

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; deepak.gupta@wichita.edu

* Correspondence: sxsingh28@shockers.wichita.edu

Abstract

Sustainable manufacturing requires schedules that balance environmental responsibility with delivery reliability. This paper studies the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T), where total carbon emissions and total tardiness penalty are the primary objectives. We propose a Policy-based Rough Optimization with a Large Neighborhood Search (Pro-LNS) framework integrating Proximal Policy Optimization (PPO) and adaptive Large Neighborhood Search (LNS). PPO constructs a feasible schedule by selecting operation-machine assignments from job-readiness, machine-availability, earliest-completion, and critical-path features. This policy-generated schedule provides a structurally informed incumbent, enabling LNS to avoid unguided search and focus destroy-and-repair refinement on high-impact operations. Both phases use the same normalized scalarized carbon-tardiness objective, which guides PPO rewards and LNS removal, reinsertion, and acceptance while preserving precedence, eligibility, and capacity constraints. Experiments on small, medium, and large workcenter benchmarks show strong due-date performance and controlled carbon emissions. Under equal objective weighting, Pro-LNS achieves a median optimality gap of 6.12% relative to the exact formulation, with all instances within 14%, while requiring 4.08 s on average and at most 10.51 s. Comparisons with PPO-only, Advantage Actor-Critic (A2C), Soft Actor-Critic (SAC), and Genetic Algorithm (GA) schedulers show that Pro-LNS attains the best weighted scalarized objective across representative instance-weight settings. Friedman and Holm-corrected Wilcoxon tests confirm significant improvements over all competitors, with average weighted-objective gains of 4.90%, 7.25%, 8.81%, and 9.51% over PPO-only, A2C, SAC, and GA, respectively. These results demonstrate that Pro-LNS is an effective and computationally practical hybrid approach for carbon-aware, tardiness-sensitive flexible job shop scheduling.



Academic Editors: Óscar Oliveira,
Eliana Costa e Silva and Dorabela
Gamboa

Received: 13 April 2026

Revised: 11 May 2026

Accepted: 12 May 2026

Published: 14 May 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: sustainable manufacturing; 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 both which machine should process each operation and in what order operations should be processed on each machine.

These two decisions are closely connected to both environmental and delivery outcomes. Machine assignment affects processing time, machine workload, energy use, and carbon emissions. Sequencing affects waiting times, machine congestion, completion times, and the risk that jobs are completed after their due dates. As a result, the same scheduling decision can improve one outcome while worsening another. A lower-carbon machine may reduce the environmental impact of an operation, but it may also be slower or more congested. A faster or more available machine may help reduce tardiness, but it may require more energy or have a higher carbon intensity. This creates a direct carbon versus due-date trade-off in real production environments [8,9].

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 [10,11]. However, sustainability-related concerns have historically received much less emphasis [12]. 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 [13]. 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 especially important in flexible job shop environments. The same machine flexibility that helps improve operations also creates differences in processing speed, machine availability, and emissions across alternative routes. Choosing one machine instead of another can, therefore, change not only the carbon footprint of the schedule but also the completion time of the affected job and the waiting times of other jobs. In the same way, a sequencing decision that protects one job from tardiness may delay another job or shift work toward machines with different emission characteristics. 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 [14].

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 [15,16]. 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 [17,18].

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 help explain schedule behavior, but 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 [19,20].

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 Flexible Job Shop Scheduling Problem (FJSP) has been studied through a wide range of exact, heuristic, metaheuristic, and hybrid approaches because it requires simultaneous decisions on operation sequencing and machine assignment [21,22]. Early and classical FJSP research established that routing flexibility makes the scheduling problem more realistic than the classical job shop problem but also increases the size and coupling of the search space [23,24]. Mixed-Integer Linear Programming (MILP) and Constraint Programming (CP) formulations have been used to express the feasibility structure of FJSP and related distributed variants [21,25]. Genetic Algorithm (GA), Ant Colony Optimization (ACO), tabu search, Non-dominated Sorting Genetic Algorithm II (NSGA-II), and hybrid heuristics have also been widely applied because they can search complex assignment-sequencing spaces where exact methods become computationally demanding [21,22].

As the FJSP literature has developed, sustainability-aware variants have introduced energy consumption, carbon emissions, machine-state behavior, and transportation-related effects into the scheduling model [12,26]. Green FJSP studies have considered makespan and carbon emissions together, particularly where alternative machines differ in processing time, power consumption, or carbon-related performance [13,27]. Other environmentally aware FJSP studies have included cost, carbon emissions, customer satisfaction, and time-of-use electricity pricing, showing that sustainability objectives can be integrated with broader production-planning criteria [28,29]. Integrated production and transportation models have further extended this direction by considering automated guided vehicles (AGVs), transport resources, and material-handling interactions [30,31]. These studies provide the environmental scheduling foundation on which the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T) builds.

A parallel stream of research has focused on due-date-oriented scheduling performance. Tardiness, delay time, due-date satisfaction, and customer-oriented service measures have been used when delivery reliability is a primary concern in shop scheduling [9,32]. Multi-objective FJSP studies have incorporated total tardiness, together with makespan and energy consumption, showing that due-date performance can be modeled directly within flexible shop scheduling [33,34]. Due-date optimization studies have also examined how schedule performance changes when delivery-related measures are treated as explicit objectives [35]. Dynamic FJSP studies using Deep Reinforcement Learning (DRL) have considered delay-related objectives in real-time scheduling environments, which further supports the relevance of learned dispatching when timing performance depends on evolving shop states [36,37]. These studies provide the due-date reliability foundation for CAFJSP-T.

Several recent studies have begun to connect environmental and due-date objectives within the same scheduling model. Low-carbon heterogeneous distributed FJSP research has considered energy consumption and weighted tardiness under dynamic job insertions and transfers [38]. Low-carbon many-objective FJSP research has included total carbon emissions and delay-related measures together with makespan and workload indicators [39]. Energy-saving FJSP research has combined total delay time and power consumption within enhanced NSGA-II frameworks [34]. Related flow shop research has paired total carbon emissions and total tardiness in a bi-objective formulation [40]. These studies show that the environmental and due-date dimensions have both been recognized in the scheduling literature while also motivating a focused flexible job shop formulation in which carbon emissions and tardiness penalty are treated as the two primary objectives.

Table 1 positions CAFJSP-T as an extension of several related scheduling streams rather than as a disconnected problem. Classical FJSP provides the assignment-sequencing foundation [21,23]. Tardiness-aware FJSP provides the due-date reliability foundation [32,33]. Green and low-carbon FJSP provide the environmental scheduling foundation [27,28]. Energy-aware and integrated green scheduling studies show how machine power, transport, and resource interactions can shape environmental performance [29,30]. CAFJSP-T builds on these streams by focusing on the direct pairing of carbon emissions and tardiness penalty within the flexible job shop structure.

Table 1. Positioning CAFJSP-T relative to existing scheduling problem classes.

Problem Class	Representative Studies	Main Objectives Commonly Considered	Contribution to the Literature	Scope Relative to CAFJSP-T	CAFJSP-T Extension
Classical FJSP	[21–23]	Makespan, workload, utilization, cost, and tardiness	Establishes the assignment-sequencing structure of flexible job shop scheduling	Environmental objectives are usually not central	Adds explicit carbon-emission evaluation to the flexible assignment-sequencing structure
Tardiness-aware FJSP	[9,32,33]	Tardiness, delay, due-date performance, and makespan	Emphasizes delivery reliability and due-date-oriented service performance	Carbon emissions or energy effects are usually not primary	Retains tardiness penalty as a primary operational objective while adding carbon awareness
Green or low-carbon FJSP	[13,27,28]	Carbon emissions, energy, cost, makespan, and customer satisfaction	Incorporates environmental performance into flexible shop scheduling	Due-date performance is often indirect or absent	Makes carbon emissions the primary environmental objective and pairs it directly with tardiness penalty
Energy-aware FJSP	[29,34,41]	Energy consumption, energy cost, power, makespan, and delay	Shows how machine states, pricing, and energy use affect scheduling	Energy is often used as the environmental measure rather than direct carbon emissions	Treats energy as a supporting indicator while optimizing carbon emissions directly
Integrated green FJSP with transport or AGVs	[30,31,42]	Energy, carbon, makespan, transportation, and AGV coordination	Extends scheduling to integrated production and material-handling decisions	Often focuses on energy, makespan, or transport rather than carbon-tardiness penalty	Provides a basis for future extensions, while CAFJSP-T focuses on the core carbon-tardiness trade-off
Carbon or energy plus tardiness scheduling	[38–40]	Energy, carbon, delay, tardiness, and makespan	Shows that environmental and due-date measures can be jointly optimized	Often uses energy rather than carbon, non-FJSP settings, or many-objective formulations	Defines a focused FJSP formulation with carbon emissions and tardiness penalty as the primary objectives

The solution literature for these problem classes includes several important methodological families. Exact optimization and MILP models provide rigorous mathematical formulations and are useful for feasibility modeling, benchmark comparison, and optimality-gap analysis [29,43]. In green FJSP, MILP has been combined with metaheuristics to study the trade-off between carbon emissions and production-time objectives [27]. CP and MILP formulations have also been used for distributed flexible job shop and multi-stage FJSP variants with assembly and AGV transportation [25,43]. These contributions are important because they formalize the problem structure and support solution validation. The literature also shows that heuristic, metaheuristic, and matheuristic methods become increasingly relevant as instance size, objective richness, and routing flexibility increase [44,45].

Evolutionary and population-based metaheuristics have been widely used because they provide flexible search mechanisms for multi-objective scheduling. Improved NSGA-II methods have been applied to low-carbon FJSP with objectives such as makespan, processing cost, and carbon emissions [46]. Enhanced NSGA-II has also been used for energy-saving FJSP with makespan, delay time, and power consumption objectives [34]. Improved sparrow search has been applied to green FJSP with learning effects and carbon emissions [27]. Enhanced GA has been used for green FJSP under time-of-use pricing with cost, carbon emissions, and customer satisfaction [28]. Multi-objective evolutionary and memetic algorithms have further been applied to FJSP variants with green considerations, fuzzy processing times, workload balance, and tardiness-related objectives [47,48]. These studies demonstrate the value of broad exploration in complex objective spaces.

Swarm and other population-based methods provide additional evidence of the field's movement toward adaptive and hybrid search. Shuffled frog-leaping algorithms have been used for FJSP with energy consumption [49]. Improved African buffalo optimization has been applied to green FJSP considering energy consumption [50]. Particle Swarm Optimization (PSO)-based and hybrid metaheuristic approaches have been used in green sustainable manufacturing and energy-aware FJSP with assembly operations [51,52]. Low-carbon joint scheduling in flexible open-shop environments has also been addressed using multi-objective PSO with problem-based neighborhood search [53]. These approaches show that population-based search remains important in sustainable scheduling. They also show that many successful implementations include local, variable-neighborhood, or problem-specific refinement components [54,55].

Reinforcement Learning (RL) has become increasingly important in scheduling because many scheduling problems can be represented as sequential decision processes. In RL-based scheduling, the state represents the current shop condition and the action represents a dispatching, sequencing, or assignment decision [56,57]. Deep Q-Network (DQN) methods have been used for dynamic flexible job shop scheduling and maintenance-constrained scheduling [58,59]. Graph-based DRL has been used to represent operation-machine relationships and variable scheduling topology [60,61]. DRL has also been applied to low-carbon FJSP, carbon-emission-aware FJSP, and dynamic low-carbon heterogeneous distributed FJSP [10,38,62]. These studies show that learning-based methods can capture state-dependent scheduling logic that may be difficult to encode manually.

