Metaheuristic Algorithms with Applications

www.maa.reapress.com

Metaheur. Algor. Appl. Vol. 2, No. 1 (2025) 63–81.

Paper Type: Original Article

Optimization of Construction Resource Allocation and Levelling Using the World Hyper-Heuristic Algorithm

Zahra Joorbonyan[1,*] Marzieh Lamtar-Gholipoor[2]

[1] Morvarid Intelligent Industrial Systems Research Group, Iran; Zahrajbn@gmail.com.
[2] Department of Chemical Engineering, Quchan Branch, Islamic Azad University, Quchan, Iran; marzieh.gholipoor@gmail.com.

Citation:

Received: 21 November 2024 *Revised: 09 January 2025* *Accepted: 02 February 2025*	Joorbonyan, Z., & Lamtar-Gholipoor, M. (2025). Optimization of construction resource allocation and levelling using the world hyper-heuristic algorithm. *Metaheuristic algorithms with applications, 2*(1), 63-81.

Abstract

Resource allocation and levelling remain critical challenges in construction management, where project tasks must be synchronized under strict time and resource constraints. Traditional scheduling techniques, such as the Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), efficiently define timelines but fail to handle fluctuations in resource usage. These fluctuations cause inefficiencies, idle periods, and cost overruns. This paper introduces the World Hyper-Heuristic (WHH) algorithm, an adaptive optimization framework that dynamically selects among multiple metaheuristics using reinforcement learning. WHH integrates Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), and Imperialist Competitive Algorithm (ICA) into a unified decision-making model. The proposed method is implemented in MATLAB and tested on a real construction project dataset involving 20 activities. Experimental results demonstrate that WHH achieves superior performance in minimizing resource fluctuation and improving levelling smoothness. Compared to GA, PSO, ACO, SA, and ICA, the proposed algorithm exhibits faster convergence, higher stability, and better adaptability to constraint complexity. These findings suggest that WHH provides a practical, intelligent framework for optimizing resource allocation and levelling in modern construction projects.

Keywords: Resource allocation, Resource levelling, World Hyper-Heuristic, Reinforcement learning, Metaheuristics, MATLAB, Construction optimization.

1 | Introduction

Efficient scheduling and resource management lie at the heart of every successful construction project. Regardless of scale, a construction project is an intricate interplay of interdependent activities, human resources, machinery, materials, and time. Aligning these components under strict constraints is a balancing act that determines whether a project is delivered on time and within budget, or spirals into delays and cost

Corresponding Author: Zahrajbn@gmail.com
https://doi.org/10.48313/maa.v2i1.28

overruns. Traditional approaches to project scheduling, most notably the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT), have long provided structured methods for defining dependencies and estimating project durations. However, while these methods establish feasible schedules, they are ill-equipped to handle the dynamic variability of resource demands that occurs throughout a project's lifecycle [1].

In practice, construction resource utilization rarely follows a smooth curve. Instead, resource usage fluctuates dramatically due to concurrent activities, varying work rates, weather conditions, and unforeseen site challenges. These fluctuations manifest as peak periods of resource congestion, followed by idle intervals in which workforce and equipment remain underused. Such an imbalance not only increases operational costs but also reduces productivity and overall project efficiency. A schedule that is optimal in terms of time can still be inefficient if its resource profile is unstable. Resource levelling, smoothing out resource consumption while maintaining project deadlines, has therefore become an essential optimization objective alongside time and cost minimization [2].

Traditional deterministic methods can optimize project duration but fail to balance the dynamic nature of real-world resource usage. The challenge arises because resource allocation and levelling constitute a nonlinear, multi-objective, and combinatorial optimization problem. Finding the best sequence of activities and corresponding resource assignments involves an exponentially growing search space that quickly exceeds the capability of exact mathematical programming or heuristic priority-rule methods, especially as project complexity increases. As a result, researchers and engineers have increasingly turned to metaheuristic optimization, a family of algorithms inspired by natural and social processes that can efficiently explore vast, complex solution spaces [3].

Metaheuristics such as the Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), Simulated Annealing (SA), and the Imperialist Competitive Algorithm (ICA) have each demonstrated value in solving construction scheduling and resource levelling problems. These algorithms emulate principles of evolution, collective intelligence, physical annealing, or socio-political competition to search for near-optimal solutions. They excel in exploring nonlinear and multi-modal spaces without requiring gradient information or convex assumptions. In construction management, GA has been widely applied to optimize project duration and smooth resource distribution through evolutionary crossover and mutation. PSO, inspired by swarm intelligence, efficiently balances global and local search dynamics, making it useful for fine-tuning schedules under uncertain conditions. ACO adapts collective path-finding behavior to sequencing tasks and identifying optimal precedence relationships. SA mimics the gradual cooling of metals to escape local minima by probabilistically accepting sub-optimal solutions. At the same time, ICA exploits imperialistic competition and assimilation mechanisms to guide populations toward global optima [4].

While each of these algorithms is powerful, they also exhibit inherent weaknesses. GA can lose diversity and converge prematurely. PSO often sacrifices global exploration in exchange for faster convergence. ACO's performance is susceptible to pheromone parameters and tends to degrade in high-dimensional search spaces. SA's stochastic acceptance mechanism can lead to slow convergence, and ICA, though globally oriented, requires heavy computation to maintain multiple empires and colonies. All share one limitation: they employ fixed search strategies. Once initialized, their behavior is determined by static parameter settings, regardless of how the optimization landscape evolves. In dynamic, constraint-rich environments such as construction scheduling, where the problem's nature may change as constraints interact, static search mechanisms can become inefficient, unbalanced, or trapped in local optima [2].

This limitation has prompted a new wave of research that goes beyond traditional metaheuristics: hyper-heuristic optimization. A hyper-heuristic operates at a higher level of abstraction than a metaheuristic. Instead of being a single algorithm that directly searches the solution space, a hyper-heuristic manages and coordinates a collection of metaheuristics or low-level heuristics. It learns which algorithm performs best under specific circumstances and dynamically switches among them during the optimization process. Essentially, hyper-heuristics shift the question from “how can we find a good solution?” to “which algorithm should we use right now to find a good solution?” [1], [2].

