activation:** ReLU
- **Learning rate:** 5×10^{-5}
- **Entropy coefficient:** 0.001
- **Mini-batch size:** 64
- **Total training timesteps:** 500,000
- **Number of parallel environments:** 8
- **Discount factor (γ):** 0.99
- **GAE parameter (λ):** 0.95
- **Clipping range:** 0.2
- **Optimization epochs per update:** 10
- **Value-function coefficient:** 0.5
- **Maximum gradient norm:** 0.5

- **Loss function:** Clipped Surrogate Objective (PPO) with Mean Squared Error (MSE) Value Loss and Entropy Regularization
- **Advantage normalization:** True (per-batch)

These settings were selected based on standard PPO implementation practices in Stable-Baselines3 and prior reinforcement-learning studies on large-scale and computationally demanding decision environments [61,64–68].

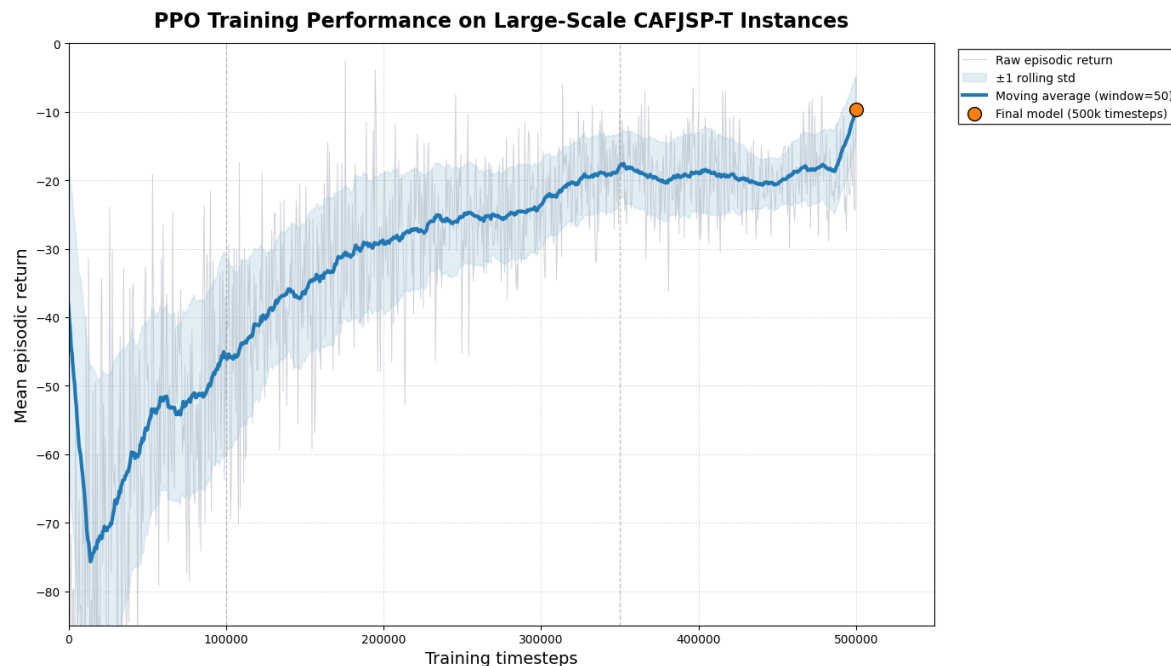


Figure 1. Training Performance - Mean Episodic Return over 500k timesteps

3.3. Benchmark Instances and Experimental Setup

The computational study was based on 16 benchmark instances from Behnke and Geiger [69], covering small, medium, and large workcenter (WC) configurations. Because these instances do not include sustainability-related parameters, they were extended using the environmental and due-date settings reported by Lu et al. [70]. Specifically, machine processing power values were sampled from [10, 20] kW, machine idle power values were sampled from [1, 3] kW, carbon intensity was fixed at 0.998 kgCO₂/kWh, the due-date tightness parameter was taken from $\theta \in [0.5, 1.5]$, and the tardiness penalty was set to 0.1 per minute of tardiness.

Using these settings, due dates were generated as

$$d_j = \theta \cdot \sum_{o \in \mathcal{O}_j} \max_{m \in M_{j,o}} p_{j,o,m}, \quad \forall j \in J.$$

This construction links each job's due date to its processing requirements while allowing the degree of due-date tightness to vary across problem instances.

To support the scalarized objective formulation, representative baseline values were computed for objective normalization using single-objective Genetic Algorithm runs [71,72]. One representative instance was selected for each benchmark category, and the resulting baseline values are reported in Table 1. For each experiment, the normalization constants were applied according to the category of the benchmark instance under study, namely Sm for small WC instances, Med for medium WC instances, and Lar for large WC instances.