Within the RL literature, Proximal Policy Optimization (PPO) is particularly relevant to policy-based schedule construction. PPO provides a policy-gradient framework with clipped policy updates, which supports stable policy improvement in sequential decision environments [63,64]. PPO has been applied to job shop scheduling with graph neural network representations, where the learned policy maps graph states to scheduling actions [65]. Multi-action PPO frameworks have been used for FJSP by decomposing scheduling decisions into operation selection and machine assignment [66]. Heterogeneous graph neural

network and PPO-based methods have also been used to solve FJSP by selecting operation-machine pairs from graph-based embeddings [60,67]. These studies support PPO as a suitable learning mechanism for policy-based schedule construction.

PPO-based scheduling research also supports the use of policy learning under delayed or multi-objective performance effects. Multi-policy PPO has been applied to multi-objective multiplicity FJSP with makespan and total tardiness objectives [68]. PPO-based reactive scheduling has been used for dynamic FJSP with random job arrivals and total tardiness minimization [37]. Hierarchical multi-agent PPO has been used for real-time dynamic multi-objective FJSP with partial no-wait constraints [69]. PPO and Mask-PPO methods with preference learning have also been explored for FJSP settings with large action spaces and invalid-action constraints [70]. These studies are useful for CAFJSP-T because the construction phase requires sequential operation-machine decisions with delayed effects on final schedule quality.

The literature also shows increasing interest in combining learning with search-based refinement. DRL-based flow shop scheduling has used learned policies together with improvement strategies to refine final solutions [71]. Knowledge-guided end-to-end RL for flow shop scheduling has used local search to improve policy-generated schedules [72]. Graph reinforcement learning has been integrated with Adaptive Large Neighborhood Search (ALNS) to improve operator selection within neighborhood-based search [73]. Search-online-learn-offline methods have combined learned heuristics with tree search for combinatorial optimization [74]. These studies support the broader principle that learning can guide the search toward promising regions, while local or neighborhood refinement can improve the final incumbent.

Hybrid learning-search methods have also appeared in green or energy-aware shop scheduling. Q-learning-guided Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D) has been used for energy-efficient FJSP to adapt neighborhoods and parameter control [75]. A DQN-based memetic algorithm has been developed for energy-efficient job shop scheduling with integrated limited AGVs [76]. DRL-based memetic algorithms have been used for energy-aware FJSP with multi-AGV coordination [42]. Reinforcement learning-based decomposition evolutionary algorithms have been used for fuzzy bi-objective FJSP [77]. These studies provide methodological support for combining learned guidance with search-based improvement in complex scheduling environments.

Large Neighborhood Search (LNS) provides the second methodological foundation for the proposed framework. Neighborhood search, tabu search, Variable Neighborhood Search (VNS), LNS, and ALNS have been widely used because they can improve incumbent schedules through structured modifications. Global-local neighborhood search and tabu search have been applied to FJSP to coordinate global operation scheduling and local machine assignment [78]. Hybrid genetic tabu search has been used for distributed FJSP, showing the value of combining exploration with local improvement [79]. Hybrid neighborhood structures have also been developed for FJSP benchmark problems to improve stability and efficiency [80]. Energy-aware FJSP with nonlinear routes and learning effects has been studied using heuristic and variable-neighborhood mechanisms [81]. These studies indicate that neighborhood design is a central issue in complex FJSP variants.

LNS is especially relevant when improvement requires coordinated changes rather than isolated swaps. Adaptive LNS was originally developed for difficult constrained routing problems, where destroy-and-repair mechanisms allow the search to preserve useful incumbent structure while reconstructing selected portions of the solution [82]. Multi-objective ALNS has been applied to distributed reentrant permutation flow shop scheduling [83]. ALNS-based decomposition has been used for green machine scheduling with makespan and energy objectives [84]. Multi-objective ALNS has also been applied

to dynamic FJSP with transportation resources [85]. These studies support the use of LNS as a refinement mechanism for complex scheduling problems with multiple interacting constraints and objectives.

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 LNS 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 methodological structure aligns closely with the requirements of CAFJSP-T. A policy-based method grounded in PPO offers a learning component suited to sequential operation-machine decisions and delayed objective effects [60,66]. Supporting structural cues such as critical-path information can further sharpen policy guidance in scheduling environments where downstream operations influence final schedule quality [42,60]. LNS then provides the refinement mechanism needed to improve beyond the initial policy output by revisiting larger groups of high-impact operations [82,84]. Taken together, this methodological combination aligns with the literature’s broader movement toward hybrid, adaptive, and structurally informed scheduling methods [71,73].

Table 2 summarizes the complementary roles of major solution families. Exact methods clarify the mathematical structure of the problem [25,43]. Evolutionary and swarm-based methods provide a broad exploration of complex objective spaces [46,47]. PPO and other RL methods provide adaptive policies for sequential scheduling decisions [60,66]. Hybrid learning–search methods show how learned guidance can be combined with refinement mechanisms [71,73]. LNS and related neighborhood-search methods provide intensification around feasible schedules and support larger incumbent-preserving modifications [78,84].

The choice of LNS in Pro-LNS follows from its role as a post-construction refinement mechanism. In CAFJSP-T, improving a completed schedule may require coordinated changes in both machine assignment and downstream sequencing. Neighborhood-search studies in FJSP show that coordinated treatment of global operation scheduling and local machine assignment can improve schedule quality [78]. Energy-aware and green scheduling studies show that neighborhood mechanisms can be designed around energy or environmental objectives [81,84]. Dynamic FJSP studies with transportation resources show that ALNS can be adapted to complex rescheduling environments [85]. These studies support the use of LNS as a targeted refinement phase after PPO has generated a feasible initial schedule.

Table 3 summarizes the rationale for selecting LNS as the refinement mechanism. Exact reoptimization remains valuable for formulation and benchmarking, but targeted partial reconstruction is more consistent with the role of a fast post-construction improvement phase [43,45]. Population-based methods provide useful exploration, while LNS intensifies around a promising incumbent generated by the policy stage [47,84]. Standalone PPO and DRL provide adaptive construction, while LNS supplies the instance-specific repair needed after the full schedule has been realized [60,72]. This combination allows Pro-LNS to preserve the useful structure of the PPO-generated schedule while refining high-impact operations under the same carbon-tardiness objective.

Table 2. Comparative assessment of existing solution approaches.

Approach	Representative Studies	Objectives Addressed	Advantages	Scope and Limitations for CAFJSP-T	Relevance to Pro-LNS
Exact optimization, MILP, and CP	[27,29,43]	Makespan, energy, carbon, feasibility, and transportation	Provides rigorous formulations and useful bounds	Scalability becomes challenging as flexible routing, sequencing, carbon, and tardiness interact	Supports the CAFJSP-T model and warm-start evaluation logic
GA, NSGA-II, and evolutionary algorithms	[28,46,47]	Makespan, energy, carbon, cost, customer satisfaction, and tardiness	Provides broad exploration of multi-objective trade-offs	Often benefits from adaptive operators or local search in complex FJSP variants	Motivates hybrid search and post-construction refinement
Swarm and population-based heuristics	[49,50,53]	Energy, carbon, makespan, workload, and transport	Flexible and adaptable across green scheduling variants	Performance can depend on parameter settings and problem-specific operators	Supports the value of adaptive search in green scheduling
DQN and value-based DRL	[38,58,59]	Dynamic scheduling, dispatching, energy, carbon, and delay	Learns state-dependent scheduling behavior	Requires careful state and reward design, and may focus on specific objective sets	Supports learning-based scheduling but motivates PPO-based construction plus refinement
PPO and policy-gradient RL	[60,65,66]	Makespan, tardiness, dynamic scheduling, and multi-objective scheduling	Supports stable policy learning for sequential decisions and large action spaces	Policy output can still benefit from instance-specific post-processing	Justifies the rough optimization phase of Pro-LNS
Hybrid RL plus search	[71–73]	Scheduling, routing, energy-aware job shop, and combinatorial optimization	Combines learned guidance with local or neighborhood refinement	Often developed for makespan, routing, energy, or operator control rather than CAFJSP-T	Supports the Pro-LNS design logic
Tabu search, local search, and VNS	[78,79,81]	Sequencing, assignment, makespan, energy, and transportation	Strong local improvement around incumbent schedules	Small neighborhoods may not capture larger assignment-sequencing interactions	Motivates larger LNS-style destroy-and-repair refinement
LNS and ALNS	[82,84,85]	Energy, makespan, workload, dynamic scheduling, and transport	Preserves useful incumbent structure while reconstructing high-impact parts	Requires problem-specific destroy, repair, and acceptance rules	Directly supports the LNS phase of Pro-LNS

Overall, the literature supports the methodological logic of Pro-LNS. The FJSP literature provides the assignment-sequencing foundation [21,24]. Green and low-carbon FJSP studies provide the carbon and energy-aware scheduling foundation [27,28]. Tardiness-aware scheduling studies provide the due-date reliability foundation [9,33]. PPO-based scheduling studies support policy-based construction in sequential decision environments [65,66]. LNS and ALNS studies support large-neighborhood refinement around promising incumbent schedules [82,84]. Building on these foundations, Pro-LNS integrates PPO-based schedule construction with LNS-based refinement for CAFJSP-T under a unified carbon-tardiness objective.

Table 3. Rationale for selecting LNS as the refinement mechanism.

Candidate Refinement Method	Strengths	Scope and Limitations for CAFJSP-T Refinement	Why LNS Is Selected or Complementary
Exact reoptimization	Provides rigorous optimization and bounds [25,43]	Full reoptimization may become computationally expensive as routing, sequencing, carbon, and tardiness interact [44,45]	LNS performs targeted partial reconstruction while preserving the incumbent
GA or NSGA-II	Provides population-level exploration [46,47]	Detailed refinement may require many evaluations or embedded local search [48,86]	LNS intensifies around one promising PPO-generated schedule
PSO, ACO, and swarm methods	Offers flexible search over complex discrete spaces [49,50]	Search quality can depend on parameters, representation, and operators [51,52]	LNS provides a direct destroy-and-repair mechanism tied to objective impact
Standalone PPO or DRL	Produces fast adaptive schedules after training [37,60]	A policy-generated schedule may still contain improvable structures [71,72]	LNS refines high-impact operations without discarding the PPO-guided incumbent
Tabu search and small-neighborhood local search	Provides strong local improvement [78,79]	Small neighborhoods may not capture larger coordinated changes [80,81]	LNS expands the move scale by reconstructing groups of operations
LNS or ALNS	Supports incumbent-preserving large destroy-and-repair moves [82,84]	Requires tailored removal, repair, and acceptance logic [85,87]	Best matches Pro-LNS because the same carbon-tardiness objective can guide removal, reinsertion, and acceptance

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: the total carbon emissions and the 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.

3.1.1. 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.

3.1.2. Notations

Table 4 summarizes the notation used in the mathematical formulation of the CAFJSP-T.

Table 4. Notation used in the CAFJSP-T formulation.

Symbol	Description
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 $j \in J$
$\mathcal{O}$	Set of all operations, $\mathcal{O} = \{(j, o) \mid j \in J, o \in \mathcal{O}_j\}$
$M_{j,o} \subseteq M$	Eligible machines for operation (j, o)
Parameters	
$p_{j,o,m}$	Processing time of operation (j, o) on machine $m \in M_{j,o}$, in minutes
P_m^{proc}	Power consumption of machine m during processing, in kW
CI_m	Carbon intensity of machine m , in kg CO ₂ /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	Sufficiently large positive constant
Derived Quantities	
$e_{j,o,m}^{\text{carbon}}$	Carbon emissions, $e_{j,o,m}^{\text{carbon}} = \frac{p_{j,o,m} P_m^{\text{proc}} CI_m}{60}$, in kg CO ₂
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

3.1.3. 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 Section 3.1.4. 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.