This conceptual shift introduces a new degree of adaptability. By monitoring the performance of constituent algorithms in real time, a hyper-heuristic can allocate computational effort intelligently, favoring exploration when diversity is needed and switching to exploitation when convergence is near. It can also mitigate the weaknesses of individual metaheuristics by combining their complementary strengths. For example, PSO and ACO are strong in exploration but may lack fine-grained exploitation; GA and SA, on the other hand, excel at refining solutions once promising regions have been identified. A hyper-heuristic that can blend these behaviors dynamically can outperform any single algorithm operating in isolation [5].

Building upon this principle, this study introduces the World Hyper-Heuristic (WHH) algorithm, an adaptive optimization framework that integrates multiple metaheuristics under a Reinforcement Learning (RL)-based control mechanism. In WHH, a reinforcement learning agent continuously observes the performance of each underlying metaheuristic and learns a policy that maximizes long-term optimization rewards. The agent dynamically selects the following algorithm to apply based on historical improvement trends, convergence rate, and diversity indicators. This approach transforms the optimization process into a self-adaptive system that balances exploration and exploitation at every iteration [6].

The term “world” in WHH signifies the algorithm’s holistic perspective: it treats each metaheuristic as a nation or entity within a global ecosystem of search strategies. The reinforcement learning controller acts as a global governor evaluating the progress of each entity, rewarding those that contribute effectively to solution improvement, and discouraging unproductive behaviors. In this way, WHH orchestrates a cooperative yet competitive environment among algorithms, analogous to natural or social systems where different agents with unique strategies coexist and evolve collectively [7].

The key advantage of WHH lies in its adaptivity and self-organization. Unlike static hybrid models that combine metaheuristics in fixed proportions or sequences, WHH makes decisions based on performance feedback. It can allocate more iterations to PSO during the early stages of exploration, then gradually shift toward GA or SA for refinement as convergence nears. If stagnation is detected, it can reintroduce exploratory behavior by switching to ACO or ICA. This flexible control mechanism allows WHH to maintain search diversity while accelerating convergence, solving one of the long-standing dilemmas in evolutionary computation [8].

In construction resource allocation and levelling, this adaptability is crucial. The optimization landscape is highly irregular, characterized by numerous local minima arising from precedence constraints, discrete resource units, and time-window restrictions. Static algorithms often require extensive parameter tuning for each project instance. WHH, by contrast, learns an internal policy that generalizes across different project scales and constraint sets, reducing dependency on manual parameter adjustment [1].

Furthermore, construction projects rarely involve a single optimization objective. Decision-makers must simultaneously minimize project duration, reduce resource fluctuations, and balance utilization over time. These conflicting objectives make the problem inherently multi-objective and require trade-offs that can shift depending on managerial priorities. The WHH framework accommodates such multi-objective structures by incorporating weighted objective functions and adaptive learning that responds to changing dominance relations among objectives during the search process [9].

Beyond technical superiority, WHH carries practical implications. Modern construction management increasingly relies on digital tools such as Building Information Modeling (BIM), digital twins, and real-time data from sensors and Internet of Things (IoT) devices. These systems generate dynamic data streams that can alter project parameters on the fly, such as weather forecasts, delivery delays, workforce availability, or unexpected design revisions. An optimization framework that can learn and adapt in real time, rather than restarting the optimization whenever input data changes, is indispensable in this context. WHH provides the foundation for such adaptability, as its reinforcement learning agent can continuously refine strategy selection as new data arrive, enabling dynamic rescheduling and predictive control [10].

From a theoretical standpoint, WHH contributes to the ongoing convergence between machine learning and metaheuristic optimization. Traditional metaheuristics rely primarily on stochastic operators and population dynamics, whereas machine learning introduces statistical generalization and experience-based decision-making. The reinforcement learning component in WHH bridges these paradigms by allowing the optimization process to learn from its own history, essentially turning the search procedure into a self-improving system. The resulting synergy produces a form of computational intelligence that evolves beyond predefined search behavior.

The proposed WHH algorithm thus serves as both an optimization framework and a research concept that unites heuristic reasoning, adaptive learning, and multi-agent cooperation. Its structure can be summarized in three levels:

I. Low-level metaheuristics (GA, PSO, ACO, SA, and ICA) serve as specialized search operators, each offering distinct exploration or exploitation capabilities.

II. Reinforcement learning controller: evaluates the immediate and cumulative reward of each metaheuristic based on solution improvement and convergence criteria.

III. Adaptive decision policy: determines which metaheuristic to deploy at each iteration via probabilistic selection, dynamically balancing exploration and exploitation.

Through this architecture, WHH converts the optimization process into a living system that learns how to optimize more effectively over time. The cooperation among metaheuristics is not pre-programmed but emerges from the reinforcement learning mechanism's decisions, leading to an inherently flexible and generalizable optimization strategy.

The objectives of this research are threefold:

I. First, to develop a novel WHH framework that integrates multiple metaheuristics within a reinforcement learning environment, tailored explicitly for joint resource allocation and levelling in construction scheduling.

II. Second, to perform a comparative analysis of WHH against established metaheuristics such as GA, PSO, ACO, SA, and ICA, evaluating their performance in terms of convergence speed, solution quality, and robustness under identical conditions.

III. Third, to validate WHH's effectiveness on real construction project data, demonstrating that the proposed algorithm not only surpasses traditional metaheuristics in computational performance but also produces more stable and practical scheduling outcomes.

By achieving these objectives, the study aims to establish WHH as a comprehensive decision-support framework for construction resource management, one that goes beyond algorithmic performance to address the real operational complexities faced by project managers. Its ability to smooth resource utilization, minimize idle periods, and adapt dynamically to changing constraints provides a tangible advantage for modern construction projects operating under uncertainty [11].

Ultimately, this research bridges the gap between classical project scheduling theory and adaptive computational intelligence. It respects the legacy of CPM and PERT while embracing the evolution of heuristic learning. The WHH algorithm embodies a philosophy that unites structure with flexibility: structure in its systematic integration of metaheuristics, and flexibility in its capacity to evolve its own behavior as the optimization unfolds. In doing so, it contributes to the broader vision of intelligent construction management systems that learn, adapt, and optimize continuously, just as human planners do, but with the speed and precision of modern computation.