Table 1. Baseline values used for objective normalization by benchmark category

Category	Representative instance	Carbon baseline B_1 (kg CO ₂)	Tardiness penalty baseline B_2
Sm	sm04_5	1869.3974	4425.480
Med	med04_5	2005.0144	2878.902
Lar	lar04_5	1820.9377	2651.852

The experimental analysis consisted of two components:

- **Benchmark-based warm-start evaluation:** The proposed Pro-LNS framework was applied to the full set of benchmark instances. For each instance, the final Pro-LNS solution was used to warm-start the MILP formulation of the same CAFJSP-T instance by providing it to the solver as an initial incumbent. The MILP solver was then run on the same instance to obtain a best bound and the corresponding optimality gap. This procedure was used to evaluate the quality of the Pro-LNS solution relative to the exact formulation and to quantify how close the final Pro-LNS schedule was to proven optimality within the allotted MILP solve time.
- **Weight-sensitivity analysis:** A weight-sensitivity analysis was conducted to examine how the schedule changes under different objective-function priorities. By varying the scalarization weights assigned to carbon emissions and tardiness penalty, the analysis was used to study how the resulting schedules respond to different relative priorities between the two objective components. This analysis illustrates the effect of the weighted objective structure on scheduling decisions.

4. Results

This section first reports the results of the *Benchmark-based warm-start evaluation* under an equal-weight scalarization setting, where carbon emissions and tardiness penalty are assigned equal importance in the objective function ($w_1 = 0.5$, $w_2 = 0.5$). The purpose of this experiment is to evaluate the quality of the final Policy-based Rough Optimization with Large Neighborhood Search (Pro-LNS) solutions across the benchmark set and to assess those solutions relative to the exact mixed-integer linear programming (MILP) formulation through warm-started optimality-gap information.

Table 2 reports the results of the benchmark-based warm-start evaluation under equal objective weighting. Several positive findings emerge from these results:

- **Pro-LNS delivers strong due-date performance on a substantial portion of the benchmark set.** Zero tardiness is achieved on sm01_1, sm01_3, med01_2, and lar01_1, and tardiness remains very small on med02_1, med02_5, and lar02_3. Thus, in 7 of the 15 reported instances, Pro-LNS produces schedules with either zero tardiness or only negligible delay, while still controlling carbon emissions under the same equal-weight objective.
- **The optimality-gap results indicate that the final Pro-LNS solutions are highly competitive with respect to the exact MILP formulation.** Across the benchmark instances reported in Table 2, the median optimality gap of 6.12%, and the maximum gap is 13.67%. Moreover, 11 of the 15 instances remain within a 10% optimality gap, and all reported instances remain within 14%. Given that these gaps are computed on the same constrained Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T) formulation after warm-starting the MILP solver with the final Pro-LNS solution, these values provide strong evidence that Pro-LNS produces high-quality incumbent solutions.
- **The method remains computationally efficient across all benchmark categories.** The average CPU time is 4.08 seconds, and the maximum reported CPU time is 10.51 seconds. This means that Pro-LNS is able to return competitive schedules with bounded optimality gaps in only a few seconds, which is especially valuable for complex flexible job shop environments where exact methods alone can become computationally burdensome.
- **Pro-LNS preserves balanced performance under equal objective weighting.** Even in instances where tardiness becomes more pronounced, the method continues to return feasible schedules

with controlled carbon emissions, reasonable makespans, and moderate optimality gaps. This indicates that Pro-LNS does not sacrifice one objective uncontrollably in order to improve the other, but instead maintains a balanced trade-off structure under the equal-weight formulation.

- **From a managerial perspective, the results suggest that Pro-LNS is well suited for practical production planning in settings where sustainability and delivery reliability must be addressed together.** The combination of low runtimes, controlled emissions, and relatively tight optimality gaps means that decision-makers can obtain strong schedules quickly, while still retaining confidence that the solutions are close to the benchmark provided by the exact formulation. This is particularly useful in operational environments where schedules may need to be generated or updated repeatedly within limited planning time.

Table 2. Benchmark-based warm-start evaluation results

Instance	Carbon (kg CO ₂)	Tardiness	Energy (kWh)	Makespan (minutes)	CPU (s)	Optimality Gap (%)
sm01_1	140.70	0.00	140.98	157.00	0.81	2.87
sm01_3	142.32	0.00	142.61	159.00	0.80	2.92
sm02_2	297.30	17.26	297.90	217.00	1.41	4.23
sm03_1	739.73	478.14	741.21	428.00	3.62	7.56
sm04_5	1499.46	3093.06	1502.46	864.00	8.55	11.34
med01_2	145.73	0.00	146.03	148.00	0.52	2.18
med02_1	297.84	2.07	298.44	160.00	1.59	4.67
med02_5	302.71	5.10	303.32	173.00	1.67	5.89
med03_3	773.86	246.11	775.41	286.00	4.43	8.92
med04_5	1580.71	1685.21	1583.88	589.00	10.51	12.78
lar01_1	142.18	0.00	142.47	122.00	1.00	2.43
lar02_3	272.26	0.07	272.81	177.00	1.33	6.12
lar03_2	691.81	98.43	693.19	283.00	5.57	9.45
lar04_1	1387.07	1170.23	1389.85	506.00	9.39	13.21
lar04_5	1438.07	1180.54	1440.95	503.00	9.95	13.67

The benchmark-based warm-start evaluation establishes that Pro-LNS performs effectively under an equal-weight objective setting. Since this case represents only one particular preference structure within the CAFJSP-T formulation, it is also important to examine how the method responds when the relative importance of carbon emissions and tardiness penalty is varied. To this end, a *weight-sensitivity analysis* was conducted on instance sm02_3, and the corresponding results are reported in Table 3.