3.1.4. 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, Equation (10) gives

$$f_1(x) = \sum_{(j,o) \in \mathcal{O}} \sum_{m \in M_{j,o}} x_{j,o,m} e_{j,o,m}^{\text{carbon}}, \quad f_2(x) = \pi \sum_{j \in J} D_j, \quad (10)$$

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 shown in Equation (11):

$$\tilde{f}_1(x) = \frac{f_1(x)}{B_1}, \quad \tilde{f}_2(x) = \frac{f_2(x)}{B_2}. \quad (11)$$

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 shown in Equations (12a) and (12b):

$$Z = w_1 \tilde{f}_1(x) + w_2 \tilde{f}_2(x) \quad (12a)$$

or, equivalently,

$$Z = w_1 \frac{f_1(x)}{B_1} + w_2 \frac{f_2(x)}{B_2}, \quad (12b)$$

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 the comparability of scale, while the scalarization weights w_1 and w_2 encode the 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), as shown in Equation (13):

$$(\mathcal{S}, \mathcal{A}, T, r, H), \quad H = |\mathcal{O}|. \quad (13)$$

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 is defined in Equation (14)

$$s_t = (f_t, m_t, e_t, c_t) \quad (14)$$

is formed by concatenating Equation (15)

$$f_t \in \{0, 1\}^{|\mathcal{O}|}, \quad m_t \in \mathbb{R}^{|\mathcal{M}|}, \quad e_t \in [0, 1]^{|\mathcal{O}| \times M_{\max}}, \quad c_t \in [0, 1]^{|\mathcal{O}| \times 2}, \quad (15)$$

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

- Ready-flag vector: Indicates which operations are currently eligible for dispatch, as shown in Equation (16):

$$f_{t,(j,o)} = \begin{cases} 1, & \text{if all predecessors of } (j, o) \text{ have been completed by step } t, \\ 0, & \text{otherwise.} \end{cases} \quad (16)$$

- Machine ready-time vector: Records the earliest time at which each machine becomes available, as shown in Equation (17):

$$m_{t,m} = \begin{cases} \max\{C_{j',o'} \mid (j', o') \text{ assigned to machine } m\}, & \text{if machine } m \text{ has processed} \\ & \text{at least one operation,} \\ 0, & \text{otherwise.} \end{cases} \quad (17)$$

- Normalized earliest-completion-time matrix: Estimates the completion time of each eligible operation on each eligible machine, as shown in Equation (18):

$$e_{t,(j,o),k} = \frac{1}{H} \left(\max\{m_{t,m_k}, C_{j,o-1}\} + p_{j,o,m_k} \right), \quad (18)$$

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 as shown in Equation (19). 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.

$$\tau_t = \min_{m \in M} m_{t,m} \quad (19)$$

denote the earliest machine-available time at decision step t . Then Equation (20)

$$c_{t,(j,o),1} = \frac{\text{succ_time}_{j,o}}{H}, \quad c_{t,(j,o),2} = \frac{\max\{0, d_j - \tau_t - \text{succ_time}_{j,o}\}}{H}. \quad (20)$$

Action Space

At state s_t , the agent selects an action in Equation (21)

$$a_t = (j, o, m), \quad (21)$$

where operation (j, o) (as shown in Equation (22)) must be ready for processing, and machine m must belong to the eligible machine set $M_{j,o}$.

$$(j, o) \in \{(j', o') \mid f_{t,(j', o')} = 1\}, \quad m \in M_{j,o}. \quad (22)$$

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, as shown in Equation (23):

$$S_{j,o} = \max\{m_{t,m}, C_{j,o-1}\}, \quad C_{j,o} = S_{j,o} + p_{j,o,m}. \quad (23)$$

The system state is then updated according to the new completion time as in Equation (24):

$$m_{t+1,m} = C_{j,o}, \quad (24)$$

and Equation (25):

$$f_{t+1,(j,o)} = 0, \quad f_{t+1,(j,o+1)} = 1 \quad \text{if operation } (j, o+1) \text{ exists.} \quad (25)$$

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 shown in Equation (26):

$$r(s_t, a_t) = - \left[w_1 \left(\frac{e_{j,o,m}^{\text{carbon}}}{B_1} \right) + w_2 \left(\frac{\pi \Delta D_t}{B_2} \right) \right], \quad (26)$$

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, as shown in Equation (27):

$$J(\theta) = \mathbb{E}_{\tau \sim \pi_\theta} \left[\sum_{t=0}^{H-1} r(s_t, a_t) \right]. \quad (27)$$

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, as shown in Equation (28):

$$J(\sigma) = w_1 \left(\frac{\text{TotalCarbonEmission}(\sigma)}{B_1} \right) + w_2 \left(\frac{\text{TotalTardinessPenalty}(\sigma)}{B_2} \right). \quad (28)$$

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 shown in Equation (29):

$$\text{score}_{j,o} = w_1 \left(\frac{\Delta \text{Carbon}_{j,o}}{B_1} \right) + w_2 \left(\frac{\Delta \text{TardinessPenalty}_{j,o}}{B_2} \right). \quad (29)$$

2. Removal: Remove the k_t operations with the highest $\text{score}_{j,o}$, producing a partial schedule in which the most disruptive operations are unscheduled.
3. Adaptive tuning: Define the destroy-size bounds dynamically as shown in Equation (30):

$$k_{\min} = \max\{2, \lceil \rho_{\min} |\mathcal{O}| \rceil\}, \quad k_{\max} = \min\{|\mathcal{O}| - 1, \lceil \rho_{\max} |\mathcal{O}| \rceil\}, \quad (30)$$

where $0 < \rho_{\min} < \rho_{\max} < 1$ are predefined ratios. If reinserting the removed operations yields an improved schedule, set k_{t+1} according to Equation (31):

$$k_{t+1} = \max(k_{\min}, k_t - 1); \quad (31)$$

otherwise, set k_{t+1} according to Equation (32):

$$k_{t+1} = \min(k_{\max}, k_t + 1). \quad (32)$$

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 the start and completion times, as shown in Equation (33):

$$s_{j,o}(m) = \max\{\text{ready_time}(m), C_{j,o-1}\}, \quad c_{j,o}(m) = s_{j,o}(m) + p_{j,o,m}. \quad (33)$$

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 shown in Equation (34):

$$m^* = \arg \min_{m \in M_{j,o}} \Delta J(m). \quad (34)$$

Acceptance and Adaptation

Let σ^{t+1} denote the schedule obtained after reinsertion. If Equation (35) holds,

$$J(\sigma^{t+1}) < J(\sigma^t), \quad (35)$$

then σ^{t+1} is accepted as the new incumbent schedule, and the removal size is reduced according to Equation (36):

$$k_{t+1} = \max(k_{\min}, k_t - 1). \quad (36)$$

Otherwise, the candidate schedule is rejected, and the removal size is increased according to Equation (37):

$$k_{t+1} = \min(k_{\max}, k_t + 1). \quad (37)$$

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.

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 [88].

The policy is implemented in Stable-Baselines3 [89] 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.

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 $J(\sigma^*)$ **Training Protocol**

To improve robustness and generalization, PPO is trained on a pool of 20 synthetically generated large-scale CAFJSP-T instances. These instances are deliberately more demand-

ing than the benchmark cases, with 90–110 jobs, 54–66 machines, five operations per job, 3–62 eligible machines per operation, and processing times of 5–35 min. This design exposes the policy to structurally difficult scheduling conditions and supports transfer to smaller or less dense benchmark instances [90,91]. Training is conducted for 500,000 timesteps using eight parallel vectorized environments, and the trained policy is saved once and reused for all benchmark evaluations without further fine-tuning. Training required 5 h and 34 min as a one-time offline cost. This cost is not incurred during benchmark evaluation, where the saved policy is used only to construct a schedule before LNS refinement. Thus, the reported evaluation times reflect online deployment cost rather than the PPO training time. Convergence is assessed using the episodic reward trajectory in Figure 1, where the returns stabilize in the later stages of training, supporting the fixed training budget as a practical stopping criterion.

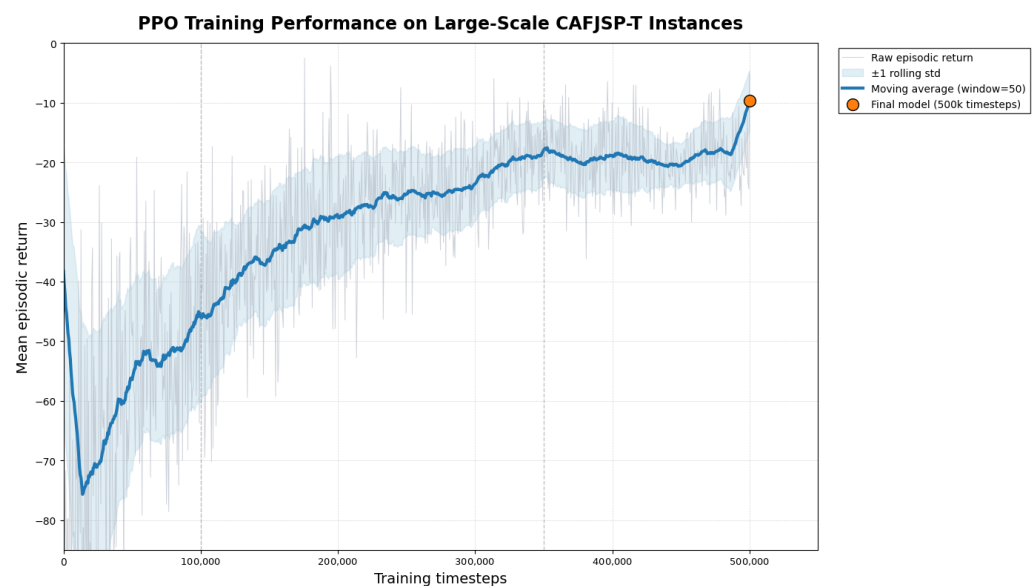


Figure 1. Training performance - mean episodic return over 500 k timesteps.

PPO Technical Details and Hyperparameter Selection

The PPO configuration used in this study is organized below by the role each hyperparameter group plays in policy learning and training stability.

1. Policy representation: MLP, [256, 256], ReLU. An MLP is used because the CAFJSP-T state is represented as structured numerical inputs, including the ready-flag vector, machine ready-time vector, normalized earliest-completion-time matrix, and critical-path metrics. The two-layer 256-neuron architecture with ReLU provides sufficient nonlinear capacity to capture interactions among job readiness, machine availability, completion-time estimates, and path criticality while keeping training and inference efficient, consistent with deep actor-critic scheduling architectures that improve makespan, tardiness, utilization, and computational performance [92–95].
2. Optimization stability: learning rate 5×10^{-5} , mini-batch size 64, 10 epochs, max gradient norm 0.5. These values are selected to make PPO updates conservative and stable, since small policy changes can strongly affect downstream sequencing and machine availability decisions. Minibatch, multi-epoch PPO training with constrained updates is commonly used in scheduling frameworks to improve convergence, robustness, makespan, and tardiness performance [64,69,93,96,97].
3. Exploration and update control: entropy coefficient 0.001, clipping range 0.2, advantage normalization. The entropy coefficient encourages limited exploration, while the 0.2 clipping range and advantage normalization prevent unstable policy updates.

