buildings

MDPI

Article

Knowledge-Driven Interval Multi-Objective Scheduling for Green Construction Under Time-Varying Carbon Emission Factors

Yajuan Deng [1], Zhang Feng [2], Weilun Tao [3], Qian Meng [4], Chongying Ling [5,*] and Hanwen Cui [6,*]

[1] Business School, Sichuan University, Chengdu 610064, China
[2] School of Civil Engineering, Southeast University, Nanjing 210096, China
[3] School of Arts, Sciences and Education, Florida International University, Miami, FL 33199, USA
[4] School of Architecture and Design, University of Technology Sydney, Sydney, NSW 2007, Australia
[5] School of Physics and Telecommunication Engineering, Yulin Normal University, Yulin 537000, China
[6] School of Engineering and Built Environment, Griffith University, Nathan, QLD 4222, Australia
* Correspondence: cyling@ylu.edu.cn (C.L.); h.cui@griffith.edu.au (H.C.)

Abstract

Reducing carbon emissions during construction is essential for meeting dual carbon targets. Current green scheduling methods assume fixed emission factors, overlooking time-dependent variations driven by grid peak-valley patterns. Under interval duration uncertainty coupled with tight dynamic carbon budgets, conventional algorithms struggle with sparse feasible solutions and slow Pareto front convergence. We formulate a bi-objective interval RCPSP model with time-varying carbon emission factors that minimizes both interval makespan and total carbon emissions. A possibility degree measure converts scalar carbon budgets into linearized hard constraints. To solve this NP-hard problem, we propose the Knowledge-Driven Interval Multi-Objective Evolutionary Algorithm (KD-IMOEA), which integrates four components: Knowledge-Driven Initialization (KDI), Adaptive Time-window Carbon-aware Decoding (TCD), Carbon Budget-aware Repair Mutation (CBM), and Interval Pareto Elite Archive (IPA), forming an end-to-end carbon-aware optimization pipeline. We validate KD-IMOEA on J30 through J120 benchmark instances; results show it outperforms four established algorithms, including NSGA-II, in both convergence and distribution, with hypervolume (HV) gains up to 6.3%. A green building case study confirms that KD-IMOEA exploits spatiotemporal decoupling to identify float time and assign energy-intensive machinery to lower-carbon operating profiles. At the optimal compromise makespan of 169.5 days, this strategy cuts carbon emissions by 3.07% over traditional baselines, enabling management-driven emission savings without extending project duration.

Keywords: resource-constrained project scheduling; interval uncertainty; carbon-constrained optimization; multi-objective evolutionary algorithm; possibility degree; construction carbon emission; Pareto dominance

Check for updates

Academic Editor: Ahmed Senouci

Received: 16 April 2026
Revised: 9 May 2026
Accepted: 14 May 2026
Published: 16 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) license.

1. Introduction

Construction ranks among the largest contributors to global carbon emissions. The United Nations Environment Programme [1] reports that buildings and construction consume 32% of global final energy and produce 34% of energy-related CO_2 emissions, totaling a record 9.8 Gt in 2023. In China, building life-cycle emissions exceed 50% of the national

total [2], largely because the construction phase relies on fossil-fuel-driven heavy machinery. China's dual carbon targets (peaking by 2030, neutrality by 2060) and national standards GB/T 50378-2019 [3] and GB/T 51366-2019 [4] now set explicit caps on construction-phase emissions. Current emission reduction measures heavily rely on equipment upgrades and technological innovations. For instance, intelligent structural health monitoring via machine learning [5,6], along with low-carbon material substitutions like eco-friendly cementitious composites [7,8] and FRP-confined concrete members [9,10], have been widely adopted to lower embedded carbon. While these hardware and material-driven advancements significantly enhance structural performance, they leave the micro-level scheduling and sequencing of construction activities largely untapped as a management-driven emission reduction lever.

How activities are sequenced, when they start, and how resources are allocated together determine mechanical equipment intensity across time periods. Grid carbon emission factors differ markedly between peak and valley hours, with peak-period factors roughly 1.5–2.0 times the valley-period values [11]. Embedding these time-varying factors into scheduling optimization and using algorithms to assign energy-intensive activities to lower-carbon operating profiles can cut emissions through better management alone, without purchasing new equipment. In this study, such assignments are represented as start-day commitments and low-carbon operating-profile recommendations rather than deterministic hour-by-hour dispatch instructions. The detailed intra-day deployment of equipment and crews remains a tactical-layer decision made by the project team under weather, safety, labor, and regulatory constraints. This gives budget-constrained firms a practical path to satisfying both schedule targets and green certification.

This engineering need maps directly to the resource-constrained project scheduling problem (RCPSP), which assigns activity start times under precedence and renewable-resource capacity constraints to minimize makespan. RCPSP is NP-hard [12], and instances beyond small scale require metaheuristics. Multi-objective evolutionary algorithms, notably NSGA-II [13], NSGA-III [14], MOEA/D [15], and SPEA2 [16], have been applied to RCPSP variants that balance makespan against cost or resource leveling [17]. Carbon emissions have recently entered scheduling models as an additional objective. Aramesh et al. [18] added CO_2 to multi-mode RCPSP under interval-valued fuzzy uncertainty. Razi and Ansari [19] combined machine learning with metaheuristics for four-way time-cost-energy-CO_2 trade-offs. Rahman and Han [20] built a decarbonization-focused production scheduling framework, confirming that activity-level sequencing contributes meaningfully to emission reduction. Ding et al. [21] surveyed the past decade of RCPSP research and identified carbon optimization and uncertainty handling as two fast-growing directions.

Activity durations in construction are affected by weather, material supply, and labor availability, making precise prediction difficult at the planning stage. Uncertainty modeling has accordingly become a major branch of scheduling research. Three approaches dominate. Stochastic programming assumes durations follow known probability distributions and optimizes expected makespan or conditional value-at-risk (CVaR). Fu et al. [22] proposed a distributionally robust RCPSP model that minimizes CVaR when duration distributions are not fully known, improving schedule robustness. Stochastic methods, however, need sufficient historical data to fit distributions, and accuracy degrades when past cases are scarce or conditions differ substantially.

Fuzzy programming uses fuzzy numbers or interval-valued fuzzy numbers to capture subjective duration uncertainty. Patoghi and Mousavi [23] modeled activity duration and material ordering uncertainty with interval type-2 fuzzy numbers in a multi-project RCPSP. Fuzzy methods reduce dependence on precise distributions, but constructing membership functions still demands strong subjective judgment. Interval programming requires only

optimistic and pessimistic duration bounds, representing durations as closed intervals. Hazır and Ulusoy [24] noted in their systematic review that interval methods strike a good balance between data requirements and modeling complexity, suiting construction scenarios where historical data are scarce.

Interval methods have been further validated in complex multi-objective scheduling. Ghasemi et al. [25] developed a scheduling model for time-cost-quality trade-offs that handles multi-mode resources and interval uncertainty. Hatami-Moghaddam et al. [26] built a robust multi-objective resource-constrained scheduling framework for duration fluctuations and time-window constraints. The value of embedding uncertainty in scheduling frameworks is now broadly accepted. Li et al. [27] confirmed that uncertainty modeling improves the executability of resulting schedules. Aghileh et al. [28] reviewed methods for handling uncertainty and resource flexibility in multi-project scheduling, noting that combining uncertainty modeling with multi-objective optimization remains an open research area.

Despite these advances, jointly addressing green scheduling and uncertainty under realistic dynamic constraints still faces three bottlenecks. At the modeling level, most studies assume constant carbon emission factors throughout the project cycle, ignoring time-varying characteristics from grid peak-valley structures. This static assumption breaks the link between energy-intensive machinery operations and external low-carbon windows, preventing fine-grained shift scheduling. At the algorithm level, when interval uncertainty couples tightly with stringent carbon budget constraints, the feasible search space shrinks dramatically. Standard multi-objective evolutionary algorithms such as NSGA-II rely on random initial populations and blind crossover-mutation operators with no awareness of carbon budget consumption, resulting in low initial feasibility rates and slow Pareto front convergence. At the evaluation level, interval multi-objective Pareto dominance is typically based on midpoints alone. While computationally simpler, this discards risk information carried by interval widths and cannot distinguish schedules with equal midpoint objectives but different uncertainty profiles, limiting support for risk-sensitive decisions.

To address these gaps, we introduce time-varying carbon emission factors into the uncertainty scheduling framework, formulate a bi-objective interval RCPSP under carbon constraints, and develop KD-IMOEA for efficient solution. The main contributions are as follows:

(1) We construct a bi-objective interval scheduling model with time-varying carbon emission factors. The model uses interval numbers for duration uncertainty and minimizes both interval makespan and interval total carbon emissions. We apply the possibility degree measure to link interval carbon emissions with a scalar carbon budget as a hard constraint, linearizing the nonlinear time-dependent emission calculation into precomputed matrices. This provides a rigorous framework for coupling uncertainty with green certification standards.

(2) We propose KD-IMOEA with four integrated carbon-aware components. To counter the sharp feasibility decline under tight carbon budgets, the algorithm embeds carbon awareness across four evolutionary stages: KDI raises initial population feasibility through carbon-guided construction rules; TCD biases early-stage decoding toward low-emission time periods; CBM repairs infeasible offspring by reallocating high-carbon activities to low-emission periods; IPA preserves historically optimal interval Pareto solutions in an external archive. Benchmark experiments show that KD-IMOEA consistently and significantly outperforms four baseline algorithms in both convergence and distribution uniformity.

(3) We reveal a spatiotemporal decoupling mechanism that achieves management-driven emission savings at zero hardware cost. A green building case study confirms that KD-IMOEA can identify energy-intensive activities with float time and assign them

to lower-carbon operating profiles without delaying overall makespan. This micro-level asymmetric scheduling releases over 3% carbon safety margin, offering budget-constrained contractors an algorithm-driven tool for carbon compliance that reduces quota violation risk without additional equipment investment.

The remainder of this paper is organized as follows. Section 2 formulates the bi-objective interval RCPSP model. Section 3 presents KD-IMOEA and its four carbon-aware components. Section 4 reports benchmark comparisons, ablation studies, and parameter sensitivity analysis. Section 5 presents a green building case study. Section 6 discusses practical implications, applicability boundaries, and project-management workflows. Section 7 concludes the paper and outlines future research directions.

2. Problem Formulation

2.1. Problem Description

Consider a green building project consisting of n construction activities, represented by an activity-on-node (AON) network $G(N, A)$. In the node set $N = \{0, 1, \ldots, n, n+1\}$, 0 and $n+1$ are virtual start and finish activities with zero duration; the arc set A encodes precedence relations. If $(i, j) \in A$, then activity i must be completed before activity j starts (finish-to-start constraint). The project planning horizon is discretized into T equal-length time periods, with the period set denoted as $\mathcal{T} = \{1, 2, \ldots, T\}$. A period is defined as the smallest temporal interval at which scheduling decisions—activity start times S_j, resource availabilities $R_{k\tau}$, and emission factors $EF_{k\tau}$—are made. Throughout this paper, a period corresponds to one working day, following the standard RCPSP convention. This day-level instantiation applies to both the benchmark experiments of Section 4 (where emission factors $EF_{k\tau}$ are parameterized at day-level resolution) and the green building case study of Section 5 (where intra-day grid-emission heterogeneity is captured at the emission-factor level via grid-category multipliers, rather than through finer temporal subdivision; see Section 5.1 for details). This unified day-level abstraction reflects the strategic-layer scope of the proposed framework: the model commits start-day decisions for each activity, while intra-day deployment of crews and equipment is determined at the tactical layer by the project team based on real-time conditions—a separation that integrates with established master-schedule workflows and is discussed in detail in Section 6.2. Activity durations d_j are stated in calendar days throughout the paper.

The project uses K types of renewable resources (tower cranes, labor crews, etc.). The available capacity of resource k in period t is denoted as R_{kt}, whose time dependence reflects external constraints such as nighttime construction bans and rainy season shutdowns. The fixed demand of activity j for resource k per period is denoted as r_{jk}. We adopt the non-preemption assumption: once an activity starts, it must be executed continuously until completion without interruption.

Construction activity durations are affected by multiple factors such as weather, material supply, and labor skill, making them difficult to predict precisely during the planning phase. We use interval numbers to characterize duration uncertainty: for activity j, only the optimistic duration lower bound d_j^- and pessimistic duration upper bound d_j^+ need to be estimated, with the interval duration:

$$\widetilde{d}_j = \left[d_j^-,\, d_j^+\right], \quad d_j^- \le d_j^+, \quad \forall j \in N \tag{1}$$

representing the possible range of duration values. Compared to stochastic programming, interval programming does not require presetting probability distributions and has lower data requirements in scenarios where construction project historical data are limited, making it more suitable for actual engineering conditions.

According to the Standard for Carbon Emission Calculation of Buildings (GB/T 51366-2019), the carbon emission intensity of energy sources such as electricity and diesel consumed by mechanical equipment during the construction phase exhibits time dependence: the electricity carbon emission factor varies with the energy structure (renewable energy proportion) of the grid during peak and valley periods, and mechanical equipment also experiences increased carbon emissions per unit energy consumption during extreme temperature periods due to reduced thermal efficiency. To this end, we define the carbon emission factor $EF_{k\tau}$ of resource k in period τ (unit: $kgCO_2e$/(unit·period)). Following the day-level period instantiation in Section 2.1, $EF_{k\tau}$ represents the day-level effective emission factor of resource k on working day τ. In the benchmark experiments (Section 4), $EF_{k\tau}$ varies across days following a representative grid-emission profile; in the green building case study (Section 5.1), intra-day grid heterogeneity is captured by assigning each working day to one of three operating-profile categories—peak, mid-level, and valley—with corresponding day-level factors $EF_{k\tau} = \overline{EF}_k \cdot m_\tau$, where $\overline{EF}_k$ is the resource-specific base factor and $m_\tau \in \{1.2, 1.0, 0.7\}$ is the grid-category multiplier (full specification in Section 5.1). The direct construction carbon emission factor for labor crews is set to zero (auxiliary energy consumption for workers' living can be amortized into the project's fixed carbon emission baseline and does not participate in dynamic scheduling decisions).

Activity durations are interval numbers. The actual resource occupation duration is uncertain. This leads to the total carbon emissions during the construction phase being an interval number (causal chain: $\widetilde{d}_j$ uncertain $\Rightarrow$ resource occupation duration uncertain $\Rightarrow$ carbon emissions uncertain). The interval expression for total carbon emissions is detailed in Equation (11) in Section 2.3, denoted here as $\widetilde{CE}$. The project has a total carbon emission quota Q (unit: $kgCO_2e$) according to the three-star green building certification requirements. Since $\widetilde{CE}$ is an interval number and Q is a scalar, they must be compared through a possibility-based constraint: the confidence level $\lambda \in [0.5, 1]$ quantifies the degree of quota satisfaction, where $\lambda = 1$ corresponds to the most conservative constraint (i.e., pessimistic carbon emissions do not exceed the quota), $\lambda = 0.5$ corresponds to the expected value criterion, and the interval $[0.5, 1]$ covers the complete decision spectrum from the expected criterion to the worst-case criterion (the formal definition of possibility degree is given in Section 2.2).

We simultaneously consider two competing interval objectives: minimizing the interval project makespan $f_1 = \widetilde{C}_{\max}$ and the interval total carbon emissions $f_2 = \widetilde{CE}$, forming a bi-objective interval RCPSP. Its mathematical essence is an interval bi-objective mixed-integer program. The carbon emission quota Q plays a dual role in this problem: as a hard constraint (when $\lambda = 1$), it is the qualification threshold for three-star green building certification, and exceeding it results in rejection; minimizing f_2 is lean optimization under the premise of meeting certification qualifications. The two have clearly distinguished semantics without redundancy. The complete mathematical model is established step by step in Section 2.3, and the interval Pareto dominance relation is given in Section 2.4; the main symbols are summarized in Table 1.

Table 1. Main symbol notation.

Symbol	Meaning
n	Total number of construction activities (excluding virtual activities)
$G(N, A)$	AON network; N is the activity set, A is the precedence arc set
$\mathcal{T} = \{1, \ldots, T\}$	Scheduling horizon, consisting of T equal-length periods
K	Set of renewable resource types ($\lvert K \rvert = K$)
$\widetilde{d}_j = \left[d_j^-, d_j^+\right]$	Interval duration of activity j (optimistic/pessimistic values)

Table 1. *Cont.*

Symbol	Meaning
R_{kt}	Available capacity of resource k in period t (time-varying)
r_{jk}	Fixed demand of activity j for resource k per period
EF_{kt}	Carbon emission factor of resource k in period t ($kgCO_2e$/(unit·period))
Q	Total carbon emission quota for the project ($kgCO_2e$, three-star green building requirement)
$\lambda \in [0.5,\ 1]$	Minimum confidence level for carbon quota possibility constraint
$x_{jt} \in \{0,1\}$	Binary decision variable indicating whether activity j starts in period t
S_j	Deterministic start period of activity j
$\widetilde{C}_j = \left[C_j^-,\ C_j^+\right]$	Interval completion time of activity j
$\phi_{jt}^-,\ \phi_{jt}^+$	Optimistic/pessimistic carbon emission parameters when activity j starts in period t (precomputed constants)
$f_1 = \widetilde{C}_{\max}$	Interval project makespan (objective one)
$f_2 = \widetilde{CE}$	Interval total carbon emissions (objective two)
$P\{\cdot\}$	Interval possibility degree measure function (see Definition 2 in Section 2.2)

2.2. Interval Number Theory

Definition 1 (Interval Number)**.** *Let $a^-, a^+ \in \mathrm{R}$ and $a^- \le a^+$, then $\widetilde{a} = [a^-, a^+]$ is called a closed interval number. When $a^- = a^+$, it degenerates to a real number. For any interval number $\widetilde{a}$, define its midpoint and width respectively as follows:*

$$m(\widetilde{a}) = \frac{a^- + a^+}{2}, \qquad w(\widetilde{a}) = a^+ - a^- \tag{2}$$

which characterize the location and uncertainty degree of the interval number, respectively.