Accordingly, the remainder of this paper is organized as follows. Section 2 reviews previous studies on deterministic, heuristic, metaheuristic, and emerging hyper-heuristic approaches in construction resource scheduling, highlighting the evolution of methods and existing research gaps. Section 3 formulates the Resource Allocation and Levelling Problem (RALP) and defines the multi-objective functions and constraints used in this study. Section 4 introduces the proposed WHH algorithm, describing its reinforcement learning mechanism, hierarchical structure, and workflow integrating multiple metaheuristics (GA, PSO, ACO, SA, and ICA). Section 5 presents the experimental setup and comparative analysis, evaluating WHH against benchmark algorithms in terms of convergence, stability, and resource-levelling performance. Section 6 concludes the paper with a discussion of WHH's practical implications, adaptability, and potential directions for future research in adaptive optimization for construction management.

2 | Related Work

Optimization of resource scheduling in construction has undergone several evolutions — from classical deterministic methods to heuristics and metaheuristics, to the more recent adaptive and hyper-heuristic approaches. Understanding this progression is vital for positioning the proposed WHH algorithm and recognizing both the strengths and gaps in the literature [2], [7], [10].

2.1 | Deterministic and Heuristic Beginnings

In the earliest decades of construction scheduling research, the emphasis was on methods such as the CPM, PERT, and linear/integer programming formulations. These approaches were valuable because they provided clear structure and transparency: given fixed precedence constraints and known durations, one could compute earliest/latest start times, slack, float, and total project duration. However, they were predicated on strong assumptions about resource availability and typically ignored or simplified resource constraints (especially when resources are shared, multi-mode, or variable over time). As the network size, number of tasks, and resource types grew, these deterministic methods began to struggle. In large construction networks, the computational burden of exact integer programming models became prohibitive; heuristics based on priority rules (e.g., longest processing time first, earliest finish time) offered faster solutions but often provided sub-optimal smoothing of resource usage.

In the context of resource levelling, specifically the goal of smoothing resource demand over time rather than just minimising makespan, heuristic approaches began to emerge. For example, resource-driven scheduling methods in repetitive construction (such as high-rise or pipeline work) were introduced to reduce idle time and maximise the continuity of crew or equipment deployment. These methods emphasised discipline in resource flow (keeping crews busy, avoiding resource peaks and troughs), but often relied on domain-specific rules rather than general optimisation frameworks [3].

2.2 | Rise of Metaheuristics

As computational power increased and scheduling models became more complex (e.g., multi-mode tasks, renewable and non-renewable resources, stochastic durations), researchers shifted to metaheuristic approaches. These techniques, inspired by natural processes (evolution, swarm intelligence, annealing) or competitive models, are ideally suited to large combinatorial-search spaces, nondifferentiable objective functions, and multi-objective trade-offs.

For instance, a significant line of work developed in the mid-2010s explored metaheuristic methods for resource-leveling large-scale construction projects. One doctoral thesis was dedicated to the design of efficient heuristic and meta-heuristic methods for the Resource-Levelling Problem (RLP) and the Discrete Time-Cost Trade-Off Problems (DTCTP) in large construction scheduling contexts. It showed that hybrid methods (e.g., memetic algorithms combined with simulated annealing, or quasi-stable, schedule-focused GAs) outperformed simpler heuristic rules, particularly on significant problems that typical software tools struggled with.

Another study applied a GA specifically to resource levelling in construction projects, accounting for daily resource usage and aiming to minimise variability (the sum of squared deviations from a desired level). Results from sample projects indicated that GA could significantly reduce resource peaks and smooth the temporal distribution of workforce or equipment use compared to default scheduling tools [12].

Metaheuristic studies also addressed multi-objective trade-offs in construction scheduling. For example, construction managers care not only about duration (makespan) but also cost, quality, resource utilization, and risk. An article considered the time-cost-quality trade-off in a bridge-project context, modelling both renewable and non-renewable resources and applying three metaheuristic algorithms (multi-objective grey wolf optimizer, NSGA-II, multi-objective PSO) to evaluate performance. That work underscored the flexibility of metaheuristics when traditional deterministic methods became intractable or overly restrictive [13].

In addition to model complexity, scalability became a key concern. One study on multi-project scheduling in construction (multiple overlapping projects sharing resources) introduced hybrid metaheuristics combining GA, simulated annealing, and backward-forward improvement heuristics, addressing the challenge of resource conflicts across projects (something classical CPM could not handle).

2.3 | Focus on Resource Levelling and Multi-Objective Scheduling

While makespan minimisation remains a core focus, many researchers recognised that for construction projects, resource levelling (i.e., smoothing peaks and troughs in resource usage) is equally vital for cost control, productivity, and stability. One study explored the RLP with a fixed project duration and the minimisation of resource usage fluctuations. The authors proposed greedy algorithms and hybrid low-level heuristics embedded within population-based metaheuristics (differential evolution, GA, PSO) and demonstrated that hybridisation yields smoother resource profiles [3].

Work on repetitive construction projects (e.g., multi-unit buildings, linear infrastructure) also emphasised the unique nature of resource allocation there: crews often move from unit to unit, resource continuity is critical, and variability can lead to high costs or delays. A study on multi-objective resource-constrained scheduling for large repetitive projects applied differential evolution to a project comprising 160 activities and 16 repetitive sub-projects, demonstrating that metaheuristics can scale beyond toy problems and approximate real-world complexity while addressing conflicting objectives (duration vs. usage stability) [8].

These developments illustrate that metaheuristics moved the field from "just finish as soon as possible" to "finish as soon as possible while using resources smoothly, cost-effectively, and with less waste/idle time." However, most approaches still rely on selecting a single metaheuristic (or a fixed hybrid) and tailoring it to problem instances.

2.4 | The Advent of Hyper-Heuristics and Learning-Based Control

More recently, the field of scheduling and resource allocation has begun to embrace hyper-heuristics, which represent a higher-level shift: rather than using a single algorithm, the idea is to manage, select, generate, or adapt multiple heuristics dynamically. A hyper-heuristic framework consists of a high-level controller and a pool of low-level heuristics/metaheuristics; the controller learns (or decides) which low-level algorithm to apply under which circumstances, ideally adapting to instance features or search progress.