These mechanisms are important in scheduling because overly random or aggressive updates can degrade dispatching quality; prior PPO scheduling frameworks link clipped updates with stable learning and improved robustness [64,65,68,93].

4. Long-horizon credit assignment: $\gamma = 0.99$, $\lambda = 0.95$. These values are used because CAFJSP-T decisions have delayed effects on machine idle times, availability, completion times, and final objective values. High discounting and GAE-style advantage estimation are suitable for long-horizon scheduling and delayed-reward environments [37,56,69,98,99].
5. Actor-critic loss: clipped surrogate objective, MSE value loss, entropy regularization, value coefficient 0.5. This standard PPO loss is used to balance policy improvement, value-function learning, and exploration. Actor-critic PPO formulations are widely used in scheduling literature because they stabilize learning and improve solution quality under complex shop-floor dynamics [64,65,93,96,100].
6. Training scale: 500,000 timesteps and eight parallel environments. This training budget is used to expose the agent to diverse production states, while parallel environments improve rollout collection efficiency. Simulation-based PPO scheduling studies similarly rely on extended training experience to improve robustness, generalization, and computational efficiency [64,94,95,101–103].

3.3. Benchmark Instances and Experimental Setup

The computational study is based on 15 benchmark instances from Behnke and Geiger [104], covering small, medium, and large workcenter (WC) configurations. Because these instances do not include sustainability-related parameters, they are extended using the environmental and due-date settings reported by Lu et al. [105]. Specifically, machine processing power values are sampled from [10, 20] kW, carbon intensity is fixed at 0.998 kgCO₂/kWh, the due-date tightness parameter is taken from $\theta \in [0.5, 1.5]$, and the tardiness penalty is set to 0.1 per minute of tardiness.

Using these settings, due dates are generated as shown in Equation (38):

$$d_j = \theta \cdot \sum_{o \in O_j} \max_{m \in M_{j,o}} p_{j,o,m}, \quad \forall j \in J. \quad (38)$$

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 are computed for objective normalization using single-objective Genetic Algorithm runs [106,107]. One representative instance is selected for each benchmark category, and the resulting baseline values are reported in Table 5. For each experiment, the normalization constants are 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 5. 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 consists of three components:

- Benchmark-based warm-start evaluation: The proposed Pro-LNS framework is applied to the full set of benchmark instances. For each instance, the final Pro-LNS solution is 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 is then run on the same instance to obtain a best bound and the corresponding optimality gap. This procedure is 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 is to proven optimality within the allotted MILP solve time.

- **Weight-sensitivity analysis:** A weight-sensitivity analysis is 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 is 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.
- **Representative-instance algorithm comparison:** Pro-LNS is compared with a Proximal Policy Optimization (PPO)-only ablation, an Advantage Actor-Critic (A2C) Scheduler [108], a Soft Actor-Critic (SAC) Scheduler [109], and a Genetic Algorithm (GA) [106,107] using one representative instance from each small, medium, and large workcenter (WC) configuration. All methods use the same benchmark data, feasibility rules, category-specific normalization constants, and carbon-tardiness scalarized objective. PPO-only corresponds to Phase 1 of Pro-LNS without LNS refinement. A2C and SAC retain their original Markov decision process (MDP) and policy structures, with their objective evaluation adapted to CAFJSP-T. The GA baseline uses the same schedule encoding and objective calculation, with a population size of 150, 200 generations, adaptive crossover probability of 0.85, problem-specific mutation probability of 0.05, including critical-path mutation, and termination after the maximum generation limit or 50 generations without improvement. The comparison is conducted under multiple scalarization weights, and statistical tests are used across WC configurations and weight settings to assess whether the observed performance differences are significant.

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 the 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 6 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 6, 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 s, and the maximum reported CPU time is 10.51 s. 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 a low runtime, controlled emissions, and relatively tight optimality gaps means that decision-makers can obtain strong schedules quickly while still retaining confidence that the solutions are relatively close to the benchmarks provided by the exact optimization frameworks. This makes the method especially applicable to low-carbon production planning, due-date-driven job shops, make-to-order manufacturing environments, and machine-flexible facilities where alternative machines differ in processing time, energy use, or carbon intensity. It is also useful in shops that require repeated rescheduling under limited planning time, such as when customer priorities change, bottlenecks emerge, or updated production plans must be generated during daily operations.

Table 6. Benchmark-based warm-start evaluation results.

Instance	Carbon (kg CO ₂)	Tardiness Penalty	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 7.

Table 7. Weight-sensitivity analysis results, for instance, sm02_3 under varying scalarization weights.

Weight	Carbon Emissions (kg CO ₂)	Tardiness Penalty	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 7 shows that the main effect of changing the scalarization weights is expressed through the 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 the 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 min) and the 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 the tardiness penalty only moderately (13.04 vs. 10.24), with makespan remaining nearly unchanged (206 vs. 208 min). 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 still retaining strong performance on the other, with $w_1 = 0.8, w_2 = 0.2$ emerging as an effective carbon-leaning policy and $w_1 = 0.2, w_2 = 0.8$ as an effective tardiness-leaning compromise.

The benchmark-based warm-start evaluation shows that Pro-LNS produces feasible, high-quality solutions under the equal-weight objective setting. The weight-sensitivity analysis then studies how the method responds when the relative priority between carbon emissions and tardiness penalty is varied. Together, these two analyses evaluate Pro-LNS in terms of exact-model proximity and preference sensitivity. However, they do not directly show whether the proposed framework outperforms alternative scheduling approaches under comparable representative conditions. To address this, a representative-instance algorithm comparison is conducted using one representative instance from each workcenter (WC) configuration: sm04_5, med04_5, and lar04_5. Pro-LNS is compared with a Proximal Policy Optimization (PPO)-only ablation of the proposed framework, an Advantage Actor-Critic (A2C) Scheduler [108], a Soft Actor-Critic (SAC) Scheduler [109], and a Genetic Algorithm (GA) [106,107]. For each representative instance, three scalarization settings are tested: $(w_1, w_2) = (0.8, 0.2)$, $(0.5, 0.5)$, and $(0.2, 0.8)$, representing carbon-leaning, equal-weight, and tardiness-leaning preferences. The component-level comparison results are reported in Table 8, and the corresponding weighted scalarized objective values are shown in Figure 5.

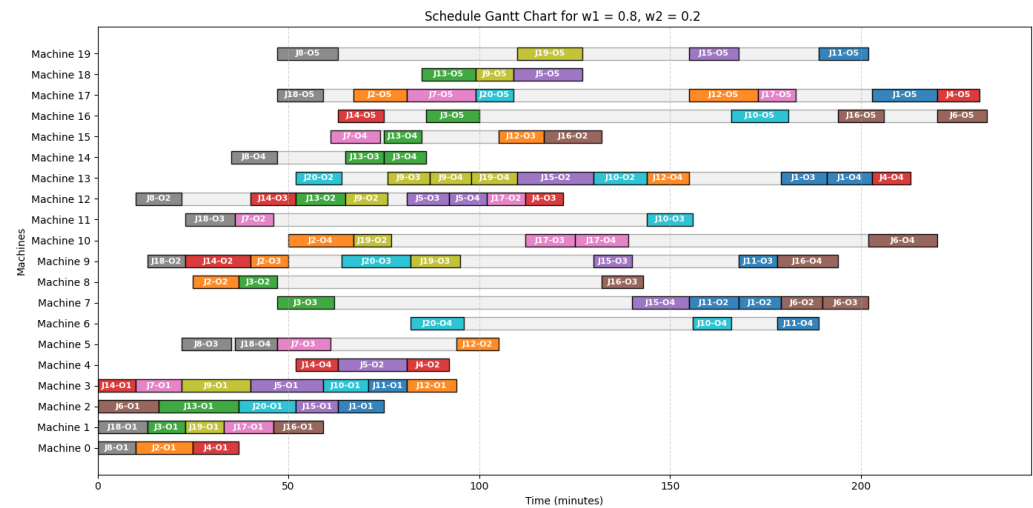


Figure 2. Schedule Gantt chart, for instance, sm02_3 under carbon-priority weighting ($w_1 = 0.8$, $w_2 = 0.2$), showing the machine assignments and operation sequencing obtained when carbon emissions are emphasized over tardiness penalty. *Note.* Each horizontal row represents one machine, and each colored block represents a scheduled operation assigned to that machine. The block labels follow the format $Ji-k$, where Ji denotes the job number, and k denotes the operation number of that job. For example, J10-05 represents the fifth operation of Job 10. The horizontal position of each block indicates the operation start time, while the block length represents its processing duration. The colors are used to visually distinguish different jobs and improve readability. Reading each machine row from left to right shows the processing sequence on that machine.

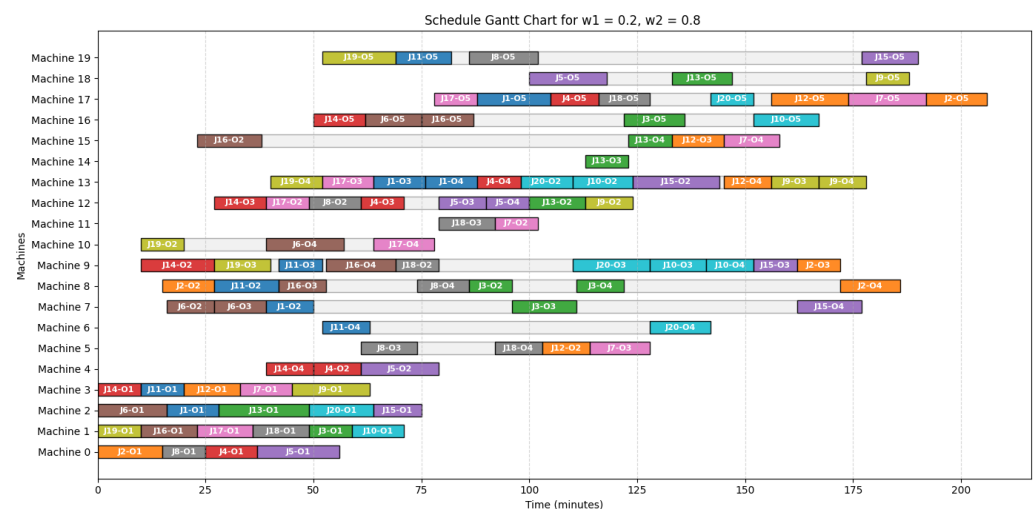


Figure 3. Schedule Gantt chart, for instance, sm02_3 under tardiness-priority weighting ($w_1 = 0.2$, $w_2 = 0.8$), illustrating the machine assignments and operation sequencing obtained when tardiness penalty is emphasized over carbon emissions. *Note.* Each horizontal row represents one machine, and each colored block represents a scheduled operation assigned to that machine. The block labels follow the format $Ji-k$, where Ji denotes the job number, and k denotes the operation number of that job. For example, J10-05 represents the fifth operation of Job 10. The horizontal position of each block indicates the operation start time, while the block length represents its processing duration. The colors are used to visually distinguish different jobs and improve readability. Reading each machine row from left to right shows the processing sequence on that machine.