Interval arithmetic operations: Let $\widetilde{a} = [a^-, a^+]$, $\widetilde{b} = [b^-, b^+]$, $\alpha \ge 0$, define:

$$\widetilde{a} + \widetilde{b} = \left[a^- + b^-,\ a^+ + b^+\right] \tag{3}$$

$$\alpha\widetilde{a} = \left[\alpha a^-,\ \alpha a^+\right] \tag{4}$$

$$\max\left(\widetilde{a},\ \widetilde{b}\right) = \left[\max\left(a^-, b^-\right),\ \max\left(a^+, b^+\right)\right] \tag{5}$$

The above operations maintain degeneration consistency with real number arithmetic, and the results remain valid interval numbers.

Definition 2 (Possibility Degree)**.** *For interval numbers $\widetilde{a}$ and $\widetilde{b}$, define the possibility degree that $\widetilde{a}$ does not exceed $\widetilde{b}$ as follows:*

$$P\{\widetilde{a} \le \widetilde{b}\} = \max\left(0,\ \min\left(1,\ \frac{1}{2} + \frac{m\left(\widetilde{b}\right) - m(\widetilde{a})}{w(\widetilde{a}) + w\left(\widetilde{b}\right)}\right)\right) \tag{6}$$

when $w(\widetilde{a}) + w\left(\widetilde{b}\right) = 0$ (both degenerate to real numbers), stipulate $P\left\{\widetilde{a} \le \widetilde{b}\right\} = 1$ if $a^- \le b^-$; otherwise 0.

Proposition 1 (Basic Properties of Possibility Degree)**.**

(1) Boundedness: $P\left\{\widetilde{a} \le \widetilde{b}\right\} \in [0,1]$;

(2) *Complementarity:* $P\left\{\widetilde{a} \le \widetilde{b}\right\} + P\left\{\widetilde{b} \le \widetilde{a}\right\} = 1$;

(3) *Degeneration consistency: when* $\widetilde{a}$ *and* $\widetilde{b}$ *both degenerate to real numbers, Equation (6) reduces to standard real number comparison;*

(4) *Neutral criterion:* $P\left\{\widetilde{a} \le \widetilde{b}\right\} \ge 0.5 \Leftrightarrow m(\widetilde{a}) \le m\left(\widetilde{b}\right)$.

Property (2) ensures the consistency of possibility degree judgments and avoids contradictory rankings; property (4) indicates that $\lambda = 0.5$ corresponds to the expected value decision criterion based on midpoints, while $\lambda = 1$ corresponds to the conservative criterion under the worst-case scenario, which is fully consistent with the engineering semantics of $\lambda \in [0.5, 1]$ in Section 2.1. It should be noted that the possibility degree in this paper is exclusively used to support the linearization derivation of constraint (C4) (see Equation (16) in Section 2.3), while the dominance relation between interval objectives is independently defined by the upper and lower bound comparison criterion in Section 2.4, and the two are not mixed.

2.3. Mathematical Model

2.3.1. Decision Variables and Derived Variables

The core of scheduling decisions is to determine the start period of each activity. From Table 1, $x_{jt} \in \{0,1\}$ is a binary decision variable, and the deterministic start period of activity j is as follows:

$$S_j = \sum_{t \in \mathcal{T}} t \cdot x_{jt}, \quad \forall j \in N \tag{7}$$

S_j is a linear derived quantity of x_{jt}, uniquely determined by the decision. Equation (7) is a definitional equality rather than an independent constraint; for brevity, subsequent constraints use S_j to replace $\sum_{t \in \mathcal{T}} t \cdot x_{jt}$, and the two are completely equivalent. Uncertainty only originates from the interval duration $\widetilde{d}_j$. We stipulate that an activity starts at the beginning of period S_j, occupies periods $S_j, S_j + 1, \ldots, S_j + d_j - 1$, and completes at the end of period $S_j + d_j - 1$, i.e., the completion time instant is $S_j + d_j$. From Equation (3), the interval completion time of activity j is as follows:

$$\widetilde{C}_j = \left[S_j + d_j^-,\ S_j + d_j^+\right], \quad \forall j \in N \tag{8}$$

2.3.2. Objective Functions

Objective one: Minimize interval project makespan

From Equation (5), taking the interval maximum over all activity interval completion times:

$$f_1 = \widetilde{C}_{\max} = \left[\max_{j \in N}\left(S_j + d_j^-\right),\ \max_{j \in N}\left(S_j + d_j^+\right)\right] \tag{9}$$

Objective two: Minimize interval total carbon emissions

Activity j, starting in period t, continuously occupies resource k (with demand r_{jk}) during consecutive periods $\tau = t, t+1, \ldots, t + d_j - 1$, generating carbon emissions. Since $d_j \in \left[d_j^-, d_j^+\right]$, precompute the carbon emission parameters for activity j starting in period t under optimistic and pessimistic durations respectively:

$$\phi_{jt}^- = \sum_{k \in K} r_{jk} \sum_{\tau=t}^{t+d_j^- -1} EF_{k\tau}, \qquad \phi_{jt}^+ = \sum_{k \in K} r_{jk} \sum_{\tau=t}^{t+d_j^+ -1} EF_{k\tau}, \quad \forall j \in N,\ t \in \mathcal{T} \tag{10}$$

Both ϕ_{jt}^- and ϕ_{jt}^+ are precomputed constant matrices depending on input data, independent of decision variables. By introducing the precomputed constant matrices ϕ_{jt}^- and

ϕ_{jt}^{+}, we linearize the originally nonlinear time-dependent carbon emission calculation into a first-order linear expression with respect to x_{jt}, significantly reducing the model's solution complexity while retaining complete information on time-varying carbon factors. From the interval addition in Equation (3), the interval total carbon emissions are as follows:

$$f_2 = \widetilde{CE} = \left[\sum_{j \in N} \sum_{t \in \mathcal{T}} \phi_{jt}^{-} \cdot x_{jt},\ \sum_{j \in N} \sum_{t \in \mathcal{T}} \phi_{jt}^{+} \cdot x_{jt} \right] \tag{11}$$

Both objective functions are linear interval functions with respect to x_{jt}, and their optimization meaning is defined by the interval Pareto dominance relation in Section 2.4.

2.3.3. Constraints

(C1) Scheduling completeness constraints: Each activity starts exactly once, and the start period is limited to the feasible window $[\mathrm{ES}_j,\ \mathrm{LS}_j]$ calculated by the critical path method (based on optimistic duration d_j^{-}):

$$\sum_{t \in \mathcal{T}} x_{jt} = 1, \quad \forall j \in N \tag{12}$$

$$xjt = 0, \quad \forall\, t \notin \mathrm{ES}j, \mathrm{LS}j, \quad \forall\, j \in N \tag{13}$$

(C2) Precedence constraints: If $(i, j) \in A$, to ensure that the precedence relation still holds under the worst-case scenario (activity i duration reaches pessimistic value d_i^{+}), the constraint takes a conservative form:

$$S_j \geq S_i + d_i^{+}, \quad \forall (i, j) \in A \tag{14}$$

This treatment is consistent with constraint (C3) pessimistic duration assumption for resources, ensuring that the generated baseline schedule has full robustness over the entire realization range of interval durations.

(C3) Renewable resource constraints: We use pessimistic duration d_j^{+} to construct resource constraints to ensure that resources are not overloaded under the worst-case scenario. Activity j is in execution state during period τ if and only if it starts in some period within $\left[\tau - d_j^{+} + 1,\ \tau\right]$, thus:

$$\sum_{j \in N} \sum_{t=\max(1,\ \tau - d_j^{+} + 1)}^{\tau} r_{jk} \cdot x_{jt} \leq R_{k\tau}, \quad \forall k \in K,\ \tau \in \mathcal{T} \tag{15}$$

The time-varying nature of $R_{k\tau}$ uniformly covers situations such as nighttime construction bans (setting $R_{k\tau} = 0$) and rainy season load reduction, without requiring additional handling.

(C4) Carbon emission quota possibility constraint: Require that the interval carbon emissions $\widetilde{CE}$ satisfy the quota Q with confidence level λ, i.e., $P\left\{\widetilde{CE} \leq Q\right\} \geq \lambda$. Treating Q as a degenerate interval $[Q, Q]$ ($m([Q, Q]) = Q$, $w([Q, Q]) = 0$), substituting into Equation (6) and expanding, this simplifies to the equivalent linear constraint:

$$\sum_{j \in N} \sum_{t \in \mathcal{T}} \left[\lambda \cdot \phi_{jt}^{+} + (1 - \lambda) \cdot \phi_{jt}^{-}\right] x_{jt} \leq Q \tag{16}$$

Equation (16) has intuitive engineering semantics: when $\lambda = 1$, it degenerates to pessimistic carbon emissions not exceeding the quota ($\sum \phi_{jt}^{+} x_{jt} \leq Q$), which is the most conservative constraint; when $\lambda = 0.5$, it degenerates to expected carbon emissions not

exceeding the quota. The continuous adjustment of λ allows decision makers to trade off between conservatism and feasible region looseness without modifying the model structure; the dual role of carbon quota Q and objective function f_2 has been clearly distinguished in Section 2.1.

(C5) Variable domain constraints:

$$x_{jt} \in \{0,1\}, \quad \forall j \in N,\ t \in \mathcal{T} \tag{17}$$

2.3.4. Complete Model Summary

Integrating Equations (9)–(17), the bi-objective interval RCPSP model in this paper is as follows:

$$\min\ (f_1,\ f_2) = \min\left(\widetilde{C}_{\max},\ \widetilde{CE}\right)$$

s.t. (C1) Equations (12) and (13): Scheduling completeness constraints
(C2) Equation (14): Precedence constraints (pessimistic robust)
(C3) Equation (15): Time-varying resource capacity constraints (pessimistic robust)
(C4) Equation (16): Carbon emission quota possibility constraint
(C5) Equation (17): Variable domain constraints

Model (P) contains $O(n \cdot T)$ binary decision variables, with constraint scale $O(|A| + K \cdot T)$. It can be proven to belong to the NP-hard class by reduction from the standard RCPSP NP-hardness, and the exact solution is only suitable for small-scale instances.

2.4. Interval Pareto Dominance

Definition 3 (Interval Pareto Dominance). *Let $\widetilde{f}_i(\mathrm{u}) = \left[f_i^-(\mathrm{u}),\ f_i^+(\mathrm{u})\right]$ be the interval objective value of solution u on objective i. For two feasible solutions u and v of model (P), u Pareto-dominates v in the interval sense (denoted $u \prec v$) if and only if for all objectives $i \in \{1,2\}$:*

$$f_i^-(\mathrm{u}) \le f_i^-(\mathrm{v}) \quad \textit{and} \quad f_i^+(\mathrm{u}) \le f_i^+(\mathrm{v}) \tag{18}$$

and at least one objective i exists such that one of the above inequalities holds strictly (<).

This definition simultaneously constrains the optimistic case (lower bound) and pessimistic case (upper bound), naturally favoring solutions with smaller interval widths, i.e., better objective values with lower uncertainty. Compared to weak dominance criteria that only compare midpoints, this definition completely preserves the risk information carried by interval widths, consistent with the original intention of introducing interval durations in this paper.

Definition 4 (Interval Pareto Optimal Solution Set and Front). *Let Ω be the set of all feasible solutions, define:*

$$\mathcal{P}^* = \{\mathrm{u} \in \Omega \mid \nexists\, \mathrm{v} \in \Omega :\ \mathrm{v} \prec \mathrm{u}\} \tag{19}$$

as the interval Pareto optimal solution set, and its corresponding objective vector set $\mathcal{F}^* = \{(f_1(\mathrm{u}), f_2(\mathrm{u})) \mid \mathrm{u} \in \mathcal{P}^*\}$ is called the interval Pareto front. The goal of the algorithm in this paper is to approximate $\mathcal{F}^*$.

2.5. Constraint Handling

During the optimization process of evolutionary algorithms, candidate solutions may violate constraint (C4). To establish a unified comparison criterion between feasible and infeasible solutions, define the constraint violation degree (VD) of solution x as follows:

$$\mathrm{VD}(\mathbf{x}) = \max\left(0,\ \frac{1}{Q}\sum_{j\in N}\sum_{t\in\mathcal{T}}\left[\lambda\ \phi_{jt}^{+} + (1-\lambda)\ \phi_{jt}^{-}\right]x_{jt} - 1\right) \tag{20}$$

$\mathrm{VD}(\mathrm{x}) = 0$ if and only if x satisfies constraint (C4), and $VD(x) > 0$ reflects the relative degree of carbon exceeding the quota.

Based on this, solution comparison follows the feasibility-first rule: If $VD(u) = 0$ and $VD(v) > 0$, then u is superior to v; if $VD(u),\ VD(v) > 0$, then the one with smaller VD is preferred; if $VD(u) = VD(v) = 0$, compare according to the interval Pareto dominance relation in Definition 3.

Since model (P) adds interval uncertainty, time-varying emissions, and dynamic quotas to the already NP-hard RCPSP, exact methods are viable only for tiny instances. Section 3 therefore develops the KD-IMOEA metaheuristic, which directly embeds the feasibility-first rule established above into its non-dominated sorting mechanism.

3. Algorithm Design

3.1. Overall Framework

Since model (P) is NP-hard (Section 2), exact methods are computationally infeasible for medium- and large-scale instances. We propose KD-IMOEA, building on the NSGA-II framework. Targeting three structural features of model (P), interval duration uncertainty, time-varying emission factors, and possibility-based carbon budget constraints, the algorithm embeds domain knowledge into four evolutionary stages: initialization, decoding, mutation, and archive management. These map to four components: Carbon-aware Knowledge-Driven Initialization (KDI, Section 3.2), Adaptive Time-window Carbon-aware Decoding (TCD, Section 3.3), Carbon Budget-aware Repair Mutation (CBM, Section 3.4), and Interval Pareto Elite Archive (IPA, Section 3.5). The algorithm applies the interval Pareto dominance from Definition 3 for solution comparison and the feasibility-first rule from Section 2.5 for constraint handling, outputting an approximate interval Pareto front $\mathcal{F}^{*}$.

The overall flow is shown in Figure 1, with complete pseudocode in Algorithm 1. Execution has three phases: precomputation, initialization, and main loop. During precomputation, emission parameter matrices ϕ_{jt}^{-}- and ϕ_{jt}^{+} are built once via Equation (10) and reused by all subsequent decoding and constraint evaluation. During initialization, KDI generates the initial population using three domain-specific rules, TCD decodes individuals, feasibility-first non-dominated sorting is performed, and the IPA archive A is initialized empty. In the main loop, each generation runs binary tournament selection, SBX crossover, and TCD decoding. When an offspring violates the carbon budget ($\mathrm{VD}(\mathrm{x}^{c}) > 0$), CBM repair is triggered. Parent and offspring populations are merged, sorted by feasibility-first non-dominated sorting, and truncated to NP individuals; the IPA archive is then updated. The loop terminates at gen > MaxGen, outputting the non-dominated set F* from the archive. Overall time complexity is $O\left(MaxGen\cdot\left(NP^{2} + NP\cdot n\cdot K\right)\right)$, the same order as standard NSGA-II.

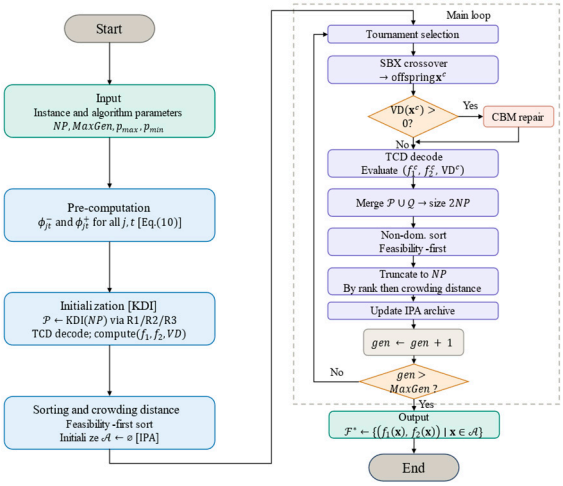


Figure 1. Flowchart of KD-IMOEA.