Table 3. Weight-sensitivity analysis results for instance sm02_3 under varying scalarization weights

Weight	Carbon Emissions (kg CO ₂)	Tardiness	Energy Consumption (kWh)	Makespan (minutes)	CPU (s)	Optimality Gap (%)
$w_1 = 1.0, w_2 = 0.0$	278.3329	44.82	278.8907	262	1.07	4.87
$w_1 = 0.8, w_2 = 0.2$	281.3272	33.76	281.8910	233	1.07	4.52
$w_1 = 0.7, w_2 = 0.3$	280.4681	45.38	281.0302	256	1.10	4.91
$w_1 = 0.5, w_2 = 0.5$	287.2771	9.04	287.8528	196	1.77	4.23
$w_1 = 0.3, w_2 = 0.7$	289.3713	10.24	289.9512	208	1.17	4.35
$w_1 = 0.2, w_2 = 0.8$	280.1544	13.04	280.7158	206	1.03	4.41
$w_1 = 0.0, w_2 = 1.0$	286.8900	7.88	287.4600	183	1.04	4.19

Table 3 shows that the main effect of changing the scalarization weights is expressed through tardiness penalty, which varies much more sharply than carbon emissions across the tested settings. Carbon emissions remain within a relatively narrow range, from 278.3329 to 289.3713 kg CO₂, whereas tardiness penalty ranges from 7.88 to 45.38, indicating that scheduling performance is more sensitive to weight changes on the due-date side than on the environmental side. This makes the analysis especially useful for identifying policy settings that preserve good performance on the secondary objective while prioritizing the primary one. On the carbon-priority side, $w_1 = 0.8, w_2 = 0.2$ (Figure 2) is preferable to $w_1 = 0.7, w_2 = 0.3$, since carbon emissions remain very close (281.3272 vs. 280.4681 kg CO₂) while tardiness penalty improves substantially (33.76 vs. 45.38), with corresponding improvements in makespan (233 vs. 256 minutes) and optimality gap (4.52% vs. 4.91%). On the tardiness-priority side, $w_1 = 0.2, w_2 = 0.8$ (Figure 3) is a stronger compromise than $w_1 = 0.3, w_2 = 0.7$, because it lowers carbon emissions substantially (280.1544 vs. 289.3713 kg CO₂) while increasing tardiness penalty only moderately (13.04 vs. 10.24), with makespan remaining nearly unchanged (206 vs. 208 minutes). If due-date performance is the dominant priority, $w_1 = 0.0, w_2 = 1.0$ (Figure 4) provides the best service-oriented outcome, yielding the lowest tardiness penalty and shortest makespan. Overall, the results suggest that suitable managerial configurations are those that prioritize one objective while

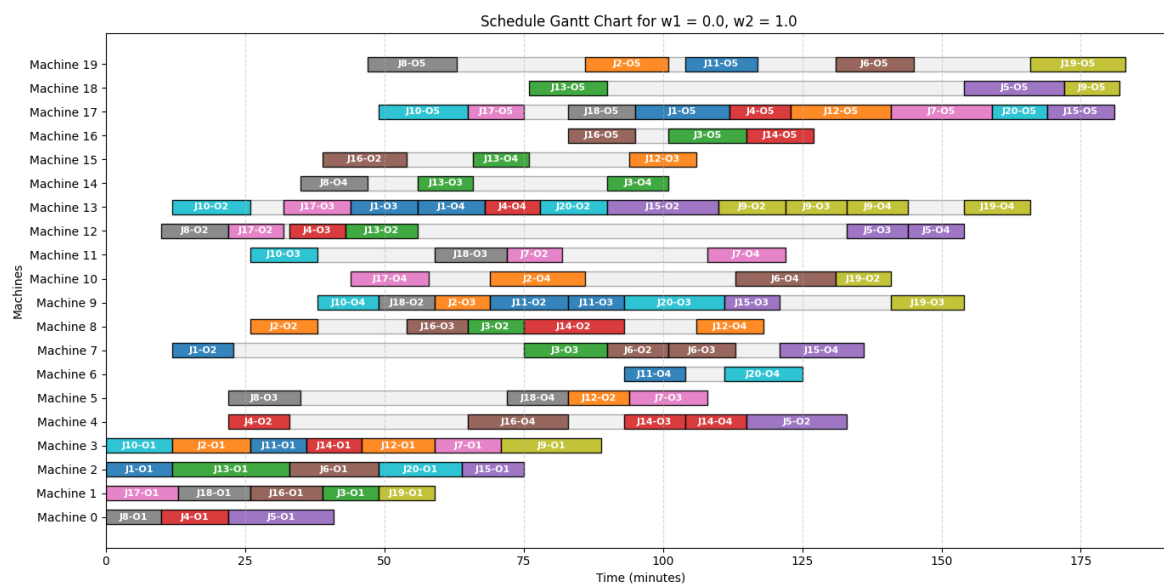


Figure 4. Detailed schedule visualization for instance sm02_3 under tardiness only priority weighting ($w_1 = 0.0, w_2 = 1.0$), highlighting how the production sequence shifts when delivery performance is the only priority.