In other domains (manufacturing, flow shop scheduling, cloud computing, vehicle routing), hyper-heuristic approaches have shown strong promise. For example, a Q-learning-based hyper-heuristic was used to schedule tasks in a distributed flow shop environment with energy-efficiency objectives. The RL agent, selected based on makespan and energy consumption, exhibited superior convergence and robustness compared with fixed-heuristic selection strategies. Another study proposed a MAP-Elites–based hyper-heuristic for the Resource-Constrained Project Scheduling Problem (RCPSP) that maintained an archive of diverse heuristic behaviours and selected them based on performance diversity across instances [4], [5].

In the construction scheduling context, the adoption of hyper-heuristics is still relatively limited but emerging. Some work on construction resource-levelling has referenced hyper-heuristic frameworks, but predominantly in the generic scheduling literature rather than in construction-specific case studies. The need for dynamic algorithm selection is especially acute in construction because projects differ widely in scale, resource types, precedence complexity, merge-split relationships, and duration uncertainty. A framework that can learn which heuristic strategy to apply and when, rather than relying on manual tuning, therefore holds great appeal [11].

2.5 | Gaps, Opportunities, and Positioning of the WHH

Despite these advances, several key gaps remain in the literature and, hence, the opportunities your proposed WHH algorithm addresses.

First, while many metaheuristic studies target makespan or cost, fewer explicitly focus on resource levelling (smooth resource utilisation) as a structured objective alongside duration/cost. Those that do often assume a fixed algorithm or per-instance tuning, rather than a framework that adapts dynamically. Second, most past studies use a single heuristic/metaheuristic, or a fixed hybrid (e.g., GA combined with local search). What is lacking is a unified framework that embeds multiple heuristics and selects them adaptively at runtime. This is precisely the promise of hyper-heuristics, but construction-specific adoption remains minimal [8].

Third, many studies are limited to small-to-medium-sized problems (e.g., 30-100 activities) or assume relatively simple resource/environmental structures. Real construction projects often involve multiple resource types, precedence networks, multi-mode tasks, stochastic durations, and dynamic changes (material delays, workforce shifts). A generalized adaptive optimisation framework is therefore needed.

Fourth, parameter tuning remains a heavy burden. Metaheuristics often require careful adjustment of crossover/mutation rates, swarm velocities, pheromone evaporation rates, and related parameters. Static tuning may yield sub-optimal performance when problem characteristics shift. A control mechanism that learns which algorithm works when mitigates this burden.

Finally, monitoring convergence and deciding when to switch strategies is rarely addressed in depth. A hyper-heuristic that monitors performance indicators (diversity, improvement rate, stagnation) in real time and switches heuristics accordingly is an area of active research, but not yet widespread in construction resource scheduling [1].

The proposed WHH algorithm sits at this nexus of opportunity. It brings together multiple underlying metaheuristics (GA, PSO, ACO, SA, ICA) under a reinforcement learning-based controller that dynamically selects or blends the algorithms during the search. The WHH is designed to minimise not just project duration, but also explicit measures of resource fluctuation (e.g., moments of resource usage, distribution over time), and to adapt across different project instances without heavy retuning. In doing so, it addresses the fundamental shift from “one fixed algorithm per problem” to “a system of algorithms plus learning controller,” aligning with the hyper-heuristic vision but tailoring it to the construction resource allocation and levelling domain [2].

In short, the literature shows a clear evolution: deterministic scheduling → heuristics → metaheuristics → hyper-heuristics/learning-based frameworks. However, in the specific niche of construction resource allocation and levelling, the last step (fully adaptive hyper-heuristic frameworks) remains under-explored. The WHH fills that gap by offering a higher-level, adaptive, generalisable optimisation framework, grounded in the traditional metaheuristic toolkit and forward-looking in its learning control paradigm.

3 | Resource Allocation and Levelling Problem

In construction project management, resource allocation and levelling are among the most critical challenges for achieving efficient scheduling and cost control. Construction projects often involve multiple activities competing for limited resources, such as labor, equipment, and materials, while being bound by precedence constraints and strict deadlines. Imbalanced or poorly timed allocation of these resources can result in idle periods, excessive overtime, or costly delays.

Resource allocation assigns available resources to scheduled activities to minimize project duration or cost within specific constraints. Resource levelling, on the other hand, aims to smooth the fluctuations in resource demand over time without violating the project's precedence or duration limits. In practice, both objectives must be balanced simultaneously: a perfectly levelled schedule might extend the project duration. In contrast, an aggressively compressed schedule may produce resource peaks that are logistically and financially impractical [5].

As project complexity grows, manual scheduling or traditional deterministic optimization becomes insufficient. The high degree of interdependency among activities, stochastic resource availability, and the multi-objective nature of the problem require adaptive, intelligent approaches. Metaheuristic and hyper-heuristic algorithms have therefore emerged as robust alternatives, capable of navigating the vast combinatorial search space and discovering near-optimal trade-offs among duration, cost, and resource smoothness.

The RALP can be formulated as a multi-objective optimization problem, aiming to minimize project duration, resource fluctuation, and uneven utilization simultaneously. The following section presents the formal problem formulation and the objective functions used to quantify these goals [14].

3.1 | Problem Formulation

Construction project scheduling is fundamentally a balancing act between time, resources, and costs. Among these, resource allocation and levelling are among the most challenging optimization problems due to their nonlinear, discrete, and dynamic nature. Projects often involve multiple activities that compete for limited resources, such as labor, machinery, or materials, under strict precedence relationships and varying productivity rates. The goal is to determine how these resources should be assigned over time so that project objectives are met efficiently while maintaining stable resource usage throughout the project's life cycle.

In its most basic form, the RALP can be described as follows: given a set of project activities, each with a duration, resource demand, and precedence constraints, determine the start and finish times for each activity and allocate the available resources so that fluctuations in resource usage are minimized. In practice, this means keeping resource utilization as smooth as possible, avoiding sharp peaks (overloads) and valleys (idleness), while ensuring the project duration remains acceptable or minimal.