Algorithm 1 KD-IMOEA

Input: Project instance $\left\{G(N, A),\ R_{kt},\ r_{jk},\ EF_{kt},\ Q,\ \lambda\right\}$; algorithm parameters NP, $MaxGen$, p_{max}, p_{min}

Output: Approximate interval Pareto front $\mathcal{F}^*$

//Pre-computation

1: for all $j \in N, t \in \mathcal{T}$ do compute ϕ_{jt}^- and ϕ_{jt}^+ via Equation (10) end for

//Initialization

2: $\mathcal{P} \leftarrow \mathrm{KDI}(NP)$ ▷ *Algorithm 2, Section 3.2*

3: for $i = 1$ to NP do

4: $\left(f_1^i,\ f_2^i,\ \mathrm{VD}^i\right) \leftarrow \text{TCD-Decode}\left(\mathrm{x}^i\right)$ ▷ *Algorithm 3, Section 3.3*

5: end for

6: Perform feasibility-first non-dominated sorting on $\mathcal{P}$; compute crowding distances

7: $\mathcal{A} \leftarrow \varnothing$; $\mathcal{A} \leftarrow \text{IPA-Update}(\mathcal{A},\ \text{non-dominated}\,(\mathcal{P}))$ ▷ *Algorithm 5, Section 3.5*

//Main loop

8: for $gen = 1$ to $MaxGen$ do

9: $\mathcal{Q} \leftarrow \varnothing$

10: for $i = 1$ to NP do

11: Select parents x^a, x^b via binary tournament selection

12: $\mathrm{x}^c \leftarrow \text{SBX-Crossover}\left(\mathrm{x}^a,\ \mathrm{x}^b\right)$

13: if $\mathrm{VD}(\mathrm{x}^c) > 0$ then

14: $\mathrm{x}^c \leftarrow \text{CBM-Repair}(\mathrm{x}^c)$ ▷ *Algorithm 4, Section 3.4*

15: end if

16: $\left(f_1^c,\ f_2^c,\ \mathrm{VD}^c\right) \leftarrow \text{TCD-Decode}(\mathrm{x}^c)$

17: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{\mathrm{x}^c\}$

18: end for

19: $\mathcal{R} \leftarrow \mathcal{P} \cup \mathcal{Q}$ ▷ *Combined population, size* $2NP$

20: Perform feasibility-first non-dominated sorting on $\mathcal{R}$; compute crowding distances

21: $\mathcal{P} \leftarrow$ Select top NP individuals from $\mathcal{R}$ by rank, then crowding distance

22: $\mathcal{A} \leftarrow \text{IPA-Update}(\mathcal{A},\ \text{non-dominated}\,(\mathcal{P}))$ ▷ *Algorithm 5, Section 3.5*

23: end for

//Output

24: return $\mathcal{F}^* \leftarrow \{(f_1(\mathrm{x}),\ f_2(\mathrm{x})) \mid \mathrm{x} \in \mathcal{A}\}$

3.2. Carbon-Aware Knowledge-Driven Initialization (KDI)

Population initialization quality directly affects convergence speed and solution diversity. Embedding domain knowledge into initial population generation using a multi-rule hybrid strategy can markedly improve initial quality and accelerate convergence [29]. For model (P), purely random initialization is blind to the carbon budget constraint (C4), producing many infeasible individuals and increasing the computational burden of subsequent repair and evolution. KDI embeds the precomputed emission matrices ϕ_{jt}^{-} and ϕ_{jt}^{+} along with resource utilization data into population generation. Based on the Serial Schedule Generation Scheme (SSGS) [30], KDI produces an initial population $\mathcal{P}_0$ of size NP by mixing three construction rules R1, R2, and R3 at ratios 40%, 30%, and 30%. The full procedure is in Algorithm 2.

Algorithm 2 Carbon-Aware Knowledge-Driven Initialization (KDI)

Input: Project instance $\left\{G(N, A),\ R_{kt},\ r_{jk},\ EF_{kt}\right\}$; population size NP; precomputed ϕ_{jt}^{-}, ϕ_{jt}^{+}
Output: Initial population $\mathcal{P}_0$
//R1: emission-guided construction (40% of NP)
for $i = 1$ to $\lfloor 0.4 \cdot NP \rfloor$ do
    Rank activities by carbon intensity $CI_j = \sum_k r_{jk} \cdot \overline{EF}_k$ in descending order
    for each activity j in ranked order do
        $t^* \leftarrow \text{argmin}_{t \in [ES_j,\ LS_j]} \sum_k r_{jk} \cdot EF_{kt}$ subject to precedence and resource feasibility
        Assign $x_{jt^*} = 1$
    end for
    $\mathrm{x}^i \leftarrow$ resulting schedule; $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \left\{\mathrm{x}^i\right\}$
end for
//R2: resource-guided construction (30% of NP)
for $i = 1$ to $\lfloor 0.3 \cdot NP \rfloor$ do
    Rank activities by resource demand $RD_j = \sum_k r_{jk}$ in descending order
    for each activity j in ranked order do
        $t^* \leftarrow \text{argmin}_{t \in [ES_j,\ LS_j]} \sum_k R_{kt}^{\text{used}}$ subject to precedence and resource feasibility
        Assign $x_{jt^*} = 1$
    end for
    $\mathrm{x}^i \leftarrow$ resulting schedule; $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \left\{\mathrm{x}^i\right\}$
end for
//R3: random construction (remaining 30% of NP)
for $i = 1$ to $NP - |\mathcal{P}_0|$ do
    $\mathrm{x}^i \leftarrow$ random topological ordering satisfying constraints (C1)–(C2)
    $\mathcal{P}_0 \leftarrow \mathcal{P}_0 \cup \left\{\mathrm{x}^i\right\}$
end for
return $\mathcal{P}_0$

Rule R1 (40%): Carbon-oriented construction. For each activity j, compute its carbon intensity $CI_j = \sum_k r_{jk} \cdot \overline{EF}_k$, where $\overline{EF}_k$ is the time-averaged emission factor for resource k. Sort activities by descending CI_j, and assign each to the lowest-emission time $t^* = \arg\min_{t \in [ES_j,\ LS_j]} \sum_k r_{jk} \cdot EF_{kt}$ satisfying precedence and resource feasibility. R1 uses the precomputed matrix ϕ_{jt}^{-} to steer carbon-intensive activities toward low-emission periods, raising the initial population's feasibility rate for constraint (C4).

Rule R2 (30%): Resource-oriented construction. Sort activities by descending total resource demand $RD_j = \sum_k r_{jk}$, and assign each to the feasible period with lowest current resource utilization $\sum_k R_{kt}^{\text{used}}$, alleviating scheduling congestion while satisfying (C3). R2 complements R1: R1 targets carbon, R2 targets resource balance, and together they improve feasibility.

Rule R3 (30%): Random construction. Randomly generate topological sequences satisfying (C1)–(C2), injecting diversity to prevent the greedy strategies of R1 and R2 from concentrating the initial population in local regions.

All individuals from R1-R3 are decoded by TCD (Section 3.3), mapped to interval schedules, and evaluated for $\left(f_1^i,\ f_2^i,\ \text{VD}^i\right)$, then passed to feasibility-first non-dominated sorting in line 6 of Algorithm 1.

3.3. Adaptive Time-Window Carbon-Aware Decoding (TCD)

The decoder maps chromosomes (priority lists) to feasible interval schedules and computes objective values and constraint violation degrees. It is the bridge between population evolution and schedule evaluation. TCD is called in line 4 of Algorithm 1 (initialization) and line 16 (offspring evaluation); its output triplet $\left(f_1^i,\ f_2^i,\ \text{VD}^i\right)$ implements Equations (9), (11) and (20).

Chromosome representation and basic decoding. We use an activity list (priority list) as chromosome encoding, where activities satisfy topological feasibility (all predecessors precede each activity). Basic decoding follows SSGS [31]: activities are processed in priority-list order and each is scheduled at the earliest feasible start period satisfying (C1)–(C3). This guarantees resource and precedence feasibility.

Carbon-aware extension: Pure SSGS ignores $EF_{k\tau}$ differences between periods, often producing high-carbon solutions under time-varying factors. TCD adds carbon-aware decision-making within the feasible window $[ES_j, LS_j]$ of basic SSGS: with adaptive probability $p_{\text{carbon}}(\text{gen})$, the activity is placed in the period $\tau \in [ES_j, LS_j]$ that minimizes the resource-weighted emission load $\sum_k r_{jk} \cdot EF_{k\tau}$; with probability $1 - p_{\text{carbon}}(\text{gen})$, the earliest-start strategy is retained, where

$$p_{carbon}(gen) = p_{max} - (p_{max} - p_{min}) \cdot \frac{gen}{MaxGen} \tag{21}$$

This probability starts high ($p_{max} = 0.9$), to guide the population toward low-carbon feasible regions, then decays to $p_{min} = 0.1$, preserving makespan optimization space and preventing excessive carbon bias that would reduce diversity [32]. At $gen = 0$ (initialization stage), $p_{carbon} = p_{max}$, consistent with the KDI carbon-aware orientation. The decoding data flow is shown in Figure 2: the priority list is expanded by SSGS into time-window heat bars, probability $p_{carbon}(gen)$ drives a two-branch decision, the Yes branch selects lowest-EF period t^*, the No branch retains ES_j, and both converge through $x_{jt^*} = 1$ to complete schedule assignment and objective evaluation.

Objective and constraint evaluation. After activity placement, the precomputed matrices ϕ_{jt}^- and ϕ_{jt}^+ (line 1 of Algorithm 1, Equation (10)) linearly compute interval total emissions; Equation (9) gives interval makespan; the objective vector (f_1, f_2) and VD from Equation (20) are obtained. When VD > 0, CBM repair is triggered (line 13 of Algorithm 1). When VD = 0, the individual enters non-dominated sorting directly. The complete process is in Algorithm 3.

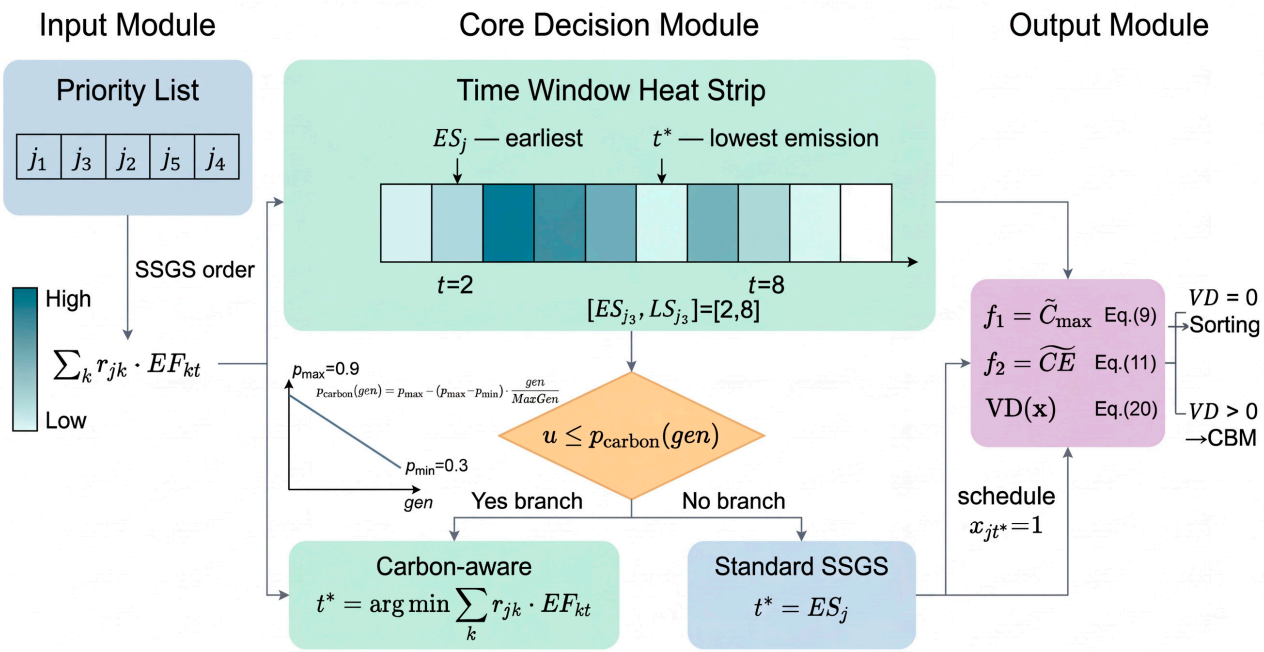


Figure 2. TCD decoding mechanism: adaptive carbon-aware time-period selection.

Algorithm 3 Adaptive Time-window Carbon-aware Decoding (TCD)

Input: Chromosome (priority list) x; generation counter gen; parameters p_{max}, p_{min}, $MaxGen$; precomputed ϕ_{jt}^-, ϕ_{jt}^+; instance $\{G(N, A),\ R_{kt},\ r_{jk},\ EF_{kt},\ Q,\ \lambda\}$

Output: Interval objective tuple $(f_1,\ f_2,\ \mathrm{VD})$

//Adaptive carbon-aware probability

$p_{carbon} \leftarrow p_{max} - (p_{max} - p_{min}) \cdot gen / MaxGen$

//Serial schedule generation (SSGS) with carbon-aware time-period selection

Initialize resource usage $R_{kt}^{used} \leftarrow 0$ for all $k \in K, t \in \mathcal{T}$

for each activity j in priority-list order do

  Determine feasible window $[ES_j,\ LS_j]$ based on constraints (C1)–(C2) and current R_{kt}^{used}

  Draw $u \sim \mathrm{Uniform}(0,\ 1)$

  if $u \le p_{carbon}$ then

    $t^* \leftarrow \mathrm{argmin}_{t \in [ES_j,\ LS_j]} \sum_k r_{jk} \cdot EF_{kt}$ ▷ *Carbon-aware: lowest-EF period*

  else

    $t^* \leftarrow ES_j$ ▷ *Standard SSGS: earliest feasible start*

  end if

  Set $x_{jt^*} = 1$; update R_{kt}^{used} for $t \in [t^*,\ t^* + d_j^+ - 1]$

end for

//Evaluate interval objectives and constraint violation

$f_1 \leftarrow \left[\max_{j \in N}\left(S_j + d_j^-\right),\ \max_{j \in N}\left(S_j + d_j^+\right)\right]$ ▷ *Equation (9)*

$f_2 \leftarrow \left[\sum_j \sum_t \phi_{jt}^- x_{jt},\ \sum_j \sum_t \phi_{jt}^+ x_{jt}\right]$ ▷ *Equation (11)*

$\mathrm{VD} \leftarrow \max\left(0,\ \frac{1}{Q}\sum_j \sum_t \left[\lambda\ \phi_{jt}^+ + (1 - \lambda)\ \phi_{jt}^-\right] x_{jt} - 1\right)$ ▷ *Equation (20)*

return $(f_1,\ f_2,\ \mathrm{VD})$

3.4. Carbon Budget-Aware Repair Mutation (CBM)

SBX crossover is blind to the carbon budget constraint (C4), and offspring x^c frequently violate it $VD(x^c) > 0$. Penalty functions require manual coefficient tuning with high sensitivity. Discarding infeasible solutions wastes crossover computation. CBM, triggered conditionally in lines 13–14 of Algorithm 1, combines repair and mutation: it restores feasibility while directly cutting emissions, achieving both constraint satisfaction and objective improvement in synergy with the feasibility-first rule (Section 2.5).

Step 1: Identify high-carbon activities. For infeasible offspring x^c, compute the carbon contribution of each activity j using the weighted parameters from Equation (16):

$$CC_j = \sum_{t \in \mathcal{T}} \left[\lambda \, \phi_{jt}^{+} + (1 - \lambda) \, \phi_{jt}^{-} \right] x_{jt}^{c}, \forall j \in N \tag{22}$$

Sort by descending CC_j and select the top 30% as the high-carbon set $\mathcal{H}$. This step reuses the precomputed matrix $\phi_{jt}^{\pm}$ (line 1 of Algorithm 1, Equation (10)) with no additional overhead.

Step 2: Lowest-EF period reallocation. For each activity j in $\mathcal{H}$, search within $[ES_j, \ LS_j]$ for periods satisfying (C1)–(C3), select t^* minimizing $\sum_k r_{jk} \cdot EF_{kt}$, and reassign. This strategy mirrors KDI rule R1 (Section 3.2) and the TCD carbon-aware branch (Section 3.3), reflecting internal consistency across algorithm components. Figure 3 illustrates the repair: high-carbon activities j_1, j_3, j_5 initially in high-emission periods (orange, ↑) are moved to lowest-EF periods within their windows (green, ↓), recovering from VD > 0 to VD = 0.

Step 3: Update. After reallocation, update the resource matrix R_{kt}^{used}. The repaired offspring returns to line 16 of Algorithm 1 for TCD re-evaluation. If $VD^c > 0$ persists after repair, the feasibility-first rule automatically downgrades it in sorting without re-triggering CBM, preventing infinite loops. The complete procedure is in Algorithm 4.