5. Conclusion

This paper presented the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T) and proposed a Policy-based Rough Optimization with Large Neighborhood Search (Pro-LNS) as a hybrid solution framework that combines Proximal Policy Optimization (PPO)-based policy learning with large neighborhood search. The study shows that carbon-aware and tardiness-sensitive scheduling can be handled within a unified decision framework in which environmental and service objectives are treated jointly rather than as separate planning concerns. Beyond algorithmic performance, the main implication of the proposed formulation is managerial: the scalarization weights can serve as a practical policy mechanism for translating operational priorities into scheduling behavior, allowing firms to emphasize sustainability, delivery reliability, or a balanced compromise depending on production context. This makes Pro-LNS relevant not only as a computational method, but also as a decision-support approach for sustainable manufacturing planning.

At the same time, the present study is subject to some limitations. The formulation assumes a deterministic and steady-state production environment, with continuously available machines, non-preemptive processing, fixed due dates, constant processing-time-based carbon estimation, and no machine failures, dynamic job arrivals, sequence-dependent setup times, or worker-related constraints. It also does not account for transient machine states or time-varying carbon intensity. Future research should therefore extend the framework to more realistic shop-floor settings by incorporating uncertainty, dynamic arrivals and disruptions, richer machine energy-state behavior, and additional operational features such as setup effects, transportation interactions, and other integrated production constraints. Such extensions would improve both the realism of the CAFJSP-T model and the practical applicability of Pro-LNS in industrial environments.

Acknowledgments: The authors would like to thank Wichita State University for providing access to the high-performance computing (HPC) cluster, which was instrumental in conducting the computational experiments for this research.

Author Contributions: Saurabh Sanjay Singh contributed to conceptualization, methodology, formal analysis, investigation, data curation, visualization, writing—original draft, and writing—review and editing. Deepak Gupta contributed to supervision, validation, project administration, resources, and writing—review and editing. All authors reviewed the results and approved the final version of the manuscript.

Funding: This work was funded in part by the U.S. Department of Energy's (DOE) Office of Manufacturing and Energy Supply Chains through the Industrial Training and Assessment Center (ITAC) program.

Data Availability Statement: The study utilizes publicly available benchmark instances from Behnke and Geiger [69].

Conflicts of Interest: The authors declare no conflicts of interest to report regarding the study.