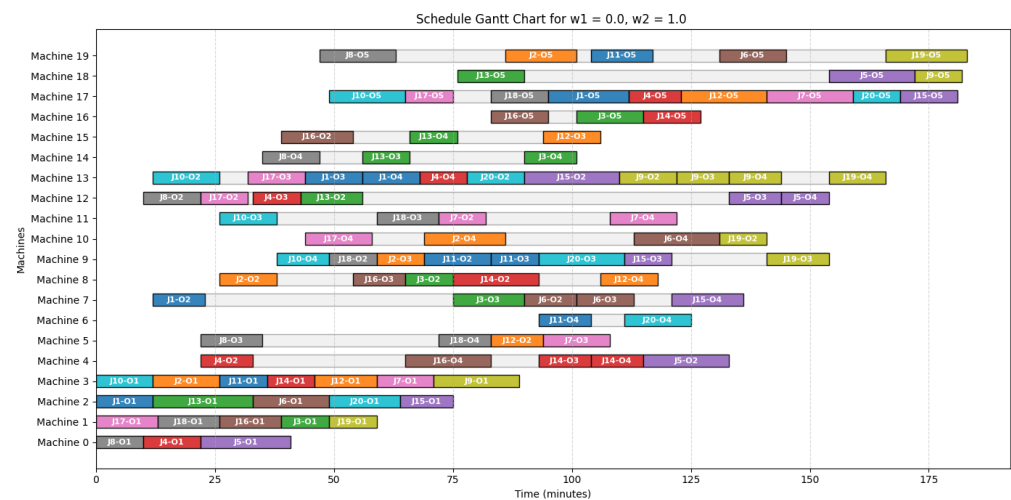


Figure 4. Detailed schedule visualization for instance sm02_3 under tardiness only priority weighting ($w_1 = 0.0, w_2 = 1.0$), highlighting how the production sequence shifts when delivery performance is the only priority. *Note.* Each horizontal row represents one machine, and each colored block represents a scheduled operation assigned to that machine. The block labels follow the format J_i-k , where J_i denotes the job number and k denotes the operation number of that job. For example, $J_{10}-05$ represents the fifth operation of Job 10. The horizontal position of each block indicates the operation start time, while the block length represents its processing duration. The colors are used to visually distinguish different jobs and improve readability. Reading each machine row from left to right shows the processing sequence on that machine.

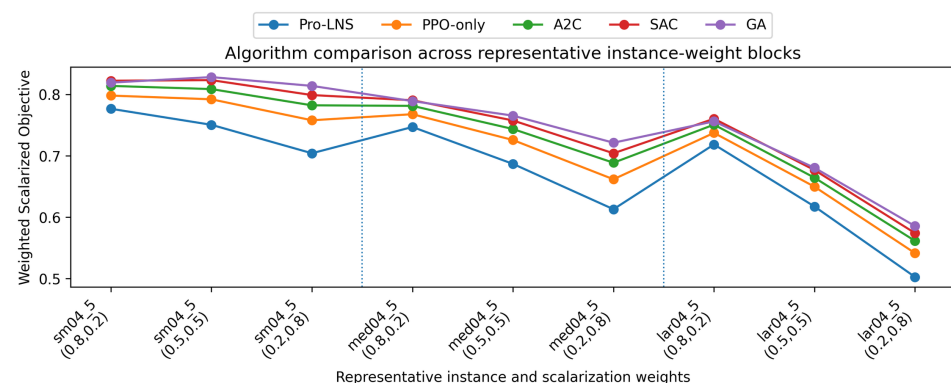


Figure 5. Weighted & Scaled objective values across representative instance-weight blocks. Lower values indicate better performance.

Table 8 and Figure 5 summarize the representative-instance comparison from two complementary views: the table reports the carbon emissions, tardiness, and runtime values, while the figure condenses carbon emissions and tardiness penalty into the weighted objective used for evaluation. Across both views, Pro-LNS remains the strongest method, producing the lowest carbon emissions and tardiness penalty in all nine instance-weight combinations and maintaining the lowest weighted objective curve across the representative blocks. The Proximal Policy Optimization (PPO)-only ablation is generally the closest competing method, which suggests that the enhanced policy state representation, including the normalized earliest-completion-time matrix and critical-path metrics, provides useful scheduling information for constructing strong initial solutions. However, the consistent gap between PPO-only and Pro-LNS shows that the LNS refinement stage contributes additional value beyond the learned policy alone by refining the incumbent schedule through targeted local improvements. This benefit is closely tied to the quality and structure of the initial schedule produced by the policy; therefore, the same refinement stage may not be expected to provide the same level of improvement if paired with less

structurally informative solutions, such as those produced by the Advantage Actor-Critic (A2C) and Soft Actor-Critic (SAC) schedulers in this comparison. In contrast, the Genetic Algorithm (GA) does not benefit from a learned policy or state representation and instead relies on population-based search, which leads to higher computational times and weaker solution quality in these experiments. Overall, Pro-LNS outperforms the evaluated alternatives by combining a structurally informative PPO-based initial schedule with targeted LNS refinement.

Table 8. Representative-instance comparison results.

Instance	Weights	Metric	Pro-LNS	PPO-Only	A2C	SAC	GA
sm04_5	(0.8, 0.2)	Carbon (kg CO ₂)	1475.28	1494.61	1510.82	1521.37	1507.19
		Tardiness Penalty	3214.78	3507.33	3702.64	3789.41	3855.72
		CPU (s)	8.59	4.81	0.61	0.15	21.07
	(0.5, 0.5)	Carbon (kg CO ₂)	1499.46	1519.71	1533.18	1545.62	1531.44
		Tardiness Penalty	3093.06	3412.89	3528.15	3627.83	3705.19
		CPU (s)	8.55	4.83	0.78	0.17	26.07
	(0.2, 0.8)	Carbon (kg CO ₂)	1508.73	1528.94	1541.57	1554.91	1539.82
		Tardiness Penalty	3002.51	3287.74	3414.88	3499.43	3591.02
		CPU (s)	8.54	4.87	0.91	0.20	27.87
med04_5	(0.8, 0.2)	Carbon (kg CO ₂)	1556.47	1577.16	1595.43	1608.72	1591.86
		Tardiness Penalty	1812.45	1993.69	2084.32	2139.57	2216.38
		CPU (s)	10.56	5.84	0.62	0.14	25.19
	(0.5, 0.5)	Carbon (kg CO ₂)	1580.71	1605.38	1620.65	1636.11	1617.93
		Tardiness Penalty	1685.21	1873.58	1954.84	2012.66	2084.19
		CPU (s)	10.54	5.88	0.89	0.17	28.14
	(0.2, 0.8)	Carbon (kg CO ₂)	1591.55	1613.17	1630.82	1647.30	1626.74
		Tardiness Penalty	1634.77	1802.46	1893.12	1942.58	2011.83
		CPU (s)	10.55	5.93	0.93	0.18	29.59
lar04_5	(0.8, 0.2)	Carbon (kg CO ₂)	1412.54	1432.46	1451.93	1466.38	1447.85
		Tardiness Penalty	1298.63	1432.58	1496.82	1540.29	1591.46
		CPU (s)	9.95	5.30	0.64	0.19	23.03
	(0.5, 0.5)	Carbon (kg CO ₂)	1438.07	1461.41	1477.63	1494.50	1473.95
		Tardiness Penalty	1180.54	1317.46	1371.63	1413.08	1462.71
		CPU (s)	9.98	5.41	0.66	0.16	29.93
	(0.2, 0.8)	Carbon (kg CO ₂)	1449.85	1469.80	1488.15	1505.74	1482.60
		Tardiness Penalty	1138.46	1259.81	1319.35	1354.92	1401.28
		CPU (s)	9.91	5.62	0.88	0.16	29.03

The consistency of the algorithmic performance differences is further examined through a statistical analysis. Each representative instance-(scalar) weight setting is treated as a paired experimental block, yielding nine paired observations in total. Since all five algorithms are evaluated on the same blocks, a Friedman rank test is first used to determine whether significant differences exist among the algorithms. Post hoc Wilcoxon signed-rank tests with Holm correction are then conducted to compare Pro-LNS directly against each competing method while accounting for multiple comparisons.

Table 9 shows statistically significant algorithm effects for all three solution-quality measures. For the weighted scalarized objective, the Friedman test gives $\chi^2(4) = 34.40$, $p = 6.17 \times 10^{-7}$, and Kendall's $W = 0.956$, indicating a strong and consistent difference among the algorithms. The average ranks further support the observed performance pattern, with Pro-LNS achieving the best rank for the weighted scalarized objective, carbon emissions, and tardiness penalty. PPO-only ranks second for all three measures, while A2C, SAC, and GA generally occupy weaker positions depending on the metric. Since the Friedman test indicates significant overall differences, pairwise post hoc comparisons are then conducted between Pro-LNS and each competing method.

Table 10 confirms that Pro-LNS significantly outperforms each competing method for the weighted scalarized objective after Holm correction. The adjusted p -value is 0.0156 in all four comparisons, which is below the 0.05 significance level. The improvement percentages are computed from the mean values across the nine paired settings and show that Pro-LNS improves the weighted scalarized objective by 4.90% over PPO-only, 7.25%

over A2C, 8.81% over SAC, and 9.51% over GA. The largest gains occur in tardiness reduction, while carbon reductions are smaller but consistent. These results provide statistical support for the conclusion that Pro-LNS delivers the strongest solution quality among the evaluated methods.

Table 9. Friedman test results and average ranks across representative instance-weight settings.

Metric	χ^2	df	p -Value	Kendall's W	Pro-LNS	PPO-Only	A2C	SAC	GA
Weighted scalarized objective	34.40	4	6.17×10^{-7}	0.956	1.00	2.00	3.00	4.33	4.67
Carbon emissions	36.00	4	2.89×10^{-7}	1.000	1.00	2.00	3.67	5.00	3.33
Tardiness penalty	36.00	4	2.89×10^{-7}	1.000	1.00	2.00	3.00	4.00	5.00

Table 10. Holm-corrected Wilcoxon post hoc comparisons against Pro-LNS.

Comparison	Wilcoxon Statistic	Raw p	Holm-Adjusted p	Objective Improvement (%)	Carbon Reduction (%)	Tardiness Reduction (%)
Pro-LNS vs. PPO-only	0.00	0.0039	0.0156	4.90	1.39	9.19
Pro-LNS vs. A2C	0.00	0.0039	0.0156	7.25	2.44	13.03
Pro-LNS vs. SAC	0.00	0.0039	0.0156	8.81	3.35	15.29
Pro-LNS vs. GA	0.00	0.0039	0.0156	9.51	2.22	17.61

Computational Environment

All experiments are implemented in Python 3.14.5 and executed on a high-performance computing (HPC) platform to support the training, evaluation, and testing of the scheduling models in high-dimensional state and action spaces. The platform is based on the x86_64 architecture and uses Intel(R) Xeon(R) Gold 6240 processors with a base frequency of 2.60 GHz and a maximum frequency of 3.90 GHz. The system provides 72 logical CPU cores across two sockets, with 18 physical cores per socket and two threads per core. The cache hierarchy includes 1.1 MiB of L1d cache and 1.1 MiB of L1i cache across 36 instances each, 36 MiB of L2 cache across 36 instances, and 49.5 MiB of L3 cache across two instances. The platform is also configured with two NUMA nodes, which support efficient memory access during parallel computation.

5. Conclusions

This paper has introduced the Carbon-Aware Flexible Job Shop Scheduling Problem with Tardiness Penalty (CAFJSP-T), a flexible job shop scheduling formulation in which carbon emissions and the tardiness penalty are treated as the two primary decision objectives. To solve this problem, a Policy-Based Rough Optimization with Large Neighborhood Search (Pro-LNS) framework is proposed. The framework combines Proximal Policy Optimization (PPO)-based schedule construction with adaptive Large Neighborhood Search (LNS) refinement, so that a feasible policy-generated schedule is first obtained and then improved through objective-guided destroy-and-repair operations. In this way, Pro-LNS addresses the carbon emission and tardiness trade-off within a unified scheduling framework rather than treating environmental performance and delivery reliability as separate concerns.