Algorithm 4 Carbon Budget-aware Repair Mutation (CBM)

Input: Infeasible offspring x^c with $VD(x^c) > 0$; precomputed ϕ_{jt}^{-}, ϕ_{jt}^{+}; parameters λ, Q; instance $\left\{ G(N, A), \ R_{kt}, \ r_{jk}, \ EF_{kt} \right\}$

Output: Repaired offspring x^c

//Step 1: Identify high-carbon activities

1: for all $j \in N$ do

2: $\quad CC_j \leftarrow \sum_{t \in \mathcal{T}} \left[\lambda \, \phi_{jt}^{+} + (1 - \lambda) \, \phi_{jt}^{-} \right] x_{jt}^{c}$ ▷ *Equation (22)*

3: end for

4: $\mathcal{H} \leftarrow$ top $\lfloor 0.3 \cdot n \rfloor$ activities ranked by CC_j descending

//Step 2: Reassign to lowest-emission feasible periods

5: for each $j \in \mathcal{H}$ do

6: $\quad t^* \leftarrow \text{argmin}_{t \in [ES_j, \ LS_j]} \sum_k r_{jk} \cdot EF_{kt}$ subject to (C1)–(C3)

7: $\quad$ Set $x_{jt^*}^{c} = 1$; clear $x_{jt\prime}^{c} = 0$ for all $t\prime \neq t^*$

8: $\quad$ Update R_{kt}^{used} for $t \in \left[t^*, \ t^* + d_j^{+} - 1 \right]$

9: end for

//Step 3: Return repaired offspring

10: return x^c ▷ *Re-evaluated by TCD, Algorithm 1 Line 16*

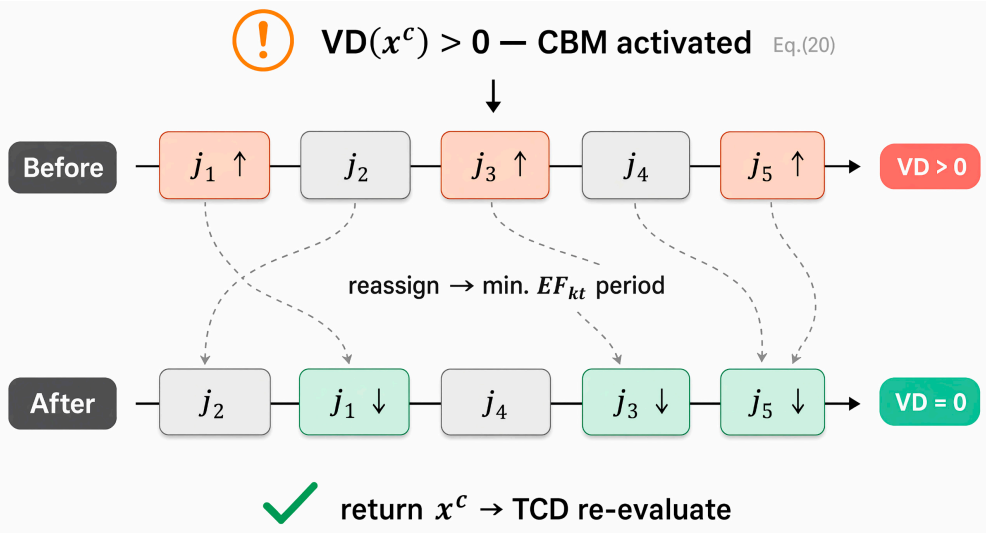


Figure 3. CBM repair mechanism.

3.5. Interval Pareto Elite Archive (IPA)

Standard NSGA-II elite preservation is tied to within-population non-dominated sorting, and each generation's truncation inevitably discards some historical high-quality solutions; therefore, the final output reflects only the last generation's local optimum rather than the global optimum across all generations. IPA addresses this as an external archive independent of the evolving population, maintaining the non-dominated set $\mathcal{A}$ discovered throughout execution and ensuring the output front $\mathcal{F}^*$ covers the entire search history. IPA is initialized as $\mathcal{A} \leftarrow \emptyset$ in line 7 of Algorithm 1 and updated in line 22 of each generation. The complete procedure is in Algorithm 5.

Algorithm 5 Interval Pareto Archive Update (IPA-Update)

Input: Archive $\mathcal{A}$; candidate set $\mathcal{S}$ (non-dominated individuals from current population); population $\mathcal{P}$; archive capacity $M = NP$
Output: Updated archive $\mathcal{A}$
//Step 1: Filter and update archive via interval Pareto dominance
1: for each x $\in \mathcal{S}$ do
2: if $\exists$ y $\in \mathcal{A}$ such that y $\prec$ x then
3: Discard x $\triangleright$ x *is dominated by archive member, reject*
4: else
5: Remove all y $\in \mathcal{A}$ with x $\prec$ y $\triangleright$ x *dominates, purge inferior members*
6: $\mathcal{A} \leftarrow \mathcal{A} \cup \{x\}$
7: end if
8: end for
//Step 2: Capacity control via crowding distance pruning
9: if $|\mathcal{A}| > M$ then
10: Compute crowding distances for all x $\in \mathcal{A}$
11: Remove individuals with the smallest crowding distance until $|\mathcal{A}| = M$
12: end if
//Step 3: Elite injection into population
13: $\mathcal{E} \leftarrow$ randomly sample $\lfloor 0.05 \cdot NP \rfloor$ individuals from $\mathcal{A}$
14: Replace $\lfloor 0.05 \cdot NP \rfloor$ individuals with the smallest crowding distance in $\mathcal{P}$ with $\mathcal{E}$
15: return $\mathcal{A}$

Filtering. Each candidate x from $\mathcal{S}$ (current population's non-dominated individuals) undergoes bidirectional filtering via Definition 3: if any $y \in \mathcal{A}$ dominates $x(y \prec x)$, x is discarded; otherwise, all solutions in $\mathcal{A}$ dominated by x are removed and x is added. This strictly follows the upper/lower bound comparison, preserving interval width information.

Capacity control. Archive capacity is $M = NP$. When $| \mathcal{A} | > M$, individuals with the smallest crowding distance are deleted until $| \mathcal{A} | = M$, maintaining distribution uniformity in objective space.

Elite injection. After each update, $\lfloor 0.05 \cdot NP \rfloor$ elites randomly selected from $\mathcal{A}$ replace equal individuals with the smallest crowding distance in $\mathcal{P}$, reintroducing historical optima to prevent genetic drift.

Upon termination, the objective vectors of all individuals in $\mathcal{A}$ form the output $\mathcal{F}^* = \{(f_1(\mathbf{x}),\ f_2(\mathbf{x})) \mid \mathbf{x} \in \mathcal{A}\}$ returned in line 24 of Algorithm 1, closing the KD-IMOEA flow.

3.6. Algorithm Parameter Calibration

KD-IMOEA has four parameters: population size NP, maximum generations $MaxGen$, and TCD carbon-aware probability bounds p_{max} and p_{min} (Equation (21)). We calibrate these using the Taguchi orthogonal experiment method [33] to avoid expensive one-at-a-time tuning.

Parameter levels. The parameter levels in Table 2 were selected based on problem scale and literature ranges [34], aiming to cover the typical operating range of population-based evolutionary algorithms while reflecting the computational scale of the proposed interval RCPSP. The population size levels $NP \in \{50,\ 100,\ 150\}$ represent small, medium, and relatively large populations under the same decoding cost. $MaxGen \in \{300,\ 500,\ 800\}$ balances convergence opportunity and computational burden. The upper carbon-guidance probability $p_{\max} \in \{0.7,\ 0.9,\ 1.0\}$ tests moderate to fully carbon-oriented early-stage decoding, whereas $p_{\min} \in \{0.1,\ 0.3,\ 0.5\}$ controls the retained carbon bias in later generations. Thus, the Taguchi design was intended to identify a robust accuracy-efficiency trade-off rather than simply select the single run with the highest observed HV.

Table 2. Parameter levels for Taguchi orthogonal experiment.

Parameter	Description	Level 1	Level 2	Level 3
NP	Population size	50	100	150
$MaxGen$	Maximum generations	300	500	800
p_{max}	Max carbon-aware probability	0.7	0.9	1.0
p_{min}	Min carbon-aware probability	0.1	0.3	0.5

Orthogonal design. The $L_9(3^4)$ array arranges nine independent runs covering main effects of four factors at three levels, reducing the 81 runs of full factorial. Hypervolume (HV) [35] serves as the performance metric: larger HV indicates better convergence and diversity [36]. Each combination is run 15 times on standard instances, with mean HV as the measure. The resulting L9 orthogonal experimental design and the corresponding mean HV values are summarized in Table 3.

The total time complexity is $O(\text{MaxGen} \cdot (NP^2 + NP \cdot n \cdot K))$. Per-generation overhead consists of non-dominated sorting ($O(NP^2)$) and the carbon-aware decoding/repair operations within TCD and CBM ($O(NP \cdot n \cdot K)$). When $NP \geq n \cdot K$, the component overhead is dominated by the sorting term and the asymptotic complexity matches standard NSGA-II $O(\text{MaxGen} \cdot NP^2)$; this regime applies to small-scale instances and many practical engineering scenarios. For medium-scale instances such as J30 ($n \cdot K = 120$, comparable to $NP = 100$, hence borderline) and J60 ($n \cdot K = 240$ with $NP = 100$), the component overhead begins to contribute non-negligibly. For larger instances such as J120 ($n \cdot K = 480 > NP = 100$), the

component overhead is not absorbed by the sorting term and contributes a measurable additive cost, as quantified by the runtime comparison reported in Section 4.2.

Table 3. Taguchi $L_9(3^4)$ orthogonal experiment results.

Run	*NP*	*MaxGen*	p_{max}	p_{min}	HV (Mean)
1	50	300	0.7	0.1	2665.85
2	50	500	0.9	0.3	2881.60
3	50	800	1.0	0.5	2884.65
4	100	300	0.9	0.5	2997.50
5	100	500	1.0	0.1	3203.16
6	100	800	0.7	0.3	2560.35
7	150	300	1.0	0.3	2871.20
8	150	500	0.7	0.5	2510.24
9	150	800	0.9	0.1	3180.00
Optimal	100	300	0.9	0.1	—

Note: Based on the main effect analysis, the optimal combination is $NP = 100$, $p_{max} = 0.9$, and $p_{min} = 0.1$. The main effect analysis indicates that MaxGen exhibits minimal sensitivity (mean HV: 2844.85 for 300, 2865.00 for 500, and 2875.00 for 800), with diminishing marginal returns beyond 300. Therefore, MaxGen = 300 is selected as the baseline to balance convergence quality and computational cost. For large-scale instances (J120), MaxGen is adaptively increased to 500 to ensure sufficient convergence.

4. Computational Experiments

4.1. Experimental Setup

Benchmark instances. The benchmark instances are constructed by augmenting the widely used PSPLIB J30, J60, and J120 RCPSP benchmark instance sets [37], which provide the original AON precedence networks, nominal activity durations, renewable-resource capacities, and baseline resource demands. For each scale, ten PSPLIB instances are selected, corresponding to 30, 60, and 120 non-dummy activities, respectively. The nominal PSPLIB duration is used as the optimistic duration d_j^-, and the pessimistic duration is generated as $d_j^+ = \left\lceil 1.2 d_j^- \right\rceil$, consistent with the interval width coefficient $\alpha = 1.2$. The four renewable resources are mapped to construction resources: $k = 0,\ 1$ represent mechanical resources with nonzero time-varying carbon emission factors, whereas $k = 2,\ 3$ represent labor resources with $EF_{k\tau} = 0$. The benchmark emission-factor time series follows the same day-level peak/mid/valley operating-profile structure described in Sections 2.1 and 5.1, using multipliers 1.2, 1.0, and 0.7 applied to resource-specific base emission factors. For each PSPLIB-derived instance, the carbon budget Q is dynamically calibrated by generating 100 random feasible topological schedules and setting Q to 85% of the median pessimistic carbon emission, thereby producing a tight but feasible carbon constraint under Equation (16). The selected PSPLIB instance IDs, emission-factor category seeds, and algorithm random seeds are fixed across all algorithms to ensure paired-comparison validity and reproducibility.

Baseline algorithms. KD-IMOEA is compared with four algorithms: NSGA-II [13], NSGA-III [14], MOEA/D [15], and SPEA2 [16]. All five share the same interval Pareto dominance (Definition 3), SSGS decoding, and order crossover (OX), ensuring that performance differences stem solely from KD-IMOEA four components (KDI, TCD, CBM, IPA).

Performance metrics. (1) Hypervolume (HV): volume dominated by the approximate front; larger is better. Reference point is 1.1× the maximum objective value across all algorithms per instance. (2) Inverted Generational Distance (IGD) [38]: average distance from reference front to approximate front; smaller is better. Reference front is the merged non-dominated set from all algorithms. (3) Spacing: distribution uniformity; smaller is better. For performance evaluation, HV, IGD, and Spacing are computed using the midpoint representation of interval objective values, $f_{1,\mathrm{mid}} = \left(f_1^- + f_1^+\right)/2$ and $f_{2,\mathrm{mid}} = \left(f_2^- + f_2^+\right)/2$,

to maintain comparability with standard multi-objective scheduling benchmarks. This midpoint-based metric calculation is used only for external evaluation; during evolutionary selection, KD-IMOEA still applies the interval Pareto dominance relation in Definition 3, which compares both lower and upper bounds of the interval objectives. Fronts are deduplicated before metric calculation.

Parameter. Taguchi-calibrated (Section 3.6): $NP = 100$, $p_{max} = 0.9$, $p_{min} = 0.1$. Baseline $MaxGen = 300$ (J30, J60), increased to 500 for J120. $\lambda = 0.7$, IPA elite injection ratio 5%, CBM trigger threshold VD > 0.05. All baselines use NP = 100 and matching MaxGen.

The Wilcoxon signed-rank test ($\alpha = 0.05$) evaluates pairwise differences between KD-IMOEA and each baseline algorithm. For each scale, the mean metric value over 10 independent runs on the same instance is used as one paired observation, yielding 10 paired observations per algorithm-scale-metric comparison. To quantify the magnitude of improvement, Cohen's d_z is calculated from the paired improvement differences, where positive values indicate better performance of KD-IMOEA. For HV, the paired difference is defined as KD-IMOEA minus the baseline; for IGD, it is defined as the baseline minus KD-IMOEA because smaller IGD is better. Holm-Bonferroni correction is applied across all 24 pairwise comparisons.

Environment. Python 3.10, Intel Xeon Platinum 8470Q (25 cores), 90 GB RAM. Each algorithm runs 10 times per instance with different random seeds.

4.2. Benchmark Comparison Results

Tables 4 and 5 report HV and IGD means and standard deviations for the five algorithms on J30, J60, and J120 instances (best values in bold). Figures 4 and 5 show box plots. Table 6 gives the Wilcoxon test results.

Table 4. Comparison of HV (mean ± std) across different problem scales. Bold values indicate the best performance.

Algorithm	J30	Rank	J60	Rank	J120	Rank
KD-IMOEA	458,540.30 ± 15,127.99	1	342,924.78 ± 16,030.30	1	88,757.32 ± 19,421.23	1
NSGA-II	435,314.55 ± 15,260.87	3	325,971.38 ± 16,583.47	2	83,119.61 ± 18,073.15	4
NSGA-III	435,088.95 ± 16,063.05	4	325,475.55 ± 16,841.87	3	83,319.14 ± 18,192.48	3
MOEA/D	419,097.30 ± 17,837.54	5	305,145.47 ± 20,769.08	5	64,731.12 ± 17,678.23	5
SPEA2	435,612.69 ± 16,215.32	2	325,195.21 ± 17,445.87	4	83,513.31 ± 18,620.69	2

Table 5. Comparison of IGD (mean ± std) across different problem scales. Bold values indicate the best performance.

Algorithm	J30	Rank	J60	Rank	J120	Rank
KD-IMOEA	**15.34 ± 12.29**	1	**20.72 ± 13.55**	1	**102.74 ± 72.56**	1
NSGA-II	22.42 ± 16.72	3	30.91 ± 17.50	3	147.54 ± 103.75	4
NSGA-III	22.78 ± 17.87	4	31.08 ± 18.05	4	146.86 ± 104.51	2
MOEA/D	38.99 ± 21.74	5	53.10 ± 25.23	5	200.64 ± 108.53	5
SPEA2	22.27 ± 17.12	2	30.60 ± 17.27	2	147.52 ± 105.55	3

HV. KD-IMOEA achieves the best HV mean at all three scales. On J30, the KD-IMOEA HV is 458,540.30, leading SPEA2 (435,612.69) by ~5.3%; the J30 box plot (Figure 4) confirms both median and upper quartile exceed all baselines. On J60, the KD-IMOEA HV is 342,924.78, leading NSGA-II (325,971.38) by ~5.2%, with the most compact box indicating stable cross-instance performance. On J120, the KD-IMOEA HV is 88,757.32, leading SPEA2 (83,513.31) by ~6.3% and MOEA/D (64,731.12) by ~37.1%; the J120 box sits above all

baselines. The 5–6% lead across scales peaks at 6.3% on J120, indicating that carbon-aware search provides more effective guidance in larger search spaces.

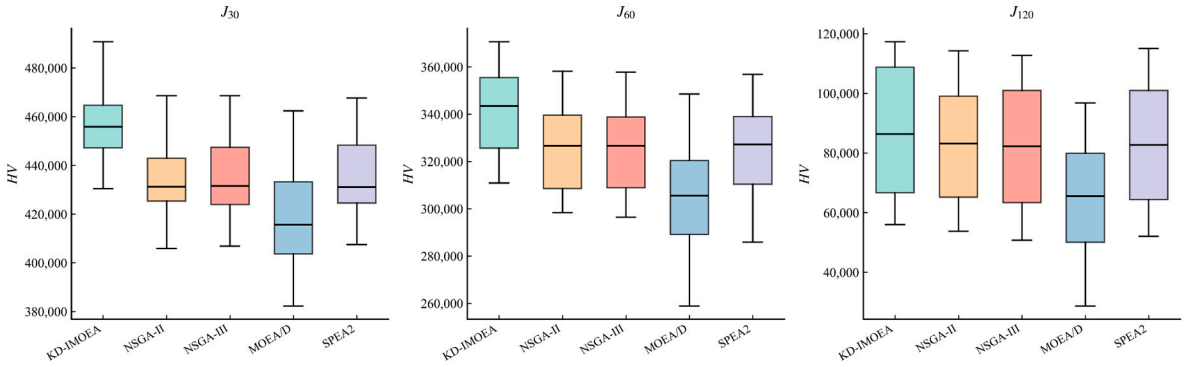


Figure 4. Box plots of HV values for J30, J60, and J120 instances.