References

1. Zeying Li, Saad Rasool, Mustafa Fedai Cavus, and Waseem Shahid. "Sustaining the future: How green capabilities and digitalization drive sustainability in modern business." *Heliyon*, 10, 2024. <https://doi.org/10.1016/j.heliyon.2024.e24158>.
2. Mohamed El Mokadem and Magdy Khalaf. "Building sustainable performance through green supply chain management." *International Journal of Productivity and Performance Management*, 2024. <https://doi.org/10.1108/ijppm-02-2024-0113>.
3. Atif Sattar Mahar, Yang Zhang, Burhan Sadiq, and Rana Faizan Gul. "Sustainability Transformation Through Green Supply Chain Management Practices and Green Innovations in Pakistan's Manufacturing and Service Industries." *Sustainability*, 17(5), 2025. <https://doi.org/10.3390/su17052204>.
4. Mengmeng Wang and Guocheng Zhang. "What motivates firms to adopt a green supply chain and how much does it matter?" *Frontiers in Environmental Science*, 2023. <https://doi.org/10.3389/fenvs.2023.1227008>.
5. A. Poggi, L. Di Persio, and M. Ehrhardt. "Electricity Price Forecasting via Statistical and Deep Learning Approaches: The German Case." *AppliedMath*, 3(2):316–342, 2023. <https://doi.org/10.3390/appliedmath3020018>.
6. Ganesh Narkhede, Satish Chinchani, Rupesh Narkhede, and Tansen Chaudhari. "Role of Industry 5.0 for driving sustainability in the manufacturing sector: an emerging research agenda." *Journal of Strategy and Management*, 2024. <https://doi.org/10.1108/jsma-06-2023-0144>.
7. Morteza Ghobakhloo, M. Iranmanesh, B. Feroz, Erfan Babaei Tirkolaee, S. Asadi, and A. Amran. "Industry 5.0 implications for inclusive sustainable manufacturing: An evidence-knowledge-based strategic roadmap." *Journal of Cleaner Production*, 2023. <https://doi.org/10.1016/j.jclepro.2023.138023>.
8. Yimin Tang, Lihong Shen, and Shuguang Han. "Low-Carbon Flexible Job Shop Scheduling Problem Based on Deep Reinforcement Learning." *Sustainability*, 16(11), 2024. <https://doi.org/10.3390/su16114544>.
9. S. Aghakhani and M.S. Rajabi. "A New Hybrid Multi-Objective Scheduling Model for Hierarchical Hub and Flexible Flow Shop Problems." *AppliedMath*, 2(4):721–737, 2022. <https://doi.org/10.3390/appliedmath2040043>.
10. Candice Destouet, Houda Tlahig, B. Bettayeb, and B. Mazari. "Flexible job shop scheduling problem under Industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement." *Journal of Manufacturing Systems*, 2023. <https://doi.org/10.1016/j.jmsy.2023.01.004>.
11. Qingshan Gong, Junlin Li, Zhigang Jiang, and Yan Wang. "A hierarchical integration scheduling method for flexible job shop with green lot splitting." *Engineering Applications of Artificial Intelligence*, 129:107595, 2024. <https://doi.org/10.1016/j.engappai.2023.107595>.
12. A. Mencaroni, P. Leyman, B. Raa, S. De Vuyst, and D. Claeys. "Towards net-zero manufacturing: Carbon-aware scheduling for GHG emissions reduction." *Journal of Cleaner Production*, 529:146787, 2025. <https://doi.org/10.1016/j.jclepro.2025.146787>.
13. G. P. Georgiadis, C. N. Dimitriadis, and M. C. Georgiadis. "Decarbonizing the Industry Sector: Current Status and Future Opportunities of Energy-Aware Production Scheduling." *Processes*, 13:1941, 2025. <https://doi.org/10.3390/pr13061941>.
14. J. T. Naidu. "A New Algorithm for the Weighted Tardiness Problem." *Journal of Applied Business & Economics*, 27(5), 2025. <https://doi.org/10.33423/jabe.v27i5.7833>.
15. B. de Athayde Prata, L. R. de Abreu, and V. Fernandez-Viagas. "A systematic review of permutation flow shop scheduling with due-date-related objectives." *Computers & Operations Research*, 106989, 2025. <https://doi.org/10.1016/j.cor.2025.106989>.
16. F. Xiong, S. Chen, N. Xiong, and L. Jing. "Scheduling distributed heterogeneous non-permutation flowshop to minimize the total weighted tardiness." *Expert Systems with Applications*, 272:126713, 2025. <https://doi.org/10.1016/j.eswa.2025.126713>.
17. M. I. Ulucak and H. Gökçen. "Dynamic Scheduling in Identical Parallel-Machine Environments: A Multi-Purpose Intelligent Utility Approach." *Applied Sciences*, 15(5):2483, 2025. <https://doi.org/10.3390/app15052483>.
18. L. Meng, W. Cheng, B. Zhang, W. Zou, and P. Duan. "A novel hybrid algorithm of genetic algorithm, variable neighborhood search and constraint programming for distributed flexible job shop scheduling problem." *International Journal of Industrial Engineering Computations*, 2024. <https://doi.org/10.5267/j.ijiec.2024.3.001>.