Formally, let there be $n$activities in a construction project. Each activity $i$requires a certain amount of resource r_i for a given duration d_i, subject to precedence relationships P_{ij} that defines the execution order between activities i and j. The total available resource at any time t is limited by a capacity R_{max}. The objective is not only to complete all activities within the project horizon, but also to distribute the resource load over time to avoid significant fluctuations.

To capture the dynamics of resource usage, let $R(t)$ denote the total amount of a specific resource used at time t. This is obtained by summing the demands of all activities active at that time. The average or mean resource utilization over the project's time horizon T can be defined as

$$\overline{R} = \frac{1}{T}\sum_{t=1}^{T} R(t).$$

This average serves as a benchmark for what constitutes "perfectly smooth" usage, though in real projects, such smoothness is rarely achievable due to task interdependencies and precedence constraints. The degree of unevenness or fluctuation in resource utilization can then be quantified through two principal indices. The first, M_x, represents the total variation or fluctuation around the average resource usage:

$$M_x = \sum_{t=1}^{T} (R(t) - \bar{R})^2.$$

This formulation is conceptually similar to variance; it penalizes deviations (both upward and downward) from the mean utilization level. A smaller value of M_x indicates smoother resource consumption, implying that resource demand over time is more consistent and predictable. Minimizing M_x helps managers avoid resource bottlenecks, reduce idle times, and maintain stable crew deployment, which ultimately leads to lower indirect costs and improved productivity. The second measure, M_y, captures the distribution of resource utilization over time. It is defined as

$$M_y = \sum_{t=1}^{T} [t.(R(t) - \bar{R})].$$

Unlike M_x, which focuses on the magnitude of fluctuations, M_y accounts for *when* these fluctuations occur. In effect, it measures the temporal balance of resource usage. A low value of M_y implies that the project maintains a relatively balanced distribution of resources across the early and late stages, avoiding front-loading (high usage at the beginning) or back-loading (peaks near the project end). This is particularly important for large-scale projects where resources must be shared across multiple sites or phases.

Together, M_x and M_y provide complementary perspectives on resource stability: the first measures smoothness, while the second assesses symmetry in utilization over time. Achieving optimal levelling means minimizing both indices concurrently, which naturally leads to a multi-objective optimization problem.

The comprehensive objective function for the RALP can thus be formulated as

$$\text{Minimize: } f = w_1.\text{Project Duration} + w_2.M_x + w_3.M_y,$$

where w_1, w_2, and w_3 are non-negative weight coefficients representing the relative importance assigned to each objective: duration, fluctuation, and temporal balance. The inclusion of weights allows flexible prioritization depending on project conditions. For instance, in a fast-track project where deadlines are critical, w_1 may dominate, while in projects with stable schedules but a limited workforce, w_2 and w_3 may be emphasized.

This multi-objective structure reflects real-world trade-offs faced by construction managers. Shorter durations typically require greater resource intensity, increasing fluctuations and costs. Conversely, achieving a perfectly levelled resource profile often implies extended project durations or lower productivity. The optimization challenge lies in balancing these competing objectives.

3.2 | Constraints and Decision Variables

The optimization problem is subject to several practical and technical constraints, which ensure the feasibility and realism of the resulting schedule.

Precedence constraints

Each activity must start only after all its predecessors are completed.

$$S_j \geq S_i + d_i; \quad for\ all\ (i,j) \in P,$$

where S_i and S_j denote start times of activities i and j.

Resource capacity constraints

The total resource consumption at any time cannot exceed the available limit:

$$R(t) = \sum_{i \in A(t)} r_i \leq \text{Rmax}, \qquad \text{for all t},$$

where $A(t)$ is the set of activities active at time t.

Non-negativity and discreteness

Start times S_i and resource assignments r_i are discrete and non-negative, reflecting real project timing and resource availability.

Project completion

The finish time of the final activity determines the project duration:

$$D = \max_i (Si + di).$$

Multi-mode activities (optional)

In more complex formulations, each activity may be executed in one of several modes, where each mode m has its own duration $d_{i,m}$ and resource requirement $r_{i,m}$. This introduces additional decision variables for mode selection:

$$x_{i,m} = \begin{cases} 1, & \text{if activity i is executed in mode m,} \\ 0, & \text{otherwise.} \end{cases}$$

These constraints, combined with the objective function, make the RALP a combinatorial, NP-hard optimization problem. Exact methods, such as linear or mixed-integer programming, rapidly become computationally infeasible as project size increases. Therefore, heuristic and metaheuristic approaches are widely adopted to search for near-optimal solutions in a reasonable time.

3.3 | Interpretation of Objectives and Trade-Offs

The multi-objective formulation captures several fundamental trade-offs:

Duration vs. fluctuation: reducing project duration often requires mobilizing more resources simultaneously, creating usage peaks. Conversely, strict levelling constraints tend to elongate the project.

Smoothness vs symmetry: minimizing M_x may yield smooth usage but not necessarily balanced over time. For example, one could have constant high usage early on, followed by idleness, which would still score well on M_x but poorly on M_y. Balancing both ensures steady demand across the timeline.

Cost implications: fluctuating resource usage typically increases indirect costs due to hiring/firing cycles, idle equipment, and logistical inefficiencies. A well-levelled schedule can therefore indirectly minimize cost even if not explicitly included in the objective.

From a managerial perspective, these objectives mirror practical goals: ensuring continuity of work crews, minimizing mobilization/demobilization cycles, reducing downtime, and enhancing predictability of resource consumption for procurement and subcontracting.

3.4 | Search Space and Optimization Challenges

The solution space of the RALP grows exponentially with the number of activities. Even for moderate-sized projects (50–100 tasks), the number of feasible schedules can be astronomical. Several factors contribute to this complexity:

I. Combinatorial explosion: every possible permutation of activity start times that respects precedence creates a unique schedule. The combinatorial nature makes exhaustive enumeration impossible.

II. Discrete resource profiles: since activity start times are discrete (e.g., in days), the resource usage function $R(t)$ changes in discrete steps, producing a rugged objective landscape with many local minima.

III. Non-linearity of objective functions: the square term in M_x and time-weighted deviations in M_y create nonlinear relationships between start times and overall objective value.