The computational results show that Pro-LNS provides strong and consistent performance for CAFJSP-T. When carbon emissions and tardiness penalty are assigned equal priority in the bi-objective formulation, the method achieves a median optimality gap of 6.12% relative to the exact Mixed-Integer Linear Programming (MILP) formulation, with all reported benchmark instances remaining within 14% and 11 of 15 instances remaining within 10%. It also maintains low computational effort, with an average CPU time of 4.08 s and a maximum CPU time of 10.51 s. These results indicate that Pro-LNS can generate

high-quality schedules quickly while preserving feasibility under precedence, eligibility, and machine-capacity constraints. The weight-sensitivity analysis further shows that the scalarization weights provide a useful managerial control mechanism. Carbon emissions remain within a relatively narrow range across the tested weight settings, while tardiness changes more sharply, showing that due-date performance is especially sensitive to the selected policy preference. Thus, the weights can be used to shift the scheduling policy toward carbon reduction, delivery reliability, or a balanced compromise depending on the production context.

The comparative and ablation results clarify the specific advantages of Pro-LNS over analogous methods discussed in the literature review. Compared with exact optimization methods such as MILP and Constraint Programming (CP), Pro-LNS does not provide a proof of optimality by itself, but it offers much faster schedule generation and produces strong solutions with moderate optimality gaps. Compared with the Genetic Algorithm (GA) and other population-based metaheuristics, Pro-LNS benefits from a learned policy that constructs structurally informed initial schedules rather than relying only on population-level exploration. In the representative-instance comparison, Pro-LNS outperforms GA in the weighted scalarized objective while also requiring less computational time. Compared with Advantage Actor-Critic (A2C) and Soft Actor-Critic (SAC) schedulers, Pro-LNS produces better carbon and tardiness outcomes, although A2C and SAC are faster in raw CPU time. Compared with the PPO-only ablation, Pro-LNS achieves better final solution quality, confirming that the LNS phase contributes additional improvement beyond policy construction alone. Across the nine representative instance-weight combinations, Pro-LNS produces the lowest carbon emissions and tardiness penalty, and the Holm-corrected Wilcoxon signed-rank tests show statistically significant improvements over PPO-only, A2C, SAC, and GA. The average weighted-objective improvements are 4.90% over PPO-only, 7.25% over A2C, 8.81% over SAC, and 9.51% over GA.

These findings also reveal important trade-offs and limitations. Pro-LNS is not always the fastest method in terms of pure runtime, since the LNS refinement phase adds computational effort compared with standalone Reinforcement Learning (RL) schedulers such as A2C, SAC, and PPO-only. Its performance also depends on the quality of the PPO-generated initial schedule and on the design of the LNS removal, repair, and acceptance rules. In addition, the scalarized objective requires decision-makers to specify carbon emissions and tardiness priorities in advance, which may be less suitable when a full Pareto front is desired. Exact optimization remains preferable when formal optimality certification is required, while population-based multi-objective algorithms may remain attractive when broad Pareto diversity is the primary goal. Therefore, the main advantage of Pro-LNS is not that it replaces all existing solution families but that it provides an effective hybrid compromise between learned construction, targeted refinement, solution quality, and computational practicality.

The present study is also limited by its modeling assumptions. The CAFJSP-T formulation assumes a deterministic and steady-state production environment with continuously available machines, non-preemptive processing, fixed due dates, deterministic tardiness penalties, and processing-time-based carbon estimation. It does not include machine failures, dynamic job arrivals, uncertain processing times, sequence-dependent setup times, worker-related constraints, transient machine states, or time-varying carbon intensity. Future research should extend the framework to more realistic and dynamic shop-floor settings by incorporating uncertainty, disruption handling, renewable-energy availability, time-dependent carbon factors, machine state transitions, setup effects, transportation resources, and integrated human or workforce constraints. Future work should also examine Pareto-based versions of Pro-LNS, adaptive weight-selection strategies, and broader com-

parisons against additional multi-objective evolutionary and hybrid scheduling methods. Such extensions would improve both the realism of CAFJSP-T and the practical applicability of Pro-LNS as a decision-support tool for sustainable manufacturing scheduling.

Author Contributions: S.S.S. contributed to conceptualization, methodology, formal analysis, investigation, data curation, visualization, writing—original draft, and writing—review and editing. D.G. contributed to supervision, validation, project administration, resources, and writing—review and editing. All authors have read and agreed to the published 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 original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

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.

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

References

- Li, Z.; Rasool, S.; Cavus, M.F.; Shahid, W. Sustaining the future: How green capabilities and digitalization drive sustainability in modern business. *Heliyon* **2024**, *10*, e24158. [[CrossRef](#)] [[PubMed](#)]
- El Mokadem, M.; Khalaf, M. Building sustainable performance through green supply chain management. *Int. J. Product. Perform. Manag.* **2024**, *74*, 203–223. [[CrossRef](#)]
- Mahar, A.S.; Zhang, Y.; Sadiq, B.; Gul, R.F. Sustainability Transformation Through Green Supply Chain Management Practices and Green Innovations in Pakistan’s Manufacturing and Service Industries. *Sustainability* **2025**, *17*, 2204. [[CrossRef](#)]
- Wang, M.; Zhang, G. What motivates firms to adopt a green supply chain and how much does it matter? *Front. Environ. Sci.* **2023**, *11*, 1227008. [[CrossRef](#)]
- Poggi, A.; Di Persio, L.; Ehrhardt, M. Electricity Price Forecasting via Statistical and Deep Learning Approaches: The German Case. *AppliedMath* **2023**, *3*, 316–342. [[CrossRef](#)]
- Narkhede, G.; Chinchani, S.; Narkhede, R.; Chaudhari, T. Role of Industry 5.0 for driving sustainability in the manufacturing sector: An emerging research agenda. *J. Strategy Manag.* **2024**, *ahead-of-print*. [[CrossRef](#)]
- Ghobakhloo, M.; Iranmanesh, M.; Foroughi, B.; Tirkolaee, E.B.; Asadi, S.; Amran, A. Industry 5.0 implications for inclusive sustainable manufacturing: An evidence-knowledge-based strategic roadmap. *J. Clean. Prod.* **2023**, *417*, 138023. [[CrossRef](#)]
- Zheng, R.; Li, Z.; Li, L.; Li, S.; Li, X. Group Technology Empowering Optimization of Mixed-Flow Precast Production in Off-Site Construction. *Environ. Sci. Pollut. Res.* **2024**, *31*, 11781–11800. [[CrossRef](#)]
- Fu, Y.; Gao, K.; Wang, L.; Huang, M.; Liang, Y.; Dong, H. Scheduling Stochastic Distributed Flexible Job Shops Using a Multi-Objective Evolutionary Algorithm with Simulation Evaluation. *Int. J. Prod. Res.* **2024**, *63*, 86–103. [[CrossRef](#)]
- Tang, Y.; Shen, L.; Han, S. Low-Carbon Flexible Job Shop Scheduling Problem Based on Deep Reinforcement Learning. *Sustainability* **2024**, *16*, 4544. [[CrossRef](#)]
- Aghakhani, S.; Rajabi, M.S. A New Hybrid Multi-Objective Scheduling Model for Hierarchical Hub and Flexible Flow Shop Problems. *AppliedMath* **2022**, *2*, 721–737. [[CrossRef](#)]
- Destouet, C.; Tlahig, H.; Bettayeb, B.; Mazari, B. Flexible job shop scheduling problem under Industry 5.0: A survey on human reintegration, environmental consideration and resilience improvement. *J. Manuf. Syst.* **2023**, *67*, 155–173. [[CrossRef](#)]
- Zhou, K.; Tan, C.; Wu, Y.; Yang, B.; Long, X. Research on Low-Carbon Flexible Job Shop Scheduling Problem Based on Improved Grey Wolf Algorithm. *J. Supercomput.* **2024**, *80*, 12123–12153. [[CrossRef](#)]
- Gong, Q.; Li, J.; Jiang, Z.; Wang, Y. A hierarchical integration scheduling method for flexible job shop with green lot splitting. *Eng. Appl. Artif. Intell.* **2024**, *129*, 107595. [[CrossRef](#)]
- Mencaroni, A.; Leyman, P.; Raa, B.; De Vuyst, S.; Claeys, D. Towards net-zero manufacturing: Carbon-aware scheduling for GHG emissions reduction. *J. Clean. Prod.* **2025**, *529*, 146787. [[CrossRef](#)]
- Georgiadis, G.P.; Dimitriadis, C.N.; Georgiadis, M.C. Decarbonizing the Industry Sector: Current Status and Future Opportunities of Energy-Aware Production Scheduling. *Processes* **2025**, *13*, 1941. [[CrossRef](#)]
- Naidu, J.T. A New Algorithm for the Weighted Tardiness Problem. *J. Appl. Bus. Econ.* **2025**, *27*, 24. [[CrossRef](#)]