19. S. Nessari, R. Tavakkoli-Moghaddam, H. Bakhshi-Khaniki, and A. Bozorgi-Amiri. "A hybrid simheuristic algorithm for solving bi-objective stochastic flexible job shop scheduling problems." *Decision Analytics Journal*, 2024. <https://doi.org/10.1016/j.dajour.2024.100485>.
20. J. C. Seck-Tuoh-Mora, N. J. Escamilla-Serna, L. J. Montiel-Arrieta, I. Barragán-Vite, and J. Medina-Marín. "A Global Neighborhood with Hill-Climbing Algorithm for Fuzzy Flexible Job Shop Scheduling Problem." *Mathematics*, 10(22), 2022. <https://doi.org/10.3390/math10224233>.
21. L. Berterottière, S. Dauzère-Pérés, and C. Yugma. "Flexible job-shop scheduling with transportation resources." *European Journal of Operational Research*, 312(3):890–909, 2023. <https://doi.org/10.1016/j.ejor.2023.07.036>.
22. Z. Li and Y.-H. Chen. "Minimizing the makespan and carbon emissions in the green flexible job shop scheduling problem with learning effects." *Scientific Reports*, 13, 2023. <https://doi.org/10.1038/s41598-023-33615-z>.
23. S. Jia, Y. Yang, S. Li, S. Wang, A. Li, W. Cai, Y. Liu, J. Hao, and L. Hu. "The Green Flexible Job-Shop Scheduling Problem Considering Cost, Carbon Emissions, and Customer Satisfaction under Time-of-Use Electricity Pricing." *Sustainability*, 16(6), 2024. <https://doi.org/10.3390/su16062443>.
24. G. Xu, Q. Bao, and H. Zhang. "Multi-objective green scheduling of integrated flexible job shop and automated guided vehicles." *Engineering Applications of Artificial Intelligence*, 126:106864, 2023. <https://doi.org/10.1016/j.engappai.2023.106864>.
25. H. Tang, J. Huang, C. Ren, Y. Shao, and J. Lu. "Integrated scheduling of multi-objective lot-streaming hybrid flowshop with AGV based on deep reinforcement learning." *International Journal of Production Research*, 63(4):1275–1303, 2024. <https://doi.org/10.1080/00207543.2024.2373426>.
26. M. Füchtenhans and C. Glock. "The impact of incentive-based programmes on job-shop scheduling with variable machine speeds." *International Journal of Production Research*, 62(12):4546–4564, 2023. <https://doi.org/10.1080/00207543.2023.2266765>.
27. M.-J. Park and A. Ham. "Energy-aware flexible job shop scheduling under time-of-use pricing." *International Journal of Production Economics*, 2022. <https://doi.org/10.1016/j.ijpe.2022.108507>.
28. Y. Chen, X. Liao, G. Chen, and Y. Hou. "Dynamic Intelligent Scheduling in Low-Carbon Heterogeneous Distributed Flexible Job Shops with Job Insertions and Transfers." *Sensors*, 24(7), 2024. <https://doi.org/10.3390/s24072251>.
29. Zhixue Wang, Maowei He, Ji Wu, Hanning Chen, and Yang Cao. "An improved MOEA/D for low-carbon many-objective flexible job shop scheduling problem." *Computers & Industrial Engineering*, 188:109926, 2024. <https://doi.org/10.1016/j.cie.2024.109926>.
30. Y. Xiao, S. Yin, G. Ren, and W. Liu. "Study on flexible job shop scheduling problem considering energy saving." *Journal of Intelligent & Fuzzy Systems*, 46:5493–5520, 2024. <https://doi.org/10.3233/JIFS-233337>.
31. W. Peng, D. Yu, and F. Xie. "Multi-mode resource-constrained project scheduling problem with multiple shifts and dynamic energy prices." *International Journal of Production Research*, 63(7):2483–2506, 2024. <https://doi.org/10.1080/00207543.2024.2403774>.
32. Z. Wei, W. Liao, and L. Zhang. "Hybrid energy-efficient scheduling measures for flexible job-shop problem with variable machining speeds." *Expert Systems with Applications*, 197:116785, 2022. <https://doi.org/10.1016/j.eswa.2022.116785>.
33. N. Rakovitis, D. Li, N. Zhang, J. Li, L. Zhang, and X. Xiao. "Novel Approach to Energy-Efficient Flexible Job-Shop Scheduling Problems." *Energy*, 2021. <https://doi.org/10.1016/j.energy.2021.121773>.
34. Y. Sang and J. Tan. "Many-Objective Flexible Job Shop Scheduling Problem with Green Consideration." *Energies*, 15(5), 2022. <https://doi.org/10.3390/en15051884>.
35. R. Li, W. Gong, L. Wang, C. Lu, and S. Jiang. "Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time." *Swarm and Evolutionary Computation*, 74:101139, 2022. <https://doi.org/10.1016/j.swevo.2022.101139>.
36. D. Lei, Y. Zheng, and X. Guo. "A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption." *International Journal of Production Research*, 55(11):3126–3140, 2016. <https://doi.org/10.1080/00207543.2016.1262082>.
37. T. Jiang, H. Zhu, and G. Deng. "Improved African buffalo optimization algorithm for the green flexible job shop scheduling problem considering energy consumption." *Journal of Intelligent & Fuzzy Systems*, 38:4573–4589, 2020. <https://doi.org/10.3233/JIFS-191370>.