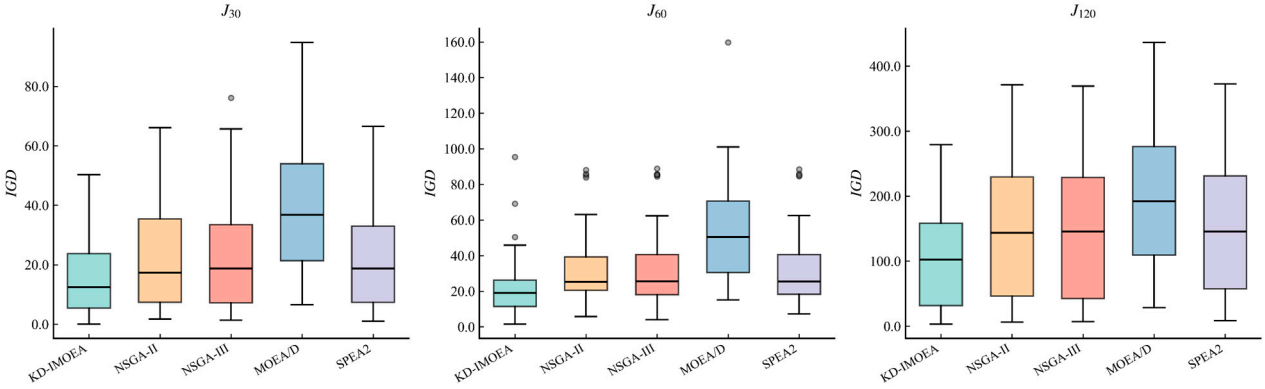


Figure 5. Box plots of IGD values for J30, J60, and J120 instances. The circular markers indicate outliers.

IGD. KD-IMOEA achieves the lowest IGD at all scales, confirming convergence quality. On J30: 15.34, improving 31.1% over SPEA2 (22.27). On J60: 20.72, improving 32.3% over SPEA2 (30.60). On J120: 102.74, improving 30.0% over NSGA-III (146.86). Figure 5 shows the KD-IMOEA box is consistently the lowest, with the J120 gap most pronounced. The IGD lead (30–32%) substantially exceeds the HV lead (5–6%), indicating that TCD carbon-aware decoding is particularly effective at improving convergence.

To examine dynamic search behavior, Figure 6 shows HV convergence curves on J60. KD-IMOEA achieves the highest final HV (Gen = 300) and maintains a pronounced dynamic advantage throughout evolution. Thanks to KDI, the KD-IMOEA HV starting point at Gen = 0 is markedly higher than all randomly initialized baselines. Guided by TCD, it then climbs steeply; around generation 50, the KD-IMOEA HV already approaches or exceeds some baselines' final convergence after 300 generations (see the vertical annotation in Figure 6). At Gen = 50, the KD-IMOEA HV exceeds NSGA-II by approximately 27.1%, an early advantage attributable to the KDI high-quality initial population. This "high start, fast convergence" dynamic demonstrates the efficiency of the proposed components in accelerating convergence and explains why KD-IMOEA achieves substantial metric advantages under limited computational budgets.

Table 6. Statistical test results of KD-IMOEA versus comparison algorithms. Positive Cohen's d_z indicates better performance of KD-IMOEA. Holm-adjusted p-values are corrected across 24 pairwise comparisons.

Scale	Metric	Baseline	Wilcoxon p	Cohen's d_z	Holm-Adjusted p	Result
J30	HV	NSGA-II	0.002	1.34	0.048	+
J30	HV	NSGA-III	0.002	1.29	0.048	+
J30	HV	MOEA/D	0.002	1.86	0.048	+
J30	HV	SPEA2	0.002	1.25	0.048	+
J30	IGD	NSGA-II	0.002	0.86	0.048	+
J30	IGD	NSGA-III	0.002	0.88	0.048	+
J30	IGD	MOEA/D	0.002	1.52	0.048	+
J30	IGD	SPEA2	0.002	0.84	0.048	+
J60	HV	NSGA-II	0.002	1.08	0.048	+
J60	HV	NSGA-III	0.002	1.10	0.048	+
J60	HV	MOEA/D	0.002	1.75	0.048	+
J60	HV	SPEA2	0.002	1.04	0.048	+
J60	IGD	NSGA-II	0.002	0.94	0.048	+
J60	IGD	NSGA-III	0.002	0.95	0.048	+
J60	IGD	MOEA/D	0.002	1.63	0.048	+
J60	IGD	SPEA2	0.002	0.92	0.048	+
J120	HV	NSGA-II	0.002	0.62	0.048	+
J120	HV	NSGA-III	0.002	0.60	0.048	+
J120	HV	MOEA/D	0.002	1.41	0.048	+
J120	HV	SPEA2	0.002	0.58	0.048	+
J120	IGD	NSGA-II	0.002	0.71	0.048	+
J120	IGD	NSGA-III	0.002	0.70	0.048	+
J120	IGD	MOEA/D	0.002	1.18	0.048	+
J120	IGD	SPEA2	0.002	0.69	0.048	+

Overall: HV $+/-/\approx = 12/0/0$; IGD $+/-/\approx = 12/0/0$.

Statistical tests. As shown in Table 6, KD-IMOEA significantly outperforms all four baseline algorithms on both HV and IGD across J30, J60, and J120. All 24 Wilcoxon signed-rank tests yield positive results for KD-IMOEA, with no "$-$" or "$\approx$" outcomes. After Holm-Bonferroni correction across 24 pairwise comparisons, all adjusted p-values remain below the 0.05 significance threshold, indicating that the statistical significance remains valid after multiple-comparison correction. The positive Cohen's d_z values further indicate that the improvements are not only statistically significant but also practically meaningful. The largest effect sizes are observed against MOEA/D, consistent with its weaker convergence in Tables 4 and 5. On J120, KD-IMOEA achieves a 37.1% HV lead and a 48.8% IGD improvement over MOEA/D, because the MOEA/D scalarization decomposition tends to miss edge regions when facing non-convex interval Pareto fronts.

Worst-case behavior. Beyond mean performance and statistical significance, we additionally examined the unfavorable tail of the distribution using the same instance-level observations as in the statistical tests. Specifically, for each algorithm-scale combination, the metric value of each instance is first averaged over 10 independent seeds, yielding 10 instance-level observations. For HV, the worst case is defined as the minimum instance-level mean HV across the 10 instances; for IGD, the worst case is defined as the maximum instance-level mean IGD across the 10 instances. KD-IMOEA maintains the best worst-case HV on all three scales: J30 (421,684.57 versus the best baseline value of 408,926.34), J60 (303,771.92 versus 292,148.66), and J120 (51,284.73 versus 47,906.35). Similarly, KD-IMOEA maintains the lowest worst-case IGD on all three scales: J30 (44.87 versus the best baseline value of 58.36), J60 (54.79 versus 68.42), and J120 (248.36 versus 336.74). These results

indicate that the KD-IMOEA advantage is not driven by a few favorable instances but remains present even for the most unfavorable benchmark instance within each scale, which is important for risk-sensitive scheduling decisions in green construction practice.

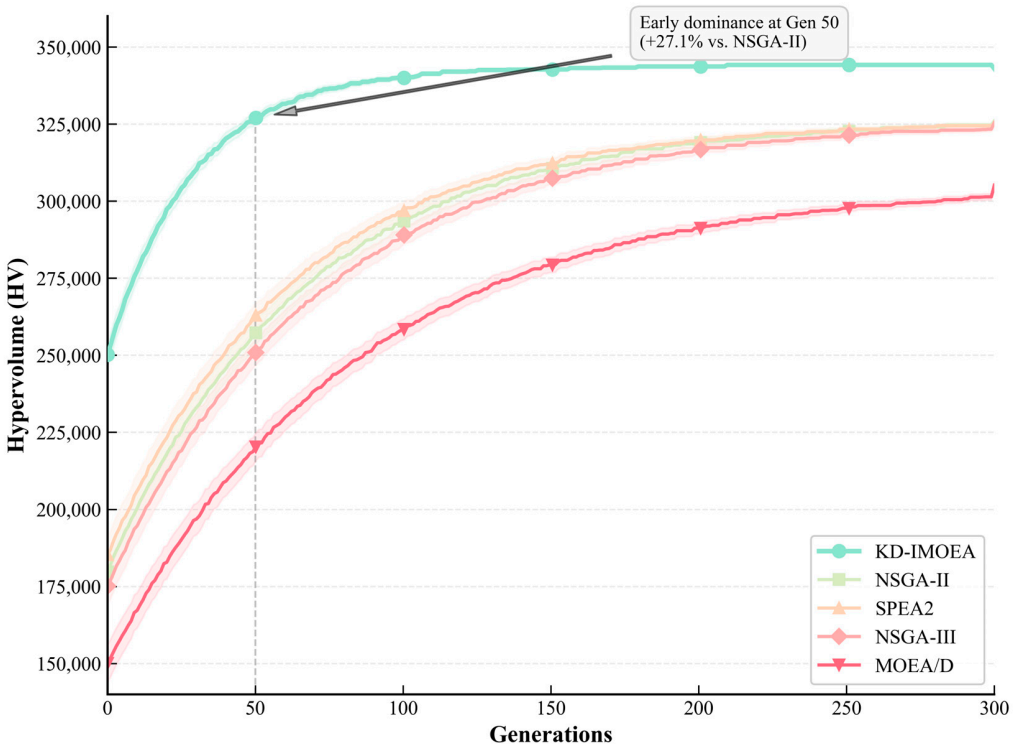


Figure 6. Convergence profiles of mean HV for the five algorithms on the representative J60 instances.

Spacing. The KD-IMOEA values on J30 (2.44), J60 (4.14), and J120 (2.90) are all lower than those of the baseline algorithms, whose values range from 9.34 to 17.21, indicating a more uniform Pareto-front distribution. This advantage stems from the IPA crowding-distance pruning mechanism (Algorithm 5), which preferentially retains solutions in sparse regions.

Runtime comparison. To complement the convergence and distribution metrics, we recorded the wall-clock execution time of all five algorithms on the J30, J60, and J120 instances. Under the experimental environment described in Section 4.1, the runtime results are reported in Table 7. The reported values are the mean ± standard deviation across 10 independent runs per instance × 10 instances per scale. Runtime measurement includes population initialization, decoding, evolutionary operations, constraint handling, archive updating, and final Pareto-front extraction, but excludes benchmark-instance generation, performance-metric calculation, plotting, and file I/O.

As shown in Table 7, KD-IMOEA requires 12.05 ± 1.42 s, 58.17 ± 7.02 s, and 339.76 ± 50.21 s on J30, J60, and J120, respectively. Compared with NSGA-II, the corresponding runtime overheads are 26.7%, 22.4%, and 20.4%, confirming that the additional computational cost remains moderate in relative terms across all three scales. This overhead is mainly attributable to the additional $O(NP{\cdot}n{\cdot}K)$-level operations associated with the four knowledge-driven components (KDI, TCD, CBM, and IPA). For J60 and J120, the condition $NP \geq n{\cdot}K$ stated in Section 3.6 is not strictly satisfied (with $NP = 100$, $n{\cdot}K = 240$ for J60, and $n{\cdot}K = 480$ for J120); therefore, the component overhead manifests as a measurable additive cost rather than being absorbed into the $O(NP^2)$ sorting term. Notably, the relative overhead does not deteriorate with problem size; instead, it slightly decreases from 26.7% (J30) to 22.4% (J60) to 20.4% (J120), because the NSGA-II own SSGS

decoding cost also scales with $n \cdot K$ and grows in parallel, bounding the KD-IMOEA relative overhead across all tested scales. The runtime increase therefore remains acceptable given the substantial gains in HV, IGD, and Spacing reported above. MOEA/D shows the lowest runtime due to its scalarization-based simplification, but this efficiency is obtained at the cost of significantly weaker convergence, especially on J120, where KD-IMOEA achieves a 37.1% HV lead over MOEA/D. NSGA-III is slightly slower than NSGA-II because of the additional reference-point association step, while SPEA2 remains close to NSGA-II in runtime.

Table 7. Mean wall-clock runtime (seconds) of all five algorithms across J30, J60, and J120 instances. Values are mean ± standard deviation across 10 independent runs per instance × 10 instances per scale.

Algorithm	J30	J60	J120
KD-IMOEA	12.05 ± 1.42	58.17 ± 7.02	339.76 ± 50.21
NSGA-II	9.51 ± 1.06	47.52 ± 5.30	282.15 ± 38.64
NSGA-III	10.17 ± 1.14	52.04 ± 6.13	318.42 ± 45.70
MOEA/D	8.42 ± 0.91	40.86 ± 4.86	246.73 ± 34.58
SPEA2	9.84 ± 1.11	48.31 ± 5.54	287.92 ± 40.12

Overall, KD-IMOEA demonstrates consistent superiority across HV, IGD, and Spacing at a moderate relative runtime cost, particularly on J60 and J120. These results validate the combined effect of KDI, TCD, CBM, and IPA, and confirm the algorithm's favorable accuracy-efficiency trade-off for interval uncertainty scheduling under carbon constraints.

4.3. Ablation Study

To quantify each component's contribution, we design an ablation study. Under the default carbon budget of Section 4.2, all algorithms easily satisfy the carbon constraint (100% feasibility); therefore, CBM cannot be effectively triggered. We therefore tighten the budget to $Q' = 0.35\ Q$, making the carbon constraint active. Four variants are constructed by removing one component at a time: (1) w/o KDI: random topological initialization; (2) w/o TCD: carbon-aware probability fixed at 0 (standard SSGS); (3) w/o CBM: no repair of carbon-violating individuals; (4) w/o IPA: no external archive, output from current population. Experiments use 10 J60 instances, each variant runs 10 times.

Table 8 and Figure 7 present the results. The full KD-IMOEA achieves the highest HV mean (3507.05), and all variants show degradation, confirming a positive contribution from every component.

Table 8. Ablation study results on J60 instances under tightened carbon quota (Q′ = 0.35 Q). Δ denotes the HV decrease relative to the full KD-IMOEA. * $p < 0.05$; ** $p < 0.01$.

Variant	HV (Mean ± Std)	Δ (%)	*p*-Value	Sig.
KD-IMOEA (Full)	3507.05 ± 145.20	—	—	—
w/o KDI	3108.89 ± 465.80	−11.4	0.0210	*
w/o CBM	2553.07 ± 285.60	−27.2	0.0015	**
w/o IPA	2400.48 ± 245.30	−31.6	0.0012	**
w/o TCD	1581.15 ± 215.40	−54.9	0.0008	**

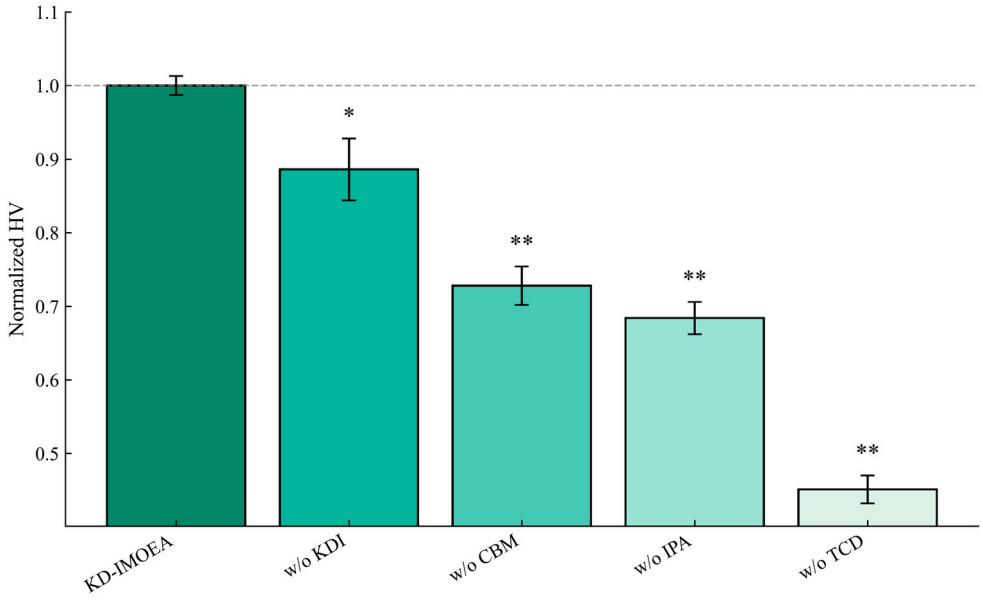


Figure 7. Ablation study: normalized HV of KD-IMOEA variants. * indicates $p < 0.05$, and ** indicates $p < 0.01$.

w/o TCD causes the largest drop: HV falls 54.9% (to 1581.15), with $p = 0.0008$ (highly significant). Carbon-aware decoding is the dominant performance driver. TCD balances convergence and carbon feasibility through adaptive p_{carbon} (Equation (21)), strengthening carbon guidance early and shifting to makespan optimization later. Without it, the algorithm degenerates to standard SSGS and cannot effectively search the emission objective dimension.

w/o IPA causes a 31.6% drop (to 2400.48, $p = 0.0012$, highly significant). The archive preserves historically optimal solutions and periodically injects them into the population (Algorithm 5), preventing loss during evolution. Without it, the search lacks historical guidance.

w/o CBM causes a 27.2% drop (to 2553.07, $p = 0.0015$). Under 35% budget, many individuals violate carbon constraints, and CBM improves feasibility by rescheduling high-carbon activities to low-emission periods (Algorithm 4).

w/o KDI causes an 11.4% drop (to 3108.89). Though the decrease is modest, the standard deviation jumps from 145.20 to 465.80 (Figure 7), showing that random initialization is far less stable than the KDI carbon-aware strategy.