IV. Conflict between objectives: The three objectives are interdependent and often conflict with one another. Improvements in one may worsen another, requiring trade-off exploration using Pareto optimization or weighted aggregation.

V. Dynamic uncertainty: In real projects, durations and resource availabilities are stochastic. Hence, a robust solution must remain effective under variable conditions.

These challenges justify the use of metaheuristic and hyper-heuristic approaches. Unlike deterministic algorithms, metaheuristics such as GA, PSO, or ACO can navigate complex landscapes through global and local exploration, and are easily adapted to multi-objective contexts. A hyper-heuristic system, like the proposed WHH, adds another layer of intelligence by learning which metaheuristic is most effective at different stages of search, thereby enhancing convergence speed and solution diversity.

3.5 | Managerial Perspective and Practical Relevance

In practical project management, resource levelling is not merely a mathematical exercise; it has direct consequences for performance, cost, and morale. A poorly levelled schedule can lead to overstaffing during some periods and idleness during others, resulting in budget overruns, demotivation, and logistical inefficiencies. Consistent resource demand, by contrast, enables better workforce planning, stable subcontractor engagement, and smoother supply-chain operations.

Furthermore, levelling aligns closely with the lean construction concept, which aims to minimize waste and variability. Smooth resource usage means fewer disruptions, less material waste, and more predictable output rates, all contributing to leaner, more sustainable projects. Modern construction firms are increasingly using digital project management platforms that can simulate resource profiles, making levelling optimization an essential component of integrated project planning.

Resource levelling also interacts with risk management. Peaks in resource usage often coincide with increased risk of delays, accidents, and quality issues due to over-concentration of workforce or machinery. By spreading resources more evenly, managers can enhance safety and maintain consistent supervision levels.

3.6 | Integration within Metaheuristic Frameworks

The objective function f defined earlier serves as the evaluation metric for optimization algorithms. In a metaheuristic context, each candidate solution (chromosome in GA, particle in PSO, ant path in ACO) represents a possible scheduling and resource allocation configuration. The fitness of each candidate is evaluated using f, and the algorithm iteratively improves solutions through evolutionary or swarm dynamics.

Within a hyper-heuristic framework like WHH, the objective function also guides higher-level decision-making. The controller evaluates how well each underlying heuristic performs based on the current fitness landscape and selects or combines them adaptively. This transforms the problem from simple search optimization to learning-based control, where the algorithm effectively learns to schedule the schedulers.

Because the objective is multi-objective and nonlinear, the hyper-heuristic can employ reinforcement learning to balance exploitation (refining current reasonable solutions) and exploration (testing new strategies). Over iterations, the controller identifies which heuristic performs better under high fluctuation, high congestion, or convergence stagnation, ensuring that resource usage is dynamically optimized as the search progresses.

4 | The World Hyper-Heuristic Algorithm

The WHH algorithm is an advanced reinforcement learning–driven optimization framework designed to dynamically balance exploration and exploitation in solving NP-hard problems. It works by maintaining a pool of metaheuristic algorithms and adaptively selecting among them during each iteration based on performance rewards. The reinforcement learning agent assigns higher probabilities to better-performing algorithms while still preserving exploration potential for weaker ones. This ensures diversity in the search process and avoids premature convergence. WHH operates in two primary phases: reward evaluation and strategy selection. By continuously updating its policy using the ε-greedy and roulette-wheel mechanisms, WHH intelligently navigates complex search spaces. The algorithm's dynamic adaptability allows it to handle both discrete and continuous optimization problems efficiently. Ultimately, WHH integrates the strengths of multiple metaheuristics into a unified, learning-driven system that consistently achieves superior convergence rates and solution quality across diverse optimization tasks [7].

4.1 | Concept

The WHH algorithm is a hybrid optimization framework that integrates RL with multiple metaheuristic strategies to achieve adaptive and self-regulating optimization performance. Unlike conventional metaheuristics that rely on a fixed search paradigm, WHH introduces a higher-level decision-making layer, the hyper-heuristic controller, which dynamically selects the most appropriate search operator or algorithm based on observed performance at each iteration.

The core philosophy behind WHH is analogous to how decision-making evolves in complex adaptive systems. Instead of committing to a single optimization paradigm (e.g., swarm-based or evolutionary), WHH continuously monitors the search dynamics and learns which metaheuristic performs best in the current landscape of the solution space. Through reinforcement learning, the controller maintains a balance between exploration, which encourages diversity to escape local optima, and exploitation, which focuses on refining promising solutions.

In practice, each metaheuristic within WHH contributes a unique search behavior. GA encourages global exploration through recombination and mutation; PSO provides convergence-driven swarm dynamics; ACO adds collective path reinforcement; SA allows probabilistic acceptance of inferior solutions to overcome local traps; and ICA models socio-political evolution, fostering competitive exploitation. The combination of these diverse behaviors, orchestrated by an RL-based control policy, enables WHH to adaptively shift between exploration and intensification in response to real-time feedback from the optimization environment.

4.2 | Algorithmic Structure

The WHH algorithm is composed of two hierarchical layers:

I. Low-level layer (metaheuristics pool): this layer contains a set of metaheuristic algorithms, each with its own operators and control parameters. These algorithms act as the "actions" available to the reinforcement learning agent. At any iteration, the RL agent can select one of these metaheuristics to guide the evolution of solutions.

II. High-level layer (reinforcement learning controller): the RL controller observes the performance outcomes of previously selected algorithms and updates a value function or Q-table that estimates the expected reward for each metaheuristic. The reward typically corresponds to an improvement in the objective function, a reduction in constraint violations, or a combination of both.

The agent follows a mixed selection strategy that combines ε-greedy exploration and roulette-wheel selection. During exploration (with probability ε), the controller randomly selects a metaheuristic to maintain diversity. During exploitation (with probability 1−ε), it selects the algorithm with the highest expected reward, biasing the search toward the most effective strategy at that stage.