38. Z. Peng, H. Zhang, H. Tang, Y. Feng, and W. Yin. "Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect." *Journal of Intelligent Manufacturing*, 33:1725–1746, 2021. <https://doi.org/10.1007/s10845-020-01713-8>.
39. W. Ren, J. Wen, Y. Yan, Y. Hu, Y. Guan, and J. Li. "Multi-objective optimisation for energy-aware flexible job-shop scheduling problem with assembly operations." *International Journal of Production Research*, 59(23):7216–7231, 2020. <https://doi.org/10.1080/00207543.2020.1836421>.
40. W. Song, X. Chen, Q. Li, and Z. Cao. "Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning." *IEEE Transactions on Industrial Informatics*, 19(2):1600–1610, 2023. <https://doi.org/10.1109/TII.2022.3189725>.
41. K. Lei, P. Guo, W. Zhao, Y. Wang, L. Qian, X. Meng, and L. Tang. "A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem." *Expert Systems with Applications*, 205:117796, 2022. <https://doi.org/10.1016/j.eswa.2022.117796>.
42. R. Liu, R. Piplani, and C. Toro. "Deep reinforcement learning for dynamic scheduling of a flexible job shop." *International Journal of Production Research*, 60(13):4049–4069, 2022. <https://doi.org/10.1080/00207543.2022.2058432>.
43. W. Yi, N. Chen, Y. Chen, and Z. Pei. "An improved deep Q-network for dynamic flexible job shop scheduling with limited maintenance resources." *International Journal of Production Research*, pages 1–22, 2025. <https://doi.org/10.1080/00207543.2025.2533528>.
44. J.-P. Huang, L. Gao, and X. Li. "An end-to-end deep reinforcement learning method based on graph neural network for distributed job-shop scheduling problem." *Expert Systems with Applications*, 238:121756, 2023. <https://doi.org/10.1016/j.eswa.2023.121756>.
45. Y. Lei, Q. Deng, M. Liao, and S. Gao. "Deep reinforcement learning for dynamic distributed job shop scheduling problem with transfers." *Expert Systems with Applications*, 251:123970, 2024. <https://doi.org/10.1016/j.eswa.2024.123970>.
46. S. Yin and Z. Xiang. "A hyper-heuristic algorithm via proximal policy optimization for multi-objective truss problems." *Expert Systems with Applications*, 256:124929, 2024. <https://doi.org/10.1016/j.eswa.2024.124929>.
47. L. van Hezewijk, N. Dellaert, T. Van Woensel, and N. Gademann. "Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem." *International Journal of Production Research*, 61(6):1955–1978, 2022. <https://doi.org/10.1080/00207543.2022.2056540>.
48. F. Zhang, R. Li, and W. Gong. "Deep reinforcement learning-based memetic algorithm for energy-aware flexible job shop scheduling with multi-AGV." *Computers & Industrial Engineering*, 189:109917, 2024. <https://doi.org/10.1016/j.cie.2024.109917>.
49. J. Shi, W. Liu, and J. Yang. "An Enhanced Multi-Objective Evolutionary Algorithm with Reinforcement Learning for Energy-Efficient Scheduling in the Flexible Job Shop." *Processes*, 12(9):1976, 2024. <https://doi.org/10.3390/pr12091976>.
50. R. Li, W. Gong, and C. Lu. "A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling." *Expert Systems with Applications*, 203:117380, 2022. <https://doi.org/10.1016/j.eswa.2022.117380>.
51. S. S. Singh, R. Joshi, and D. Gupta, "An Advantage Actor-Critic Approach for Energy-Conscious Scheduling in Flexible Job Shops," *J. Artif. Intell.*, vol. 7, no. 1, pp. 177–203, 2025. <https://doi.org/10.32604/jai.2025.065078>.
52. W. Shao, Z. Shao, and D. Pi. "A multi-neighborhood-based multi-objective memetic algorithm for the energy-efficient distributed flexible flow shop scheduling problem." *Neural Computing and Applications*, 34:22303–22330, 2022. <https://doi.org/10.1007/s00521-022-07714-3>.
53. B. Zhang and A. Che. "An enhanced decomposition-based multi-objective evolutionary algorithm with neighborhood search for multi-resource constrained job shop scheduling problem." *Swarm and Evolutionary Computation*, 93:101834, 2025. <https://doi.org/10.1016/j.swevo.2024.101834>.
54. I. Smit, J. Zhou, R. Reijnen, Y. Wu, J. Chen, C. Zhang, Z. Bukhsh, W. Nuijten, and Y. Zhang. "Graph Neural Networks for Job Shop Scheduling Problems: A Survey." *arXiv*, abs/2406.14096, 2024. <https://doi.org/10.48550/arxiv.2406.14096>.
55. Zixiao Pan, Ling Wang, Jing-jing Wang, and Jiawen Lu. "Deep Reinforcement Learning Based Optimization Algorithm for Permutation Flow-Shop Scheduling." *IEEE Transactions on Emerging Topics in Computational Intelligence*, 7:983–994, 2023. <https://doi.org/10.1109/TETCI.2021.3098354>.
56. M. Khadivi, T. Charter, M. Yaghoubi, M. Jalayer, M. Ahang, A. Shojaeinasab, and H. Najjaran. "Deep reinforcement learning for machine scheduling: Methodology, the state-of-the-art, and future directions." *arXiv*, abs/2310.03195, 2023. <https://doi.org/10.1016/j.cie.2025.110856>.