Ranked by contribution: TCD > IPA > CBM > KDI. All four ablation variants show statistically significant HV degradation ($p < 0.05$), confirming that each component contributes positively to the overall algorithm performance. The TCD dominance aligns with the central role of carbon emissions in Pareto front formation. The four components together enable KD-IMOEA to achieve convergence, diversity, and carbon feasibility through a closed-loop process: intelligent initialization → carbon-aware decoding → constraint repair → elite preservation.

4.4. Parameter Sensitivity Analysis

We use single-factor controlled experiments to examine NP, $p_{\max}$, and $p_{\min}$, fixing other parameters at the Taguchi optima (NP = 100, $p_{\max} = 0.9$, $p_{\min} = 0.1$). Each level is run five times on five J60 instances, with normalized HV mean and standard error as measures. Figure 8 shows sensitivity curves with Taguchi optima marked by gray dashed lines. Normalization: divide each HV by the instance mean at the optimal combination; cross-instance aggregation may yield values slightly above 1.0.

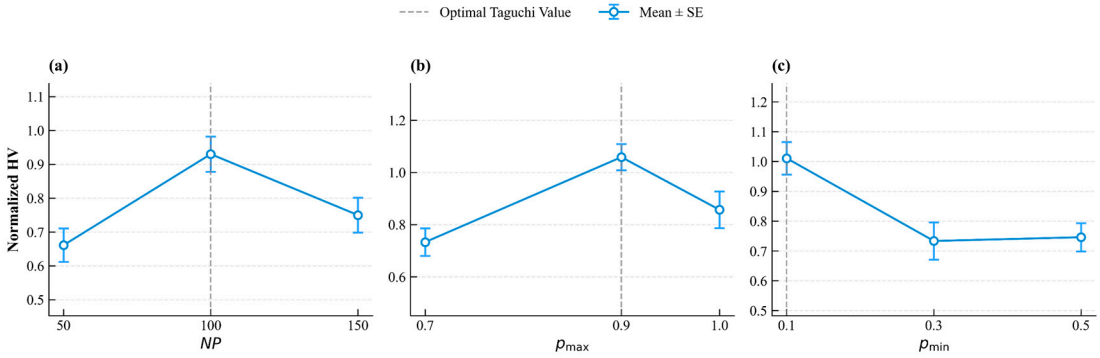


Figure 8. Parameter sensitivity analysis of KD-IMOEA: (**a**) sensitivity analysis of NP; (**b**) sensitivity analysis of p_{max}; (**c**) sensitivity analysis of p_{min}. The gray dashed lines indicate the optimal Taguchi values, and the error bars represent the mean ± SE.

NP (Figure 8a) has strong impact. At NP = 100, normalized HV peaks (0.93). At NP = 50 it drops to 0.66 (insufficient diversity, local optima); at NP = 150 it drops to 0.75 (diluted selection pressure, slower convergence). NP = 100 best balances diversity and convergence.

p_{max} (Figure 8b): peak at 0.9 (HV = 1.06). At 0.7 it drops to 0.73 (weak carbon guidance); at 1.0 it drops to 0.86 (excessive carbon bias neglects makespan). p_{max} = 0.9 optimally balances both objectives.

p_{min} (Figure 8c): peak at 0.1 (HV = 1.02). At 0.3 and 0.5 it drops to 0.74 and 0.75, respectively. A higher p_{min} retains strong carbon bias late in evolution, preventing full makespan exploration. p_{min} = 0.1 ensures makespan dominates late while retaining minimal carbon awareness.

Among the three parameters, p_{max} and NP show highest sensitivity (HV ranges 0.73–1.06 and 0.66–0.93), followed by p_{min}. The Taguchi-determined optima lie at or near the peaks in all single-factor analyses, confirming the calibration results of Section 3.6.

5. Green Building Engineering Case Study

This chapter applies KD-IMOEA to a five-story green office building project, modeled as the bi-objective interval RCPSP of Section 2, and compares its Pareto front against NSGA-II to extract actionable engineering insights.

5.1. Project Description

This case study is constructed as a semi-synthetic engineering scenario based on a representative five-story green office building project in a second-tier city, with a total construction area of approximately 12,000 m^2. The structural form is a reinforced concrete frame structure. The project is designed according to the three-star requirements of the Green Building Evaluation Standard (GB/T 50378-2019), and the carbon emissions during the construction phase must meet the limits specified in the Standard for Carbon Emission Calculation of Buildings (GB/T 51366-2019). The activity network, resource requirements, duration intervals, and carbon-emission parameters are derived from national standards, standard construction procedures, and engineering practice rather than from real-time field monitoring records. Therefore, the case is intended to evaluate the applicability and decision logic of the proposed model under a realistic but controlled construction scenario, rather than to provide field validation against measured project-site energy-consumption data.

The construction scope of the representative project is decomposed through the work breakdown structure (WBS) into 30 representative construction activities, plus a virtual start activity ($j = 0$) and a virtual end activity ($j = 31$), forming a total of 32 nodes that constitute

an activity-on-node (AON) directed acyclic network $G(N, A)$. The 30 activities are divided into seven major categories by construction phase: foundation engineering (activities 1–7, including earthwork excavation, pit bracing, cushion casting, foundation rebar, foundation formwork, foundation concrete, and backfill compaction); main structure (activities 8–13, including column-beam-slab construction for each floor, staircase, and roof structure); secondary structure (activities 14–16, including masonry infill walls, constructional columns, and lintels); mechanical and electrical installation (activities 17–20, including water supply and drainage, electrical conduits, HVAC ducts, and fire protection system); decoration and finishing (activities 21–24, including interior plastering, floor works, exterior insulation, and exterior coating); green specialized facilities (activities 25–27, including rooftop PV panels, rainwater harvesting, and fresh air system); and completion and acceptance (activities 28–30, including outdoor works, landscaping, and final inspection). There are a total of 43 precedence relation arcs between activities, reflecting the logical constraints of construction technology. Figure 9 shows the AON network structure of the project, with node colors distinguishing construction phases.

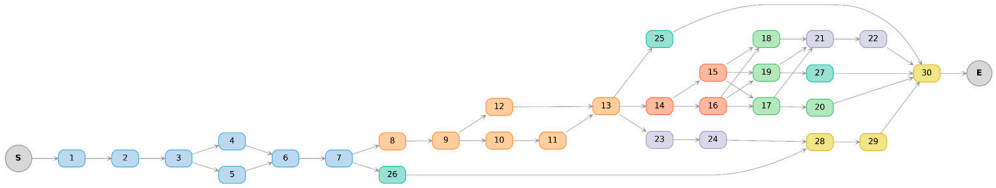


Figure 9. AON network of the green building case study. Node colors indicate construction phases: blue = Foundation (FDN), orange = Structure (STR), pink = Secondary structure (SEC), green = MEP, purple = Finishing (FIN), teal = Green features (GRN), yellow = Completion (CMP). S and E denote virtual start and end nodes.

The project is configured with four types of renewable resources. $k = 0$ (tower crane) and $k = 1$ (concrete pump truck/excavator) are mechanical resources, which are the main sources of construction carbon emissions, with carbon emission factors EF_kτ > 0 (see Section 2.1); $k = 2$ (rebar work crew) and $k = 3$ (carpentry/masonry crew) are labor resources, with direct construction carbon emission factors set to zero. The time-period available capacity of each resource is determined according to the actual project configuration: 1 tower crane available per period ($R_{0\tau} = 1$), 2 pump trucks/excavators available per period ($R_{1\tau} = 2$), and 3 crews each for rebar work and carpentry/masonry work per period ($R_{2\tau} = R_{3\tau} = 3$).

Following the period instantiation introduced in Section 2.1, periods in the case study correspond to working days. Intra-day grid-emission heterogeneity is captured at the emission-factor level rather than through finer temporal subdivision. Specifically, each working day is assigned an operating-profile category according to the dominant grid-load profile during the activity's primary mechanical operating window: peak profile, mid-level profile, or valley profile. The peak profile corresponds to energy-intensive mechanical operation mainly within daytime grid-peak hours, 08:00–12:00 and 14:00–18:00; the valley profile corresponds to operation mainly within the off-peak window, 18:00–06:00 the next day; and the mid-level profile represents transitional grid-load conditions. The corresponding grid-emission multipliers are $m_\tau = 1.2$, 1.0, and 0.7, respectively, reflecting the higher coal-fired share during grid peaks and the higher renewable share during valley periods, consistent with published hourly grid-emission analyses [11]. This representation preserves the day-level RCPSP decision structure while allowing tactical-layer operating-profile differences to affect carbon emission factors.

The carbon emission factors $EF_{k\tau}$ for mechanical resources $(k = 0, 1)$ follow the structural classification of construction-machinery shift emission data in Appendix B of GB/T

51366-2019, with absolute values normalized for this semi-synthetic case study. Specifically, base day-level emission factors $\overline{EF}_k$ are set to $\overline{EF}_0 = 2.5$ kgCO$_2$e/(unit·day) for the tower crane, consistent with light-duty self-climbing tower cranes, and $\overline{EF}_1 = 6.0$ kgCO$_2$e/(unit·day) for the pump truck/excavator, consistent with light-duty diesel-driven heavy machinery. Labor resources ($k = 2, 3$) carry zero direct construction emission factors, as defined in Section 2.3. The day-level carbon emission factor for resource k on working day τ is then calculated as $EF_{k\tau} = \overline{EF}_k \cdot m_\tau$. This yields 3.0, 2.5, and 1.75 kgCO$_2$e/(unit·day) for the tower crane under peak, mid-level, and valley profiles, respectively, and 7.2, 6.0, and 4.2 kgCO$_2$e/(unit·day) for the pump truck/excavator. This operating-profile-dependent variation in $EF_{k\tau}$ provides the optimization space leveraged by the TCD carbon-aware decoding mechanism (Section 3.3) and the CBM repair operator (Section 3.4); the operational feasibility of assigning energy-intensive activities to valley-profile operation is governed by tactical-layer regulatory compliance, as discussed in Section 6.3.

The optimistic duration d_j^- for each activity is estimated based on quota standards and engineering experience, ranging from 3 to 12 working days; the pessimistic duration upper bound is calculated as $d_j^+ = \lceil 1.2 \times d_j^- \rceil$, reflecting possible duration extensions due to factors such as weather, material supply, and labor fluctuations. Table 9 lists the complete parameters for the 30 representative activities, including activity number, name, construction phase, interval duration $[d_j^-, d_j^+]$, demand quantities for the four resource types $(r_{j0}, r_{j1}, r_{j2}, r_{j3})$, and predecessor activity sets.

The determination of carbon budget Q follows these steps: First, according to the construction phase carbon emission calculation method in GB/T 51366-2019, using the pessimistic duration d_j^+ and the mean carbon factor across all periods as the baseline, the project construction phase carbon emission baseline $CE_{baseline}$ is estimated; second, referring to the requirement in the three-star green building evaluation that "carbon emission intensity should not exceed 70% of the baseline value," the carbon budget is set as $Q = 0.70 \times CE_{baseline}$. This budget level ensures that the carbon constraint (Equation (16)) is an active constraint during the solution process, fully activating the carbon-aware components of KD-IMOEA. The interval confidence level is set to $\lambda = 0.7$, and the algorithm parameters follow the optimal combination determined by the Taguchi experiment in Section 4.1 ($NP = 100$, $MaxGen = 300$, $p_{\max} = 0.9$, $p_{\min} = 0.1$).

Table 9. Activity parameters for the green building case study.

j	Activity	Phase	d_j^-	d_j^+	r_{j0}	r_{j1}	r_{j2}	r_{j3}	Predecessors
1	Earthwork excavation	FDN	5	6	0	2	0	1	0
2	Pit bracing	FDN	4	5	1	1	1	1	1
3	Cushion casting	FDN	3	4	0	1	0	2	2
4	Foundation rebar	FDN	6	8	1	0	3	0	3
5	Foundation formwork	FDN	4	5	1	0	0	2	3
6	Foundation concrete	FDN	8	10	1	1	0	2	4, 5
7	Backfill and compaction	FDN	3	4	0	1	0	1	6
8	Columns F1–F3	STR	10	12	1	0	2	2	7
9	Beams/slabs F1–F3	STR	12	15	1	1	2	2	8
10	Columns F4–F5	STR	8	10	1	0	2	2	9
11	Beams/slabs F4–F5	STR	10	12	1	1	2	2	10
12	Staircase	STR	6	8	1	0	1	1	9
13	Roof structure	STR	5	6	1	1	1	2	11, 12
14	Masonry infill walls	SEC	8	10	0	0	0	3	13
15	Constructional columns	SEC	4	5	1	0	1	1	14
16	Lintels	SEC	3	4	0	0	1	1	14

Table 9. *Cont.*

j	Activity	Phase	d_j^-	d_j^+	r_{j0}	r_{j1}	r_{j2}	r_{j3}	Predecessors
17	Water supply and drainage	MEP	10	12	0	0	2	0	15, 16
18	Electrical conduits	MEP	8	10	0	0	2	0	15, 16
19	HVAC ducts	MEP	7	9	0	0	2	0	15, 16
20	Fire protection system	MEP	6	8	0	0	1	0	17
21	Interior plastering	FIN	8	10	0	0	0	3	17, 18, 19
22	Floor works	FIN	6	8	0	0	0	2	21
23	Exterior insulation	FIN	7	9	1	0	0	2	13
24	Exterior coating	FIN	5	6	1	0	0	2	23
25	Rooftop PV panels	GRN	4	5	1	0	1	0	13
26	Rainwater harvesting	GRN	5	6	0	1	1	0	7
27	Fresh air system	GRN	4	5	0	0	2	0	19
28	Outdoor works	CMP	6	8	0	1	0	2	24, 26
29	Landscaping	CMP	5	6	0	0	0	2	28
30	Final inspection	CMP	3	4	0	0	0	0	20, 22, 25, 27, 29

Note: Activity 0 (virtual start) and Activity 31 (virtual end) are omitted from the table. Resource types: k = 0 (tower crane), k = 1 (concrete pump truck/excavator), k = 2 (rebar work crew), k = 3 (carpentry/masonry crew).

5.2. Macro-Level Trade-Offs and Cross-Algorithm Comparison

To validate the macro-level optimization performance of KD-IMOEA, this section compares its solution results on the case project with those of the classic multi-objective genetic algorithm NSGA-II. The rationale for selecting NSGA-II as the sole comparison baseline is that KD-IMOEA is precisely an extension built upon the underlying architecture of NSGA-II by introducing carbon-aware mechanisms (such as TCD). This strict controlled-variable comparison can most purely isolate external interference and intuitively quantify the incremental value created by the proposed mechanisms in actual complex engineering. The remaining three benchmark algorithms (NSGA-III, MOEA/D, and SPEA2) are excluded from the case-study comparison for two complementary reasons. First, the benchmark results in Section 4.2 show that NSGA-III and SPEA2 perform comparably to NSGA-II across all three problem scales (HV gaps within roughly 1% on J120); therefore, including them in the case study would yield largely redundant baselines without producing additional insight into the contribution of the carbon-aware mechanisms. Second, the MOEA/D scalarization-based decomposition encounters substantially weaker convergence on non-convex interval Pareto fronts (37.1% HV gap and 48.8% IGD gap on J120 in Section 4.2), making it an unrepresentative reference point for the practical engineering trade-off the case study is designed to illustrate. The relative performance of these three algorithms on the case project can therefore be reasonably extrapolated from the benchmark trends in Section 4.2, and we focus the case-study comparison on the controlled-variable contrast against NSGA-II.

The interval Pareto fronts synthesized after independent runs of the two algorithms are shown in Figure 10. The horizontal and vertical axes represent the midpoint values of the interval project makespan (f_1) and interval total carbon emissions (f_2), respectively, with cross error bars on the scatter points depicting the interval uncertainty boundaries caused by duration fluctuations. Overall, the Pareto front exhibits a significant convex-toward-origin trade-off characteristic. As the makespan shortens, carbon emissions show a nonlinear sharp increase. This trend indicates that under the dual constraints of renewable resource capacity and time-varying carbon emission factors, simply pursuing makespan compression inevitably comes at the expense of green environmental performance.