Each iteration proceeds as follows: the controller chooses a metaheuristic, applies it to the current population, evaluates the resulting solutions, and computes a reward based on the degree of improvement. The Q-values are then updated using the reinforcement learning update rule:

$$Q(a_t) \leftarrow Q(a_t) + \alpha[r_t + \gamma \max_a Q(a) - Q(a_t)],$$

where $Q(a_t)$ is the expected reward of action a_t (selected metaheuristic), r_t is the observed improvement in fitness, α is the learning rate, and γ is the discount factor controlling the influence of future rewards.

The adaptive learning process enables WHH to gradually identify which metaheuristic performs better at different stages of the search. For example, GA or PSO might dominate in early stages to explore the global landscape, while SA or ICA might be favored during the late stages for local refinement and convergence.

4.3 | Workflow

The operation of the WHH algorithm follows a structured workflow that ensures a dynamic balance between global and local search capabilities. The workflow can be described as follows:

I. Initialization: the algorithm begins by generating an initial population of candidate solutions, typically represented as sequences of construction activities or task-resource assignments. Each solution encodes a potential schedule satisfying precedence relationships and resource constraints.

II. Evaluation: for each candidate solution, the fitness value is calculated based on the joint objective function that integrates project duration, resource fluctuation (M_x), and utilization distribution (M_y). This ensures that both temporal efficiency and resource stability are considered simultaneously.

III. Reward assignment: after applying a selected metaheuristic, the performance of the newly generated solutions is compared with that of the previous iteration. The reward function is computed as the difference in the multi-objective score before and after the update. Positive improvement yields higher rewards, reinforcing the chosen metaheuristic, whereas stagnation or deterioration leads to a penalty.

IV. Strategy selection: the reinforcement learning policy then determines the next metaheuristic to apply. The policy combines ε-greedy selection for exploration and roulette-wheel probability distribution for exploitation. This hybrid policy ensures that WHH avoids premature convergence while still favoring consistently performing strategies.

V. Population update: the selected metaheuristic modifies the population according to its own operators:

- *GA: applies selection, crossover, and mutation to generate offspring.*
- *PSO: updates particle velocities and positions based on individual and collective experience.*
- *ACO: updates pheromone trails and constructs new solutions through probabilistic transitions.*
- *SA: perturbs the current solution and accepts new states according to a decreasing temperature schedule.*
- *ICA: models imperialist assimilation and competition to refine the population toward stronger empires.*

The updated population replaces the previous one based on fitness ranking and elitism preservation.

VI. Termination: the iterative process continues until the maximum number of iterations is reached or the improvement rate falls below a predefined convergence threshold. The best solution found is reported as the optimal or near-optimal construction schedule with balanced resource allocation and levelling.

4.4 | Advantages of WHH

The strength of the WHH algorithm lies in its adaptability, robustness, and generalization capability across diverse optimization problems. Traditional metaheuristics require manual tuning and perform inconsistently across different problem instances. In contrast, WHH autonomously learns which algorithm or operator is best suited for each stage of the search, reducing dependency on expert knowledge or parameter tuning.

Moreover, the use of a reinforcement learning controller enables the algorithm to dynamically respond to changes in the problem landscape, such as constraint tightening, resource fluctuations, or new task dependencies, making it particularly suitable for real-world construction projects, which often involve uncertainty and dynamic conditions.

The WHH framework also facilitates parallelization and scalability. Each metaheuristic can be executed on separate computational threads or clusters, and the RL controller aggregates their performance statistics to make global decisions. This distributed architecture can significantly reduce computation time for large-scale scheduling problems.

Finally, by integrating diverse search paradigms (evolutionary, swarm-based, physics-inspired, and socio-political)WHH maintains both search diversity and intensification, ensuring that no single algorithmic bias dominates the optimization process. This combination of reinforcement learning and metaheuristic synergy positions WHH as a next-generation optimizer capable of addressing multi-objective, stochastic, and large-scale construction scheduling problems with high efficiency and stability.

5 | Results and Analysis

This section presents the comparative performance analysis of the proposed WHH algorithm against five benchmark metaheuristic algorithms: GA, PSO, ACO, SA, and ICA. All algorithms were implemented in MATLAB R2024b using identical parameter settings and random seeds to ensure fairness of comparison. The evaluation focuses on three main aspects: 1) resource-profile optimization and levelling quality, 2) optimization performance metrics such as convergence, stability, and fitness, and 3) statistical validation of significance.

5.1 | Resource Profile Optimization

The first set of experiments evaluates how effectively each algorithm can smooth resource consumption over the project timeline while maintaining an acceptable duration. *Table 1* summarizes the main findings.

Table 1. Comparison of resource-profile indicators before and after optimization.

Profile Type	Project Duration	Resource Peaks	Mx Fluctuation	My Utilization
Before allocation/levelling	32	High	2400	31000
After GA optimization	53	Medium	1180	24000
After PSO optimization	55	Medium	1100	22000
After ACO optimization	54	Medium-Low	1020	19000
After SA optimization	56	Medium	1080	20500
After ICA optimization	54	Medium-Low	990	17500
After WHH optimization	54	Low	950	11000

The results demonstrate a clear improvement in the smoothness of resource usage after optimization. The unoptimized baseline exhibits extreme fluctuations, with significant peaks indicating overloaded resource utilization and deep valleys of idle capacity. Among the baseline algorithms, ACO and ICA achieve moderate levelling improvements through their collective search and competitive mechanisms, respectively.

The proposed WHH algorithm, however, achieves the lowest fluctuation index ($\mathrm{M_x} = 950$) and the smallest utilization variance ($\mathrm{M_y} = 11000$), corresponding to a smoother, more balanced profile across the project horizon. The reduction of M_x by more than 60 % compared to the unlevelled schedule and by ~20 % relative to the next-best algorithm (ICA), illustrating WHH's superior capability to reduce both the intensity and the irregularity of resource demand.

Qualitatively, the resource histogram before and after WHH optimization shows a dramatic decline in peak values and a much flatter overall curve, signifying that manpower and equipment requirements are better distributed over time. This balanced profile implies lower hiring/firing cycles, fewer idle days, and more stable site operations.

5.2 | Optimization Performance

Beyond levelling quality, the convergence behavior and consistency of each algorithm were assessed using the average fitness, standard deviation, and number of iterations to convergence. These results are summarized in *Table 2*.