57. O. Ogunfowora and H. Najjaran. "Reinforcement and Deep Reinforcement Learning-based Solutions for Machine Maintenance Planning, Scheduling Policies, and Optimization." *arXiv*, abs/2307.03860, 2023. <https://doi.org/10.48550/arxiv.2307.03860>.
58. Z. Abadi, N. Mansouri, and M. Javidi. "Deep reinforcement learning-based scheduling in distributed systems: a critical review." *Knowledge and Information Systems*, 66:5709-5782, 2024. <https://doi.org/10.1007/s10115-024-02167-7>.
59. J. Fernandes, S. Homayouni, and D. Fontes. "Energy-Efficient Scheduling in Job Shop Manufacturing Systems: A Literature Review." *Sustainability*, 14(10):6264, 2022. <https://doi.org/10.3390/su14106264>.
60. K. Chung, C. Lee, and Y. Tsang. "Neural combinatorial optimization with reinforcement learning in industrial engineering: a survey." *Artificial Intelligence Review*, 58(130), 2025. <https://doi.org/10.1007/s10462-024-11045-1>.
61. A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. "Stable-Baselines3: Reliable Reinforcement Learning Implementations." *Journal of Machine Learning Research*, 22(268):1-8, 2021. <http://jmlr.org/papers/v22/20-1364.html>.
62. Z. Chen, K. Zhang, P. Liu, G. Xin, Z. Sun, Z. Tao, Y. Zhang, W. Ji, Y. Lu, L. Jia, and H. Meng. "Worst-Case Soft Actor-Critic-Based Safe Reinforcement Learning Method for Nonlinear Constrained Waterflood Reservoir Production Optimization." *SPE Journal*, 2025. <https://doi.org/10.2118/230322-pa>.
63. Y. Liang, Y. Sun, R. Zheng, and F. Huang. "Efficient adversarial training without attacking: worst-case-aware robust reinforcement learning." arXiv preprint arXiv:2210.05927, 2022. <https://arxiv.org/abs/2210.05927>.
64. J. Wong and L. Liu. "Portfolio Optimization through a Multi-modal Deep Reinforcement Learning Framework." *Engineering: Open Access*, 3(4):1-8, 2025. <https://doi.org/10.33140/ea.03.04.a03>.
65. M. Su, H. Chai, C. Zhao, Y. Lyu, and J. Hu. "Lightweight Obstacle Avoidance for Fixed-Wing UAVs Using Entropy-Aware PPO." *Drones*, 9(9):598, 2025. <https://doi.org/10.3390/drones9090598>.
66. J. Park, J. Chun, S. Kim, Y. Kim, and J. Park. "Learning to schedule job-shop problems: representation and policy learning using graph neural network and reinforcement learning." *International Journal of Production Research*, 59:3360-3377, 2021. <https://doi.org/10.1080/00207543.2020.1870013>.
67. D. Hafner, J. Pašukonis, J. Ba, and T. Lillicrap. "Mastering diverse control tasks through world models." *Nature*, 640:647-653, 2025. <https://doi.org/10.1038/s41586-025-08744-2>.
68. J. Quan, W. Hu, X. X., and G. Chen. "Reinforcement Learning Stabilization for Quadrotor UAVs via Lipschitz-Constrained Policy Regularization." *Drones*, 9(10):675, 2025. <https://doi.org/10.3390/drones9100675>.
69. D. Behnke and M. J. Geiger. "Test instances for the flexible job shop scheduling problem with work centers." *Research Paper, Helmut-Schmidt-Universität, Lehrstuhl für Betriebswirtschaftslehre, insbes. Logistik-Management*, 2012.
70. Y. Lu, Q. Zhu, C. Tian, E. He, and T. Zhang. "Low-Carbon and Energy-Efficient Dynamic Flexible Job Shop Scheduling Method Towards Renewable Energy Driven Manufacturing." *Machines*, 14(1):88, 2026. <https://doi.org/10.3390/machines14010088>.
71. D. Cinar, Y. I. Topcu, and J. A. Oliveira. "A priority-based genetic algorithm for a flexible job shop scheduling problem." *Journal of Industrial & Management Optimization*, 12(4):1-18, 2016.
72. K. Deb and R. B. Agrawal. "Simulated binary crossover for continuous search space." *Complex Systems*, 9(2):115-148, 1995. http://www.complex-systems.com/abstracts/v09_i02_a02/.

Short Biography of Authors



Saurabh Sanjay Singh is a Ph.D. candidate in Industrial, Systems, and Manufacturing Engineering at Wichita State University. His research lies at the operations research–data science interface, focusing on production scheduling in job shops. At Wichita State University, he serves as an Instructor in the College of Engineering and previously supported graduate programs at the W. Frank Barton School of Business. He has taught programming and data science in India and brings industry experience in data science and machine learning from Tech Mahindra. Saurabh holds an M.S. in Data Science and a B.C.A. from CHRIST (Deemed to be University), India, and a Graduate Certificate in Business Analytics. Proficient in Python, R, SQL, and optimization and simulation toolchains, he builds deployable decision-support tools for operations management.



Deepak Gupta is Professor, Department Chair, and Graduate Program Coordinator in Industrial, Systems, and Manufacturing Engineering at Wichita State University. He earned a Ph.D. and M.S. in Industrial Engineering from West Virginia University and a bachelor's degree from IIT Roorkee. His expertise spans manufacturing system optimization, energy management, supply chain optimization, and data analytics. He has secured funding from USDA, U.S. DOE, U.S. DOL, NSF EPSCoR, regional utilities, and industry partners. Dr. Gupta has collaborated with more than 200 manufacturing companies across over ten U.S. states, integrating hands-on experience for students and bridging academic research with practical industrial applications.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.