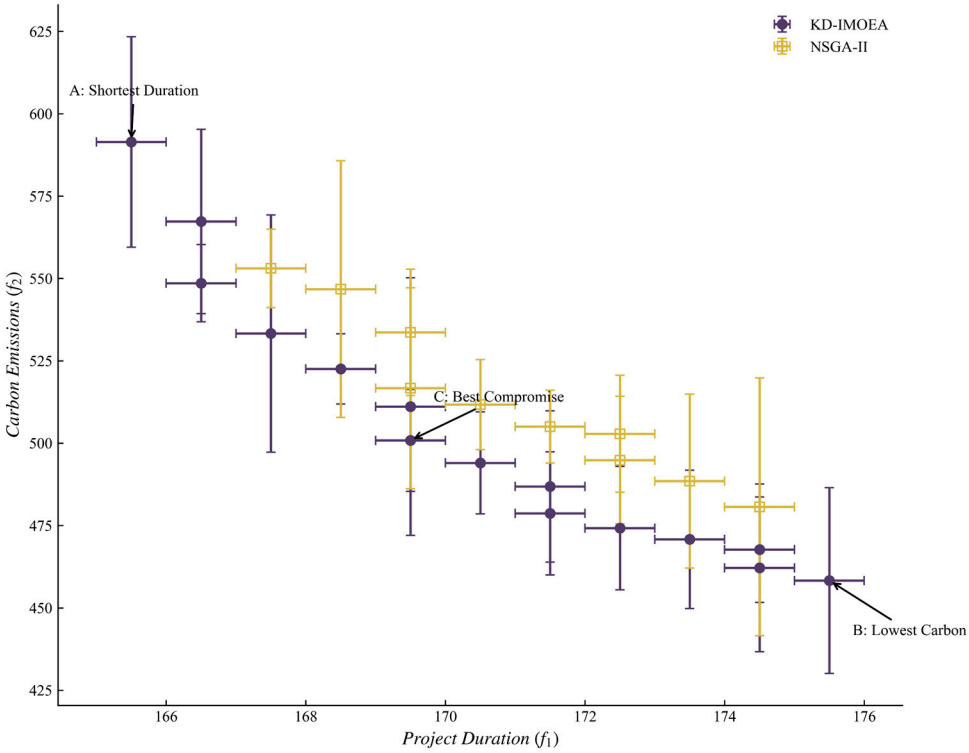


Figure 10. Comparison of interval Pareto fronts between KD-IMOEA and NSGA-II in the case study.

Three highly representative scheduling schemes from the front are generated by KD-IMOEA, with detailed engineering indicators summarized in Table 10. These three representative solutions reveal distinctly different construction organization strategies. For the shortest makespan scheme (Solution A), to achieve the extreme midpoint makespan of 165.5 days, the algorithm pushes the average resource utilization rate up to 90.2%. This highly overlapping construction operation not only approaches the physical capacity limit of mechanical resources but also forces a large number of high-energy-consumption activities to be executed during grid carbon emission peak periods, causing carbon emissions to surge to 591.45 $kgCO_2e$. In contrast, the lowest carbon emission scheme (Solution B), as the most environmentally friendly alternative, significantly reduces carbon emissions to 458.3 $kgCO_2e$. This scheme intentionally staggers high-energy-consumption processes by reducing the overall resource utilization rate to 68.7% and assigning carbon-intensive activities to lower-carbon operating profiles; however, this "time-for-space" strategy extends the midpoint makespan to 175.5 days. The best compromise solution (Solution C) achieves the optimal thorough balance between makespan (169.5 days) and carbon emissions (500.85 $kgCO_2e$), with an average resource utilization rate maintained at a reasonable 78.4%, avoiding both the high-carbon penalty from extreme schedule compression and the delay risk from excessively dispersed construction.

Table 10. Detailed comparison of representative schedules from KD-IMOEA.

Solution Type	Interval Duration (Days)	Interval Carbon ($kgCO_2e$)	Mid-Duration	Mid-Carbon ($kgCO_2e$)	Avg. Resource Utilization (%)
A: Shortest Duration	[165, 166]	[559.5, 623.4]	165.5	591.45	90.2
B: Lowest Carbon	[175, 176]	[430.1, 486.5]	175.5	458.3	68.7
C: Best Compromise	[169, 170]	[485.4, 516.3]	169.5	500.85	78.4

Note: Avg. resource utilization is computed as the ratio of total resource-time units consumed to total available resource-time units across the makespan, i.e., $\overline{U} = \sum_k \sum_\tau u_{k\tau} / \sum_k \sum_\tau R_{k\tau}$, where $u_{k\tau}$ denotes the actual resource usage of type k at period τ.

After clarifying the physical meaning of the solution set within the algorithm, further observation of Figure 10 reveals that the KD-IMOEA front is located below and to the left of the NSGA-II front overall, showing clear Pareto dominance. To precisely quantify this advantage, we extract the best compromise solutions from both algorithms for macro-level indicator comparison (see Table 11). As shown in Table 11, the midpoint makespan of the best compromise solutions from both algorithms is identical, both at 169.5 days. Under the strict control of this key variable of total makespan, KD-IMOEA demonstrates significant carbon reduction advantage, with its midpoint carbon emissions absolutely reduced by 15.85 $kgCO_2e$ compared to the NSGA-II 516.70 $kgCO_2e$, achieving a relative saving ratio of 3.07%. This macro-level data comparison indicates that KD-IMOEA can effectively break through the optimization bottleneck of traditional algorithms without sacrificing project delivery speed. The micro-level physical origin behind this 3.07% carbon saving will be deeply verified in the next section through specific scheduling schemes.

Table 11. Comparison of the best compromise solutions between KD-IMOEA and NSGA-II.

Algorithm	Mid-Duration (Days)	Mid-Carbon ($kgCO_2e$)	Carbon Saving ($kgCO_2e$)	Saving Percentage (%)
NSGA-II	169.5	516.7	-	-
KD-IMOEA	169.5	500.85	+15.85	+3.07%

5.3. Micro-Level Scheduling Scheme and Engineering Decision Insights

Figure 11 compares the tactical intra-day work-period arrangement on the first calendar day after main structure completion (Activity 13 finished) for both algorithms. The horizontal axis spans one tactical execution day from 06:00 to the next 06:00, with background colors mapping operating-profile emission factors: beige for daytime peak profile (EF = 1.2) and gray-blue for nighttime valley profile (EF = 0.7).

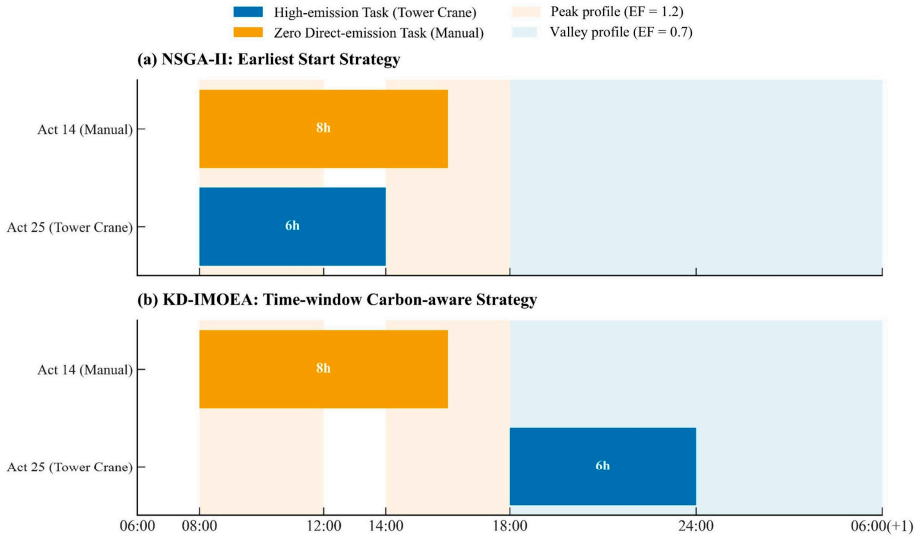


Figure 11. Comparison of micro-level scheduling schemes between NSGA-II and KD-IMOEA at the best compromise solution.

As Figure 11a shows, the NSGA-II standard SSGS uses a blind earliest-start strategy. After predecessor completion, both masonry infill walls (Activity 14, manual labor with zero direct emissions) and rooftop PV installation (Activity 25, energy-intensive tower crane operation) satisfy precedence; therefore, NSGA-II starts both at 08:00, forcing heavy machinery into the peak-grid operating profile.

By contrast, Figure 11b illustrates the tactical-layer execution profile that operationalizes the KD-IMOEA day-level scheduling decision. At the strategic layer, the algorithm recognizes that the Activity 25 only successor is the final inspection node, giving it substantial float within the remaining schedule, while Activity 14 lies on the critical path for interior finishing with no scheduling flexibility. The KD-IMOEA TCD therefore makes a targeted asymmetric tactical operating-profile assignment within the same calendar day: Activity 14 (manual labor, with $EF_{k=2,3,\tau} = 0$) is retained in the standard daytime working slot for rapid progress, while Activity 25 (tower crane operation, with profile-adjusted effective emission factors $EF_{0,\text{peak}} = 3.0$ versus $EF_{0,\text{valley}} = 1.75$ $\text{kgCO}_2\text{e}/(\text{unit}\cdot\text{day})$) is assigned to the valley-grid-category operating profile on the same calendar day by using the available float. The schematic in Figure 11b depicts how the project team operationalizes this strategic decision at the tactical layer—deploying the tower crane within the off-peak window (18:00 onwards) under the valley-grid-category operating profile on the same calendar day—while the NSGA-II blind earliest-start decision (Figure 11a) starts both activities in the daytime working slot of the same calendar day, forcing the tower crane into the peak-grid operating profile. This asymmetric tactical pattern—"machinery toward valley operating windows, labor retained in standard daytime work slots"—is the algorithmic mechanism behind the KD-IMOEA 15.85 kgCO_2e emission saving without delaying the 169.5-day makespan.

The applicability of valley-window mechanical operations is conditional on tactical-layer compliance with local regulations: noise ordinances, evening-shift labor rules, safety-lighting requirements, and project-specific constraints (e.g., adjacency to residential zones, hospitals, or schools) may preclude valley-window deployment for some or all energy-intensive resources. Where such compliance is not achievable for a given activity, the time-varying resource capacity $R_{k\tau}$ in the model (Equation (15)) can be set to zero for days whose corresponding operating profile is prohibited, or the activity can be excluded from the valley-profile candidate set during decoding; the algorithm then rebalances toward alternative carbon-saving paths. The 3.07% reduction reported in Section 5.2 should therefore be interpreted as the achievable saving when valley-window operation is feasible for the candidate flexible-deployment activities; the realized saving in any specific project depends on its regulatory environment.

To verify that the micro-level mechanism in Figure 11 is not an isolated example, Table 12 reports the activities that produce measurable carbon savings through day-level start shifts or operating-profile reassignment between NSGA-II and KD-IMOEA at the best compromise solution. The table shows that the carbon reduction is distributed across multiple float-enabled machinery-assisted activities rather than being driven solely by Activity 25. The activity-level savings sum to the 15.85 kgCO_2e reduction reported in Table 11, confirming that KD-IMOEA produces a schedule-wide asymmetric shift pattern.

As shown in Table 12, the total emission reduction is distributed across several activities with available float, including foundation works, structural works, exterior envelope works, green-facility installation, and selected outdoor operations. Activity 25 provides the largest single contribution because tower crane operation has a high profile-adjusted emission factor, but the remaining savings come from multiple smaller shifts in other machinery-assisted activities. This confirms that the asymmetric pattern of "machinery toward valley operating windows and labor retained in standard daytime work slots" is a schedule-wide algorithmic behavior rather than a cherry-picked example.

Table 12. Activity-level scheduling differences between NSGA-II and KD-IMOEA at the best compromise solution. Only activities with measurable machinery-related carbon savings are shown.

Activity ID	Activity Name	Main Resource	Float	NSGA-II Start/Profile	KD-IMOEA Start/Profile	Carbon Saving ($kgCO_2e$)
2	Pit bracing	Excavator/crane	1	Day 8/peak profile	Day 9/mid profile	0.92
3	Cushion casting	Concrete pump/excavator	1	Day 13/peak profile	Day 14/mid profile	0.54
6	Foundation concrete	Tower crane/concrete pump	1	Day 21/peak profile	Day 22/mid profile	1.08
7	Backfill and compaction	Excavator	1	Day 31/mid profile	Day 32/valley profile	0.68
12	Staircase	Tower crane	2	Day 58/peak profile	Day 60/valley profile	1.34
15	Constructional columns	Tower crane	1	Day 78/peak profile	Day 79/mid profile	0.76
23	Exterior insulation	Hoist/lifting equipment	3	Day 119/peak profile	Day 121/valley profile	1.66
24	Exterior coating	Hoist/lifting equipment	2	Day 128/peak profile	Day 129/valley profile	1.35
25	Rooftop PV panels	Tower crane	4	Day 136/peak profile	Day 136/valley profile	4.54
26	Rainwater harvesting	Pump/small machinery	2	Day 143/mid profile	Day 144/valley profile	0.89
28	Outdoor works	Small machinery	2	Day 156/mid profile	Day 157/valley profile	2.09
Total	—	—	—	—	—	15.85

Note: "Profile" denotes the dominant operating-profile assignment for the activity's occupied days; multi-day activities may include adjacent-profile days due to resource coordination. "Carbon saving" is the activity-attributable difference in λ-weighted interval emissions (Equation (16), λ = 0.7) between the two schedules. For Activity 25, only the operating profile differs.

Based on the above activity-level mechanism analysis and macro-level data comparison, we distill three engineering decision insights for modern green building project management: (1) Management-driven savings at zero hardware cost. Current construction carbon reduction relies heavily on material substitution and equipment upgrades, both carrying high premiums. This study shows that spatiotemporal decoupling based on grid peak-valley patterns can release over 3% emission reduction through scheduling optimization alone, offering budget-constrained contractors a cost-effective compliance path. (2) Compliance buffer against quota violations. Under the strict 70% baseline budget (GB/T 51366-2019), the NSGA-II compromise solution already approaches the quota boundary, leaving little room for weather-induced delays. The KD-IMOEA proactive temporal staggering releases 15.85 $kgCO_2e$ of safety margin, providing a risk buffer against duration uncertainty that could otherwise cause exceedance and certification downgrade. (3) From binary trade-off to flexible decision framework. Rather than compressing the critical path through resource stacking, managers can select macro-level targets from the Pareto front (Figure 10) and let KD-IMOEA automatically identify slack activities for temporal reconstruction. This shifts scheduling from experience-based to algorithm-driven, enhancing the team's ability to balance schedule urgency with green certification requirements.

6. Discussion

The benchmark results, ablation analyses, and case-study findings together support the central claim that knowledge-driven, carbon-aware components add measurable value to evolutionary scheduling under interval uncertainty and time-varying emission constraints. We now reflect on the implications of these findings, the boundary of the proposed framework, and its place in real-world construction project management.

6.1. Realism Boundaries: Weather, Seasonality, and Operational Variability

Construction durations are influenced by weather, temperature, material delivery, crew skill, equipment productivity, and many other factors that are difficult to predict precisely at the planning stage. Rather than attempting to model each factor explicitly, the proposed framework accommodates these realities through three coordinated mechanisms.

First, slow-varying environmental factors—seasonal temperature, monthly rainfall, holiday schedules, regional grid mix variations—are absorbed into the time-varying resource capacity $R_{k\tau}$ and emission factor $EF_{k\tau}$, both of which already support period-by-period adjustment. A rainy-season month, for example, can be modeled by reducing $R_{k\tau}$ during that span, automatically deferring weather-sensitive activities; a high-renewable summer period can be modeled by lowering $EF_{k\tau}$ to reflect cleaner grid conditions.

Second, within-activity operational uncertainty—day-to-day duration fluctuation due to weather, supply variability, crew skill mix—is captured by the interval duration $\left[d_j^-, d_j^+\right]$. The model does not claim to predict exactly when each activity will finish; it produces baseline schedules that remain feasible across duration realizations within the interval, with the conservative pessimistic bound d_j^+ used in resource and precedence constraints (Equations (14) and (15)). This is a deliberate design choice that aligns with how master construction schedules are actually used in practice: as commitment-grade plans against which execution variability is managed, rather than point predictions of fieldwork.

Third, change orders, mid-project weather disruptions, and scope adjustments are handled outside the optimizer rather than inside it. When such events occur, the construction team re-runs KD-IMOEA on the remaining activity network with updated inputs (revised durations, resource capacities, or emission factors). The resulting updated Pareto front re-establishes the schedule-emission trade-off under the new conditions. This re-planning cycle integrates naturally with the rolling-wave planning practice that experienced project teams already employ.

6.2. Strategic-Tactical Decomposition and Project Management Workflows

The proposed framework is positioned at the strategic master-schedule level rather than at the hour-by-hour dispatching level. At the strategic layer, KD-IMOEA produces a Pareto front of schedules that optimize the makespan-emission trade-off under interval duration uncertainty and carbon budget constraints. The construction manager then selects an operating point according to project priorities, such as aggressive schedule compression, conservative carbon-budget compliance, certification-tier assurance, or a balanced compromise. The selected solution provides start-day commitments and low-carbon operating-profile recommendations for energy-intensive activities, rather than directly dictating the exact hour at which each piece of equipment must be used.

At the tactical layer, crew dispatching, equipment deployment, overtime decisions, and intra-day operating windows are determined by the project team through established site-management procedures. The strategic schedule identifies where carbon-saving opportunities exist, especially activities with sufficient float that can be assigned to lower-emission operating profiles. However, actual execution must account for real-time field conditions, including weather, equipment availability, crew composition, safety lighting, noise limits, local construction regulations, and coordination with subcontractors. Therefore, valley-window or low-carbon operating recommendations should be interpreted as conditional opportunities rather than mandatory nighttime operations.

Weather, seasonal effects, and other site-specific uncertainties can be incorporated through parameter updates rather than by changing the model structure. For example, adverse weather or seasonal productivity loss can widen the duration intervals, reduce resource availability in affected periods, or modify the effective emission factors if machinery

efficiency changes under extreme temperatures. Local restrictions such as nighttime construction bans can likewise be represented by setting the corresponding resource availability to zero or excluding specific activities from valley-window operating profiles.