18. de Athayde Prata, B.; de Abreu, L.R.; Fernandez-Viagas, V. A systematic review of permutation flow shop scheduling with due-date-related objectives. *Comput. Oper. Res.* **2025**, *177*, 106989. [[CrossRef](#)]
19. Xiong, F.; Chen, S.; Xiong, N.; Jing, L. Scheduling distributed heterogeneous non-permutation flowshop to minimize the total weighted tardiness. *Expert Syst. Appl.* **2025**, *272*, 126713. [[CrossRef](#)]
20. Ulucak, M.I.; Gökçen, H. Dynamic Scheduling in Identical Parallel-Machine Environments: A Multi-Purpose Intelligent Utility Approach. *Appl. Sci.* **2025**, *15*, 2483. [[CrossRef](#)]
21. Meng, L.; Cheng, W.; Zhang, B.; Zou, W.; Duan, P. A novel hybrid algorithm of genetic algorithm, variable neighborhood search and constraint programming for distributed flexible job shop scheduling problem. *Int. J. Ind. Eng. Comput.* **2024**, *15*, 813–832. [[CrossRef](#)]
22. Nessari, S.; Tavakkoli-Moghaddam, R.; Bakhshi-Khaniki, H.; Bozorgi-Amiri, A. A hybrid simheuristic algorithm for solving bi-objective stochastic flexible job shop scheduling problems. *Decis. Anal. J.* **2024**, *11*, 100485. [[CrossRef](#)]
23. Seck-Tuoh-Mora, J.C.; Escamilla-Serna, N.J.; Montiel-Arrieta, L.J.; Barragán-Vite, I.; Medina-Marín, J. A Global Neighborhood with Hill-Climbing Algorithm for Fuzzy Flexible Job Shop Scheduling Problem. *Mathematics* **2022**, *10*, 4233. [[CrossRef](#)]
24. Berterottière, L.; Dauzère-Pérés, S.; Yugma, C. Flexible job-shop scheduling with transportation resources. *Eur. J. Oper. Res.* **2023**, *312*, 890–909. [[CrossRef](#)]
25. Yang, S.; Meng, L.; Ullah, S.; Zhang, B.; Sang, H.; Duan, P. MILP Modeling and Optimization of Multi-Objective Three-Stage Flexible Job Shop Scheduling Problem With Assembly and AGV Transportation. *IEEE Access* **2025**, *13*, 25369–25386. [[CrossRef](#)]
26. Fernandes, J.; Homayouni, S.; Fontes, D. Energy-Efficient Scheduling in Job Shop Manufacturing Systems: A Literature Review. *Sustainability* **2022**, *14*, 6264. [[CrossRef](#)]
27. Li, Z.; Chen, Y.-H. Minimizing the makespan and carbon emissions in the green flexible job shop scheduling problem with learning effects. *Sci. Rep.* **2023**, *13*, 6369. [[CrossRef](#)]
28. Jia, S.; Yang, Y.; Li, S.; Wang, S.; Li, A.; Cai, W.; Liu, Y.; Hao, J.; Hu, L. The Green Flexible Job-Shop Scheduling Problem Considering Cost, Carbon Emissions, and Customer Satisfaction under Time-of-Use Electricity Pricing. *Sustainability* **2024**, *16*, 2443. [[CrossRef](#)]
29. Park, M.-J.; Ham, A. Energy-aware flexible job shop scheduling under time-of-use pricing. *Int. J. Prod. Econ.* **2022**, *248*, 108507. [[CrossRef](#)]
30. Xu, G.; Bao, Q.; Zhang, H. Multi-objective green scheduling of integrated flexible job shop and automated guided vehicles. *Eng. Appl. Artif. Intell.* **2023**, *126*, 106864. [[CrossRef](#)]
31. Tang, H.; Huang, J.; Ren, C.; Shao, Y.; Lu, J. Integrated scheduling of multi-objective lot-streaming hybrid flowshop with AGV based on deep reinforcement learning. *Int. J. Prod. Res.* **2024**, *63*, 1275–1303. [[CrossRef](#)]
32. Deliktas, D.; Özcan, E.; Ustun, O.; Torkul, O. Evolutionary algorithms for multi-objective flexible job shop cell scheduling. *Appl. Soft Comput.* **2021**, *113*, 107890. [[CrossRef](#)]
33. Lei, D.; Li, M.; Wang, L. A Two-Phase Meta-Heuristic for Multiobjective Flexible Job Shop Scheduling Problem With Total Energy Consumption Threshold. *IEEE Trans. Cybern.* **2019**, *49*, 1097–1109. [[CrossRef](#)] [[PubMed](#)]
34. Luan, F.; Zhao, H.; Liu, S.; He, Y.; Tang, B. Enhanced NSGA-II for multi-objective energy-saving flexible job shop scheduling. *Sustain. Comput. Inform. Syst.* **2023**, *39*, 100901. [[CrossRef](#)]
35. Ojsteršek, R.; Tang, M.; Buchmeister, B. Due date optimization in multi-objective scheduling of flexible job shop production. *Adv. Prod. Eng. Manag.* **2020**, *15*, 481–492. [[CrossRef](#)]
36. Wu, Z.; Fan, H.; Sun, Y.; Peng, M. Efficient Multi-Objective Optimization on Dynamic Flexible Job Shop Scheduling Using Deep Reinforcement Learning Approach. *Processes* **2023**, *11*, 2018. [[CrossRef](#)]
37. Zhao, L.; Fan, J.; Zhang, C.; Shen, W.; Jing, Z. A DRL-Based Reactive Scheduling Policy for Flexible Job Shops With Random Job Arrivals. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 2912–2923. [[CrossRef](#)]
38. Chen, Y.; Liao, X.; Chen, G.; Hou, Y. Dynamic Intelligent Scheduling in Low-Carbon Heterogeneous Distributed Flexible Job Shops with Job Insertions and Transfers. *Sensors* **2024**, *24*, 2251. [[CrossRef](#)]
39. Wang, Z.; He, M.; Wu, J.; Chen, H.; Cao, Y. An improved MOEA/D for low-carbon many-objective flexible job shop scheduling problem. *Comput. Ind. Eng.* **2024**, *188*, 109926. [[CrossRef](#)]
40. Piroozfard, H.; Wong, K.Y.; Wong, W.P. Minimizing total carbon footprint and total late work criterion in flexible job shop scheduling by using an improved multi-objective genetic algorithm. *Resour. Conserv. Recycl.* **2016**, *128*, 267–283. [[CrossRef](#)]
41. Wei, Z.; Liao, W.; Zhang, L. Hybrid energy-efficient scheduling measures for flexible job-shop problem with variable machining speeds. *Expert Syst. Appl.* **2022**, *197*, 116785. [[CrossRef](#)]
42. Zhang, F.; Li, R.; Gong, W. Deep reinforcement learning-based memetic algorithm for energy-aware flexible job shop scheduling with multi-AGV. *Comput. Ind. Eng.* **2024**, *189*, 109917. [[CrossRef](#)]
43. Meng, L.; Zhang, C.; Ren, Y.; Zhang, B.; Lv, C. Mixed-integer linear programming and constraint programming formulations for solving distributed flexible job shop scheduling problem. *Comput. Ind. Eng.* **2020**, *142*, 106347. [[CrossRef](#)]
44. Ji, B.; Zhang, S.; Yu, S.; Zhang, B. Mathematical Modeling and a Novel Heuristic Method for Flexible Job-Shop Batch Scheduling Problem with Incompatible Jobs. *Sustainability* **2023**, *15*, 1954. [[CrossRef](#)]

45. Fan, J.; Zhang, C.; Shen, W.; Gao, L. A matheuristic for flexible job shop scheduling problem with lot-streaming and machine reconfigurations. *Int. J. Prod. Res.* **2022**, *61*, 6565–6588. [\[CrossRef\]](#)
46. Mei, Z.; Lu, Y.; Lv, L. Research on Multi-Objective Low-Carbon Flexible Job Shop Scheduling Based on Improved NSGA-II. *Machines* **2024**, *12*, 590. [\[CrossRef\]](#)
47. Sang, Y.; Tan, J. Many-Objective Flexible Job Shop Scheduling Problem with Green Consideration. *Energies* **2022**, *15*, 1884. [\[CrossRef\]](#)
48. Li, R.; Gong, W.; Wang, L.; Lu, C.; Jiang, S. Two-stage knowledge-driven evolutionary algorithm for distributed green flexible job shop scheduling with type-2 fuzzy processing time. *Swarm Evol. Comput.* **2022**, *74*, 101139. [\[CrossRef\]](#)
49. Lei, D.; Zheng, Y.; Guo, X. A shuffled frog-leaping algorithm for flexible job shop scheduling with the consideration of energy consumption. *Int. J. Prod. Res.* **2016**, *55*, 3126–3140. [\[CrossRef\]](#)
50. Jiang, T.; Zhu, H.; Deng, G. Improved African buffalo optimization algorithm for the green flexible job shop scheduling problem considering energy consumption. *J. Intell. Fuzzy Syst.* **2020**, *38*, 4573–4589. [\[CrossRef\]](#)
51. Peng, Z.; Zhang, H.; Tang, H.; Feng, Y.; Yin, W. Research on flexible job-shop scheduling problem in green sustainable manufacturing based on learning effect. *J. Intell. Manuf.* **2021**, *33*, 1725–1746. [\[CrossRef\]](#)
52. Ren, W.; Wen, J.; Yan, Y.; Hu, Y.; Guan, Y.; Li, J. Multi-objective optimisation for energy-aware flexible job-shop scheduling problem with assembly operations. *Int. J. Prod. Res.* **2020**, *59*, 7216–7231. [\[CrossRef\]](#)
53. Tan, W.; Yuan, X.; Huang, G.; Liu, Z. Low-carbon joint scheduling in flexible open-shop environment with constrained automatic guided vehicle by multi-objective particle swarm optimization. *Appl. Soft Comput.* **2021**, *111*, 107695. [\[CrossRef\]](#)
54. Yang, X.; Zhang, J.; Zhang, N.; Li, Y. Low Carbon Multi-Objective Shop Scheduling Based On Genetic and Variable Neighborhood Algorithm. *J. Phys. Conf. Ser.* **2020**, *1574*, 012155. [\[CrossRef\]](#)
55. Hayat, I.; Tariq, A.; Shahzad, W.; Masud, M.; Ahmed, S.; Ali, M.; Zafar, A. Hybridization of Particle Swarm Optimization with Variable Neighborhood Search and Simulated Annealing for Improved Handling of the Permutation Flow-Shop Scheduling Problem. *Systems* **2023**, *11*, 221. [\[CrossRef\]](#)
56. Wang, L.; Pan, Z.; Wang, J. A Review of Reinforcement Learning Based Intelligent Optimization for Manufacturing Scheduling. *Complex Syst. Model. Simul.* **2021**, *1*, 257–270. [\[CrossRef\]](#)
57. Khadivi, M.; Charter, T.; Yaghoubi, M.; Jalayer, M.; Ahang, M.; Shojaeinasab, A.; Najjaran, H. Deep reinforcement learning for machine scheduling: Methodology, the state-of-the-art, and future directions. *arXiv* **2023**, arXiv:2310.03195. [\[CrossRef\]](#)
58. Liu, R.; Piplani, R.; Toro, C. Deep reinforcement learning for dynamic scheduling of a flexible job shop. *Int. J. Prod. Res.* **2022**, *60*, 4049–4069. [\[CrossRef\]](#)
59. Yi, W.; Chen, N.; Chen, Y.; Pei, Z. An improved deep Q-network for dynamic flexible job shop scheduling with limited maintenance resources. *Int. J. Prod. Res.* **2025**, *63*, 9112–9133. [\[CrossRef\]](#)
60. Song, W.; Chen, X.; Li, Q.; Cao, Z. Flexible Job-Shop Scheduling via Graph Neural Network and Deep Reinforcement Learning. *IEEE Trans. Ind. Inform.* **2023**, *19*, 1600–1610. [\[CrossRef\]](#)
61. Huang, J.-P.; Gao, L.; Li, X. An end-to-end deep reinforcement learning method based on graph neural network for distributed job-shop scheduling problem. *Expert Syst. Appl.* **2023**, *238*, 121756. [\[CrossRef\]](#)
62. Wang, S.; Li, J.; Tang, H.; Wang, J. CEA-FJSP: Carbon emission-aware flexible job-shop scheduling based on deep reinforcement learning. *Front. Environ. Sci.* **2022**, *10*, 1059451. [\[CrossRef\]](#)
63. van Hezewijk, L.; Dellaert, N.; Van Woensel, T.; Gademann, N. Using the proximal policy optimisation algorithm for solving the stochastic capacitated lot sizing problem. *Int. J. Prod. Res.* **2022**, *61*, 1955–1978. [\[CrossRef\]](#)
64. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347. [\[CrossRef\]](#)
65. Park, J.; Chun, J.; Kim, S.; Kim, Y.; Park, J. Learning to schedule job-shop problems: Representation and policy learning using graph neural network and reinforcement learning. *Int. J. Prod. Res.* **2021**, *59*, 3360–3377. [\[CrossRef\]](#)
66. Lei, K.; Guo, P.; Zhao, W.; Wang, Y.; Qian, L.; Meng, X.; Tang, L. A multi-action deep reinforcement learning framework for flexible Job-shop scheduling problem. *Expert Syst. Appl.* **2022**, *205*, 117796. [\[CrossRef\]](#)
67. Tang, H.; Dong, J. Solving Flexible Job-Shop Scheduling Problem with Heterogeneous Graph Neural Network Based on Relation and Deep Reinforcement Learning. *Machines* **2024**, *12*, 584. [\[CrossRef\]](#)
68. Ding, L.; Guan, Z.; Rauf, M.; Yue, L. Multi-policy deep reinforcement learning for multi-objective multiplicity flexible job shop scheduling. *Swarm Evol. Comput.* **2024**, *87*, 101550. [\[CrossRef\]](#)
69. Luo, S.; Zhang, L.; Fan, Y. Real-Time Scheduling for Dynamic Partial-No-Wait Multiobjective Flexible Job Shop by Deep Reinforcement Learning. *IEEE Trans. Autom. Sci. Eng.* **2022**, *19*, 3020–3038. [\[CrossRef\]](#)
70. Liu, X.; Han, L.; Kang, L.; Liu, J.; Miao, H. Preference learning based deep reinforcement learning for flexible job shop scheduling problem. *Complex Intell. Syst.* **2025**, *11*, 144. [\[CrossRef\]](#)
71. Pan, Z.; Wang, L.; Wang, J.-J.; Lu, J. Deep Reinforcement Learning Based Optimization Algorithm for Permutation Flow-Shop Scheduling. *IEEE Trans. Emerg. Top. Comput. Intell.* **2023**, *7*, 983–994. [\[CrossRef\]](#)