The framework is also compatible with rolling project-control workflows. When change orders, weather disruptions, resource shortages, or schedule deviations occur, the current project state can be used as a new baseline, completed activities can be fixed, and the remaining activities can be re-optimized with updated duration intervals, resource calendars, emission factors, and carbon-budget residuals. Thus, KD-IMOEA is not intended to replace field coordination or daily dispatching decisions; rather, it serves as a decision-support tool that helps managers identify carbon-efficient scheduling opportunities and update the master schedule as project conditions evolve.

6.3. Tactical Compliance and Regulatory Boundaries

The validity of the carbon savings reported in the case study depends on the tactical feasibility of valley-profile mechanical deployment. Local noise ordinances, evening-shift labor regulations, safety-lighting requirements, and project-specific constraints may preclude valley-window deployment for some or all energy-intensive resources. The framework accommodates such constraints through the time-varying resource capacity $R_{k\tau}$: for working days during which valley-window deployment of resource k is prohibited by local regulation, $R_{k\tau}$ is set to zero, automatically excluding those days from the algorithm's feasible solution space. The algorithm then rebalances toward alternative carbon-saving paths—shifting flexible activities to other low-EF operating profiles or reallocating activities to crew configurations whose emission profiles permit valley-profile deployment.

We adopt this scope-limiting approach in preference to embedding explicit night-window constraints into the core mathematical model for three reasons: (i) the existing time-varying capacity mechanism (Equation (15)) already provides the necessary degree of freedom; (ii) introducing explicit binary intra-day selection variables would substantially inflate the decision space without changing the dual-objective trade-off structure that KD-IMOEA optimizes; and (iii) regulatory contexts vary widely across jurisdictions, and embedding specific rules into the core model would harm portability. The strategic-tactical separation lets local regulatory rules be applied at the resource-capacity calibration stage, preserving the model's generality.

6.4. Resource Modeling Simplifications

The model treats resource demands r_{jk} as fixed activity-specific constants and assumes non-preemptive execution. Real construction practice exhibits more complex behaviors: equipment productivity varies with weather and operator skill; crew sizes are routinely adjusted to accelerate or absorb delays; overtime work compresses durations at the cost of higher unit emissions; and weather-sensitive activities may be paused mid-execution. Three considerations bound the impact of these simplifications.

First, the interval duration $\left[d_j^-, d_j^+\right]$ partially absorbs productivity variation. The optimistic bound d_j^- corresponds to favorable conditions (skilled crew, cooperative weather, full equipment availability); the pessimistic bound d_j^+ corresponds to adverse conditions. The model produces schedules robust across this range without explicitly modeling the underlying drivers.

Second, the strategic-tactical decomposition (Section 6.2) confines the model's responsibility to commitment-grade schedules. Tactical adjustments—overtime authorization, crew resizing, equipment reallocation—are made outside the optimizer based on real-time conditions, with the model invoked only for major re-planning events.

Third, beyond these internal absorption mechanisms, the framework targets construction-execution emissions and is structurally complementary to material-driven approaches addressing embodied carbon at the structural-design stage, such as CFRP-strengthened structural members [39] and FRP–steel composite tube columns confining seawater–sea-sand concrete [40]. The fixed-resource simplifications therefore do not preclude integration with such material innovations; the two pathways operate on different carbon-reduction layers and remain jointly optimizable in lifecycle-aware extensions.

7. Conclusions

We address the challenge of managing duration uncertainty and time-varying carbon constraints in green building scheduling by formulating a bi-objective interval RCPSP under carbon constraints and proposing KD-IMOEA. The model minimizes both interval makespan and interval total emissions, linking carbon budget to interval emissions through possibility-based constraints and applying interval Pareto dominance that constrains both optimistic and pessimistic objective bounds, providing a rigorous framework for green scheduling under uncertainty.

KD-IMOEA integrates four carbon-aware components into the NSGA-II framework: KDI, TCD, CBM, and IPA, forming an end-to-end optimization pipeline from population generation to solution output. Benchmark experiments on three problem scales show KD-IMOEA outperforms four classical algorithms in convergence, coverage, and distribution uniformity, with HV gains up to 6.3% and all 24 statistical tests reaching significance. Convergence analysis reveals high-start, fast-convergence dynamics, validating the synergy between knowledge-driven initialization and carbon-aware decoding. Ablation studies under tight constraints confirm that all four components contribute positively, with TCD being most critical. The green building case study validates practical applicability: KD-IMOEA produces a well-spread set of non-dominated solutions delineating the nonlinear makespan-emission tradeoff, and cuts emissions by 3.07% at the same optimal compromise makespan of 169.5 days compared to NSGA-II. The underlying mechanism—algorithmic identification of float-bearing energy-intensive activities at the strategic layer, coupled with valley-profile deployment at the tactical layer—achieves management-driven savings without additional equipment, offering budget-constrained contractors a practical path to balancing schedule targets and green certification.

Several limitations bound the present study. First, the model treats carbon emission factors as known and deterministic; in practice, $EF_{k\tau}$ exhibits stochastic variability driven by real-time grid mix fluctuations, weather-induced renewable output variation, and equipment-specific operating conditions. Second, resource demands r_{jk} are treated as fixed activity-specific constants, abstracting away productivity changes induced by crew configuration, overtime, or weather; the framework partially absorbs this through the interval duration $\left[d_j^-, d_j^+\right]$ but does not couple resource allocation with duration as a joint decision. Third, the intra-day grid-emission ratios (1.2:1.0:0.7) are calibrated against representative grid mix patterns under nominal conditions; finer-grained grid-emission profiles—either higher temporal resolution within a day or spatial differentiation across regional grids—would yield more accurate $EF_{k\tau}$ matrices. Fourth, the case study uses standard-derived parameters rather than real-site monitoring data, leaving the day-level emission factor calibration introduced in Section 5.1 unverified against actual construction monitoring data.

Promising directions for future research include (i) modeling emission factors as fuzzy or random variables to enable dual-uncertainty scheduling under both duration and emission stochasticity; (ii) coupling resource flexibility with carbon-aware scheduling by allowing r_{jk} to be a decision variable, with associated trade-offs between duration and unit

emission—the interval Pareto framework developed here generalizes naturally to such an extension; (iii) extending the KD-IMOEA carbon-aware mechanisms from the bi-objective formulation to three-objective or higher problems involving cost and quality dimensions; (iv) integrating real-time grid-emission data feeds to enable adaptive re-planning that responds to grid mix fluctuations within the construction horizon; and (v) field validation on instrumented construction sites to calibrate the day-level emission factors and quantify the realized vs. potential carbon savings under varying regulatory and operational contexts.

Author Contributions: Conceptualization, Y.D.; methodology, Y.D., Z.F. and C.L.; software, Y.D., Z.F. and W.T.; validation, Y.D. and H.C.; formal analysis, Y.D., Z.F. and C.L.; investigation, Y.D. and Z.F.; resources, Y.D. and W.T.; data curation, Y.D., C.L. and H.C.; writing—original draft preparation, Y.D., Z.F., W.T., Q.M., C.L. and H.C.; writing—review and editing, Y.D., Z.F., W.T., Q.M., C.L. and H.C.; visualization, Y.D.; supervision, C.L.; project administration, Y.D., Q.M. and C.L.; funding acquisition, Y.D. and H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This study was funded by the National Social Science Foundation of China (Grant No. 23XGL016).

Data Availability Statement: The benchmark instance files, PSPLIB IDs, emission-factor seeds, and algorithm random seeds supporting the findings of this study are available from the corresponding author upon reasonable request.

Acknowledgments: We would like to thank the esteemed reviewers for their constructive comments and valuable suggestions, which have significantly enhanced the quality of this manuscript.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Hamilton, I.; Kennard, H.; Rapf, O.; Amorocho, J.; Steuwer, S.; Kockat, J.; Toth, Z. *2023 Global Status Report for Buildings and Construction: Beyond Foundations—Mainstreaming Sustainable Solutions to Cut Emissions from the Buildings Sector*; United Nations Environment Programme: Nairobi, Kenya, 2024.
2. Huang, Z.; Zhou, H.; Miao, Z.; Tang, H.; Lin, B.; Zhuang, W. Life-cycle carbon emissions (LCCE) of buildings: Implications, calculations, and reductions. *Engineering* **2024**, *35*, 115–139. [CrossRef]
3. *GB/T 50378-2019*; Assessment Standard for Green Building. China Construction Industry Press: Beijing, China, 2019.
4. *GB/T 51366-2019*; Standard for Building Carbon Emission Calculation. China Construction Industry Press: Beijing, China, 2019.
5. Sun, Z.; Han, T.; Wang, X.; Wang, L.; Fu, H.; Li, Y.; Zhong, Z.; Liu, J.; Huang, H.; Wu, Z. Bidirectional mapping modeling of pipeline vertical deformation and axial strain based on multi-source monitoring data and machine learning. *J. Pipeline Sci. Eng.* **2026**, 100432. [CrossRef]
6. Yao, Z.; Li, Y.; Fu, H.; Tian, J.; Zhou, Y.; Chin, C.-L.; Ma, C.-K. Research on concrete crack and depression detection method based on multi-level defect fusion segmentation network. *Buildings* **2025**, *15*, 1657. [CrossRef]
7. Wu, X.; He, J.; Tian, J.; Liu, M.; Zheng, Y.; Zhang, W.; Yuan, J.; Wang, W.-W.; Fu, H.; Wang, R. Multi-scale performance of underwater-castable ultra-high-volume fly-ash eco-friendly engineered cementitious composites: Microstructural characterization, workability, mechanical properties and environmental effects. *Constr. Build. Mater.* **2026**, *506*, 144870. [CrossRef]
8. Wu, X.; Ouyang, X.; Tian, J.; Zheng, Y.; Yuan, J.; Wang, W.-W.; Zuo, Y.; Fu, H.; Yu, W.; Wang, R. Self-sensing mechanism and theoretical model of prestressed FRP-ECC multifunctional composite beams under bending loads. *Structures* **2026**, *87*, 111585. [CrossRef]
9. Fu, H.; Tian, J.; Chin, C.-L.; Liu, H.; Yuan, J.; Tang, S.; Mai, R.; Wu, X. Axial compression behavior of GFRP-steel composite tube confined seawater sea-sand concrete intermediate long columns. *Eng. Struct.* **2025**, *333*, 120157. [CrossRef]
10. Fu, H.; Tian, J.; Chen, S.; Chin, C.-L.; Ma, C.-K. Axial compressive performance of CFRP-steel composite tube confined seawater sea-sand concrete intermediate slender columns. *Constr. Build. Mater.* **2024**, *441*, 137399. [CrossRef]
11. Flórez, L.Á.; Péan, T.; Salom, J. Hourly based methods to assess carbon footprint flexibility and primary energy use in decarbonized buildings. *Energy Build.* **2023**, *294*, 113213. [CrossRef]
12. Hartmann, S.; Briskorn, D. An updated survey of variants and extensions of the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **2022**, *297*, 1–14. [CrossRef]

13. Deb, K.; Pratap, A.; Agarwal, S.; Meyarivan, T. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Trans. Evol. Comput.* **2002**, *6*, 182–197. [CrossRef]
14. Deb, K.; Jain, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part I: Solving problems with box constraints. *IEEE Trans. Evol. Comput.* **2013**, *18*, 577–601. [CrossRef]
15. Zhang, Q.; Li, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. *IEEE Trans. Evol. Comput.* **2007**, *11*, 712–731. [CrossRef]
16. Zitzler, E.; Laumanns, M.; Thiele, L. SPEA2: Improving the strength Pareto evolutionary algorithm. *Tech. Rep.* **2001**, *103*, 259–271.
17. Pellerin, R.; Perrier, N.; Berthaut, F. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. *Eur. J. Oper. Res.* **2020**, *280*, 395–416. [CrossRef]
18. Aramesh, S.; Mousavi, S.M.; Ghasemi, M.; Shahabi-Shahmiri, R. An optimization model for construction project scheduling by considering CO_2 emissions with multi-mode resource constraints under interval-valued fuzzy uncertainty. *Int. J. Eng. Sci. Technol.* **2023**, *20*, 87–102. [CrossRef]
19. Razi, N.; Ansari, R. A prediction-based model to optimize construction programs: Considering time, cost, energy consumption, and CO_2 emissions trade-off. *J. Clean. Prod.* **2024**, *445*, 141164. [CrossRef]
20. Rahman, M.; Chowdhury, R.I.; Elkabalawy, M.; Wasiew, Q.A.; Moon, S.; Han, S.H. Optimizing production scheduling for decarbonization in off-site construction. In Proceedings of the Joint CSCE Construction Specialty & CRC Conference 2025, Montreal, QC, Canada, 28–31 July 2025.
21. Ding, H.; Zhuang, C.; Liu, J. Extensions of the resource-constrained project scheduling problem. *Autom. Constr.* **2023**, *153*, 104958. [CrossRef]
22. Fu, F.; Liu, Q.; Yu, G. Robustifying the resource-constrained project scheduling against uncertain durations. *Expert Syst. Appl.* **2024**, *238*, 122002. [CrossRef]
23. Patoghi, A.; Mousavi, S.M. A new approach for material ordering and multi-mode resource constraint project scheduling problem in a multi-site context under interval-valued fuzzy uncertainty. *Technol. Forecast. Soc. Change* **2021**, *173*, 121137. [CrossRef]
24. Hazır, Ö.; Ulusoy, G. A classification and review of approaches and methods for modeling uncertainty in projects. *Int. J. Prod. Econ.* **2020**, *223*, 107522. [CrossRef]
25. Ghasemi, M.; Mousavi, S.M.; Aramesh, S.; Shahabi-Shahmiri, R.; Zavadskas, E.K.; Antucheviciene, J. A new approach for production project scheduling with time-cost-quality trade-off considering multi-mode resource-constraints under interval uncertainty. *Int. J. Prod. Res.* **2023**, *61*, 2963–2985. [CrossRef]
26. Hatami-Moghaddam, L.; Khalilzadeh, M.; Shahsavari-Pour, N.; Sajadi, S.M. Developing a robust multi-skill, multi-mode resource-constrained project scheduling model with partial preemption, resource leveling, and time windows. *Mathematics* **2024**, *12*, 3129. [CrossRef]
27. Li, H.; Zhu, H.; Zheng, L.; Xie, F. Software project scheduling under activity duration uncertainty. *Ann. Oper. Res.* **2024**, *338*, 477–512. [CrossRef]
28. Aghileh, M.; Tereso, A.; Alvelos, F.; Monteiro Lopes, M.O. Multi-project scheduling under uncertainty and resource flexibility: A systematic literature review. *Int. J. Prod. Res.* **2024**, *12*, 2319574. [CrossRef]
29. Yu, F.; Lu, C.; Zhou, J.; Yin, L.; Wang, K. A knowledge-guided bi-population evolutionary algorithm for energy-efficient scheduling of distributed flexible job shop problem. *Eng. Appl. Artif. Intell.* **2024**, *128*, 107458. [CrossRef]
30. Kolisch, R.; Hartmann, S. Experimental investigation of heuristics for resource-constrained project scheduling: An update. *Eur. J. Oper. Res.* **2006**, *174*, 23–37. [CrossRef]
31. Kolisch, R. Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *Eur. J. Oper. Res.* **1996**, *90*, 320–333. [CrossRef]
32. Li, D.; Wang, L.; Li, L.; Jiang, C.; Li, Y. A large-scale multiobjective particle swarm optimizer with enhanced balance of convergence and diversity. *IEEE Trans. Cybern.* **2024**, *54*, 1596–1607. [CrossRef]
33. Mitra, A. The Taguchi method. *Wiley Interdiscip. Rev. Comput. Stat.* **2011**, *3*, 472–480. [CrossRef]
34. Trentini, E.V.W.; Parsekian, G.A.; Bittencourt, T.N. Multiobjective optimization of bridge and viaduct design: Comparative study of metaheuristics and parameter calibration. *Eng. Struct.* **2024**, *312*, 118252. [CrossRef]
35. While, L.; Bradstreet, L.; Barone, L. A fast way of calculating exact hypervolumes. *IEEE Trans. Evol. Comput.* **2011**, *16*, 86–95. [CrossRef]
36. Ibrahim, A.; Bidgoli, A.A.; Rahnamayan, S.; Deb, K. A novel Pareto-optimal ranking method for comparing multi-objective optimization algorithms. *arXiv* **2024**, arXiv:2411.17999.
37. Kolisch, R.; Sprecher, A. PSPLIB—A project scheduling problem library. *Eur. J. Oper. Res.* **1997**, *96*, 205–216. [CrossRef]
38. Tanabe, R.; Ishibuchi, H. A review of evolutionary multimodal multiobjective optimization. *IEEE Trans. Evol. Comput.* **2019**, *24*, 193–200. [CrossRef]

39. Liu, M.; Tian, J.; Zhao, J.; Sun, Y.; Liang, J.; Zheng, Y.; Zhang, W.; Wu, X.; Yuan, J.; Fu, H. Seismic performance of CFRP-strengthened steel fiber reinforced concrete composite columns: Experimental, numerical and theoretical analyses. *Constr. Build. Mater.* **2025**, *505*, 144692. [CrossRef]
40. Jiang, S.; Wu, L.; Chen, C.; Tian, J.; Ling, C.; Mai, R.; Fu, H.; Lyu, P.; Cui, H. FRP–steel composite tube confined seawater–sea-sand concrete columns: State-of-the-art review. *Buildings* **2026**, *16*, 1351. [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.