72. Pan, Z.; Wang, L.; Dong, C.; Chen, J. A Knowledge-Guided End-to-End Optimization Framework Based on Reinforcement Learning for Flow Shop Scheduling. *IEEE Trans. Ind. Inform.* **2024**, *20*, 1853–1861. [\[CrossRef\]](#)
73. Johnn, S.; Darvari, V.; Handl, J.; Kalcsics, J. A Graph Reinforcement Learning Framework for Neural Adaptive Large Neighbourhood Search. *Comput. Oper. Res.* **2024**, *172*, 106791. [\[CrossRef\]](#)
74. Oren, J.; Ross, C.; Lefarov, M.; Richter, F.; Taitler, A.; Feldman, Z.; Daniel, C.; Di Castro, D. SOLO: Search Online, Learn Offline for Combinatorial Optimization Problems. *Proc. Int. Symp. Comb. Search* **2021**, *12*, 97–105. [\[CrossRef\]](#)
75. Shi, J.; Liu, W.; Yang, J. An Enhanced Multi-Objective Evolutionary Algorithm with Reinforcement Learning for Energy-Efficient Scheduling in the Flexible Job Shop. *Processes* **2024**, *12*, 1976. [\[CrossRef\]](#)
76. Yao, Y.; Li, X.; Gao, L. A DQN-based memetic algorithm for energy-efficient job shop scheduling problem with integrated limited AGVs. *Swarm Evol. Comput.* **2024**, *87*, 101544. [\[CrossRef\]](#)
77. Li, R.; Gong, W.; Lu, C. A reinforcement learning based RMOEA/D for bi-objective fuzzy flexible job shop scheduling. *Expert Syst. Appl.* **2022**, *203*, 117380. [\[CrossRef\]](#)
78. Mora, J.; Escamilla-Serna, N.; Marín, J.; Hernández-Romero, N.; Barragán-Vite, I.; Corona-Armenta, J. A global-local neighborhood search algorithm and tabu search for flexible job shop scheduling problem. *PeerJ Comput. Sci.* **2020**, *7*, e574. [\[CrossRef\]](#)
79. Xie, J.; Li, X.; Gao, L.; Gui, L. A hybrid genetic tabu search algorithm for distributed flexible job shop scheduling problems. *J. Manuf. Syst.* **2023**, *71*, 82–94. [\[CrossRef\]](#)
80. Xie, J.; Teng, Y.; Gao, L.; Li, X.; Zhang, C. An efficient and stable intelligent scheduling algorithm based on hybrid neighbourhood structure for flexible job shop scheduling problem benchmarks. *Int. J. Prod. Res.* **2025**, *63*, 7921–7935. [\[CrossRef\]](#)
81. Birgin, E.; Riveaux, J.; Ronconi, D. Energy-aware flexible job shop scheduling problem with nonlinear routes and position-based learning effect. *Int. Trans. Oper. Res.* **2025**, *33*, 860–891. [\[CrossRef\]](#)
82. Röpke, S.; Pisinger, D. An Adaptive Large Neighborhood Search Heuristic for the Pickup and Delivery Problem with Time Windows. *Transp. Sci.* **2006**, *40*, 455–472. [\[CrossRef\]](#)
83. Rifai, A.; Nguyen, H.; Dawal, S. Multi-objective adaptive large neighborhood search for distributed reentrant permutation flow shop scheduling. *Appl. Soft Comput.* **2016**, *40*, 42–57. [\[CrossRef\]](#)
84. Cota, L.; Guimarães, F.; Ribeiro, R.; Meneghini, I.; Oliveira, F.; Souza, M.; Siarry, P. An adaptive multi-objective algorithm based on decomposition and large neighborhood search for a green machine scheduling problem. *Swarm Evol. Comput.* **2019**, *51*, 100601. [\[CrossRef\]](#)
85. Liu, J.; Sun, B.; Li, G.; Chen, Y. Multi-objective adaptive large neighbourhood search algorithm for dynamic flexible job shop schedule problem with transportation resource. *Eng. Appl. Artif. Intell.* **2024**, *132*, 107917. [\[CrossRef\]](#)
86. Cao, S.; Li, R.; Gong, W.; Lu, C. Inverse model and adaptive neighborhood search based cooperative optimizer for energy-efficient distributed flexible job shop scheduling. *Swarm Evol. Comput.* **2023**, *83*, 101419. [\[CrossRef\]](#)
87. Hariri, F.; Santosa, B. A Hybrid Genetic Algorithm and Adaptive Large Neighborhood Search for Flexible Job Shop Scheduling with Fuzzy Processing Time. In *2025 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*; IEEE: Piscataway, NJ, USA, 2025; pp. 86–90. [\[CrossRef\]](#)
88. Chung, K.; Lee, C.; Tsang, Y. Neural combinatorial optimization with reinforcement learning in industrial engineering: A survey. *Artif. Intell. Rev.* **2025**, *58*, 130. [\[CrossRef\]](#)
89. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-Baselines3: Reliable Reinforcement Learning Implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8. Available online: <http://jmlr.org/papers/v22/20-1364.html> (accessed on 11 May 2026).
90. Chen, Z.; Zhang, K.; Liu, P.; Xin, G.; Sun, Z.; Tao, Z.; Zhang, Y.; Ji, W.; Lu, Y.; Jia, L.; et al. Worst-Case Soft Actor-Critic-Based Safe Reinforcement Learning Method for Nonlinear Constrained Waterflood Reservoir Production Optimization. *SPE J.* **2025**, *30*, 7745–7766. [\[CrossRef\]](#)
91. Liang, Y.; Sun, Y.; Zheng, R.; Huang, F. Efficient adversarial training without attacking: Worst-case-aware robust reinforcement learning. *arXiv* **2022**, arXiv:2210.05927. <https://arxiv.org/abs/2210.05927>.
92. Liu, C.; Chang, C.; Tseng, C. Actor-Critic Deep Reinforcement Learning for Solving Job Shop Scheduling Problems. *IEEE Access* **2020**, *8*, 71752–71762. [\[CrossRef\]](#)
93. Liu, C.; Huang, T. Dynamic Job-Shop Scheduling Problems Using Graph Neural Network and Deep Reinforcement Learning. *IEEE Trans. Syst. Man. Cybern. Syst.* **2023**, *53*, 6836–6848. [\[CrossRef\]](#)
94. Ruiz, J.; Mula, J.; Escoto, R. Job shop smart manufacturing scheduling by deep reinforcement learning. *J. Ind. Inf. Integr.* **2024**, *38*, 100582. [\[CrossRef\]](#)
95. Wang, R.; Jing, Y.; Gu, C.; He, S.; Chen, J. End-to-End Multitarget Flexible Job Shop Scheduling With Deep Reinforcement Learning. *IEEE Internet Things J.* **2025**, *12*, 4420–4434. [\[CrossRef\]](#)
96. Zhang, L.; Feng, Y.; Xiao, Q.; Xu, Y.; Li, D.; Yang, D.; Yang, Z. Deep reinforcement learning for dynamic flexible job shop scheduling problem considering variable processing times. *J. Manuf. Syst.* **2023**, *71*, 257–273. [\[CrossRef\]](#)

97. Zhou, Y.; Jiang, J.; Shi, Q.; Fu, M.; Zhang, Y.; Chen, Y.; Zhou, L. GA-HPO PPO: A Hybrid Algorithm for Dynamic Flexible Job Shop Scheduling. *Sensors* **2025**, *25*, 6736. [\[CrossRef\]](#)
98. Chen, Y.; Zhang, F.; Liu, Z. Adaptive Advantage Estimation for Actor-Critic Algorithms. In *2021 International Joint Conference on Neural Networks (IJCNN)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–8. [\[CrossRef\]](#)
99. Chen, Y.; Zhang, F.; Liu, Z. Adaptive bias-variance trade-off in advantage estimator for actor-critic algorithms. *Neural Netw.* **2023**, *169*, 764–777. [\[CrossRef\]](#)
100. Li, Y.; Yu, C. Flexible Job Shop Scheduling with Job Precedence Constraints: A Deep Reinforcement Learning Approach. *J. Manuf. Mater. Process.* **2025**, *9*, 216. [\[CrossRef\]](#)
101. Wang, Z.; Liao, W. Smart scheduling of dynamic job shop based on discrete event simulation and deep reinforcement learning. *J. Intell. Manuf.* **2023**, *35*, 2593–2610. [\[CrossRef\]](#)
102. Zhang, Y.; Zhu, H.; Tang, D.; Zhou, T.; Gui, Y. Dynamic job shop scheduling based on deep reinforcement learning for multi-agent manufacturing systems. *Robot. Comput.-Integr. Manuf.* **2022**, *78*, 102412. [\[CrossRef\]](#)
103. Yu, C.; Velu, A.; Vinitzky, E.; Wang, Y.; Bayen, A.; Wu, Y. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. *Adv. Neural Inf. Process. Syst.* **2021**, *35*, 24611–24624. [\[CrossRef\]](#)
104. Behnke, D.; Geiger, M.J. *Test Instances for the Flexible Job Shop Scheduling Problem with Work Centers*; Research Paper; Helmut-Schmidt-Universität, Lehrstuhl für Betriebswirtschaftslehre, insbes; Logistik-Management: Hamburg, Germany, 2012.
105. Lu, Y.; Zhu, Q.; Tian, C.; He, E.; Zhang, T. Low-Carbon and Energy-Efficient Dynamic Flexible Job Shop Scheduling Method Towards Renewable Energy Driven Manufacturing. *Machines* **2026**, *14*, 88. [\[CrossRef\]](#)
106. Cinar, D.; Topcu, Y.I.; Oliveira, J.A. A priority-based genetic algorithm for a flexible job shop scheduling problem. *J. Ind. Manag. Optim.* **2016**, *12*, 1391. [\[CrossRef\]](#)
107. Deb, K.; Agrawal, R.B. Simulated binary crossover for continuous search space. *Complex Syst.* **1995**, *9*, 115–148. Available online: http://www.complex-systems.com/abstracts/v09_i02_a02/ (accessed on 11 May 2026).
108. Singh, S.S.; Joshi, R.; Gupta, D. An Advantage Actor-Critic Approach for Energy-Conscious Scheduling in Flexible Job Shops. *J. Artif. Intell.* **2025**, *7*, 177–203. [\[CrossRef\]](#)
109. Singh, S.S.; Gupta, D.P. A Soft Actor-Critic Approach for Energy-Conscious Flexible Job Shop Scheduling Incorporating Machine Usage Constraints and Job Release Times. *J. Manag. Eng. Integr.* **2025**, *18*, 116–125. [\[CrossRef\]](#)

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.