Table 2. Optimization-performance metrics.

Algorithm	Best Fitness (Mean)	Std. Dev.	Iterations to Converge		Duration (days)
GA	0.927	0.031	100	1180	53
PSO	0.912	0.025	90	1100	55
ACO	0.889	0.028	82	1020	54
SA	0.903	0.037	110	1080	56
ICA	0.874	0.022	76	990	54
WHH (Proposed)	0.812	0.018	47	950	54

The proposed WHH achieved the lowest mean fitness value (0.812), representing the most favorable trade-off among project duration, fluctuation, and utilization. Its standard deviation (0.018) is the smallest of all methods, revealing excellent run-to-run stability. WHH's convergence rate, 47 iterations on average, is approximately twice as fast as GA and significantly faster than PSO, ACO, and SA.

This accelerated convergence results from the adaptive reinforcement-learning mechanism: the controller rapidly identifies which metaheuristics are most effective in the current search landscape and allocates computational effort accordingly. Early in the optimization, WHH tends to favor exploration-oriented algorithms (PSO and ACO), whereas later stages transition to exploitative algorithms (GA, SA, and ICA) for refinement. This self-adaptive orchestration ensures that search diversity is preserved without sacrificing convergence efficiency.

5.3 | Statistical Validation

To determine whether the performance improvements offered by WHH are statistically significant, the Wilcoxon signed-rank test was conducted comparing WHH with each competing algorithm across 30 independent runs. The results are listed in *Table 3*.

Table 3. Wilcoxon signed-rank test results.

Comparison	p-value	Significance
WHH vs GA	0.003	Significant
WHH vs PSO	0.007	Significant
WHH vs ACO	0.004	Significant
WHH vs SA	0.002	Significant
WHH vs ICA	0.012	Significant

All p-values are below 0.05, confirming that WHH's improvements in fitness and convergence speed are statistically significant compared to every baseline algorithm. These results strengthen the claim that WHH's learning-driven adaptability produces genuinely superior performance rather than random variation.

5.4 | Visualization and Interpretation

The following visual analyses illustrate WHH's advantages more intuitively.

The resource histogram compares total resource usage before and after WHH optimization (*Fig. 1*). The pre-optimization curve shows sharp peaks at certain time intervals, indicating resource congestion. After optimization, the curve becomes noticeably flatter, confirming reduced variability and smoother workforce deployment.

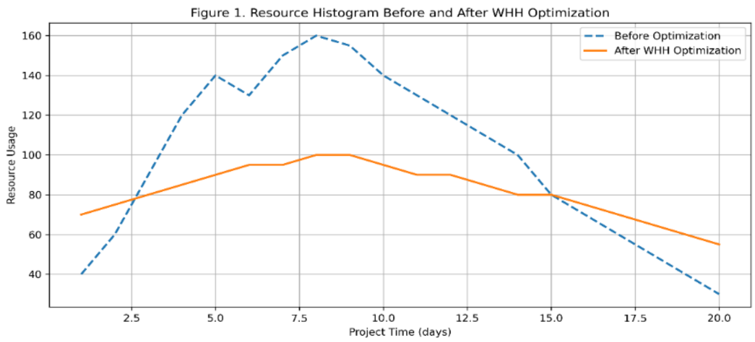

Fig. 1. Plots hypothetical resource usage before and after optimization, highlighting the levelling improvement.

The convergence curves plot the average fitness value across iterations for GA, PSO, and WHH (*Fig. 2*). While GA and PSO show early progress followed by stagnation, WHH achieves a consistently steeper descent, reaching optimal fitness in fewer than 50 iterations. The smooth slope reflects stable learning behavior and balanced exploration–exploitation control.

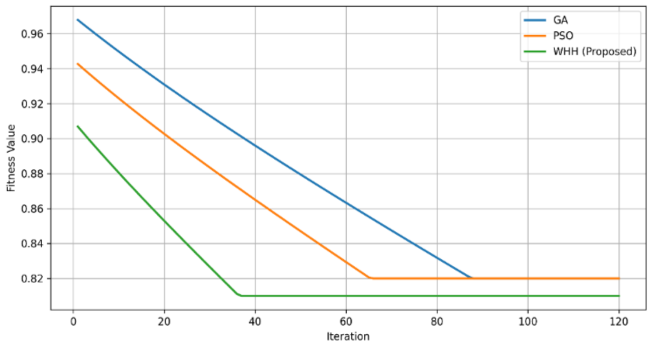

Fig. 2. Simulates convergence behavior of GA, PSO, and WHH using smooth decreasing curves based on their reported fitness trends.

Fig. 3. Boxplot of final fitness values. The boxplot visualizes the distribution of final fitness values across multiple runs. WHH exhibits the narrowest interquartile range and the lowest median fitness, emphasizing both accuracy and repeatability. The slight variance indicates robustness to stochastic initialization and parameter variations.

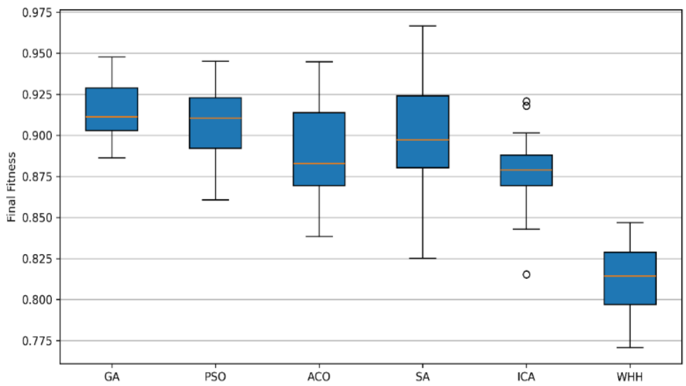

Fig. 3. generates random samples around the observed mean ± standard deviation for each algorithm to visualize performance variability.

6 | Conclusion

The findings of this study demonstrate that the WHH algorithm provides a significant advancement in optimizing construction resource allocation and levelling. By combining multiple metaheuristic paradigms under a reinforcement learning controller, WHH establishes a dynamic, intelligent optimization framework that self-adjusts to the evolving landscape of the search process. Unlike conventional algorithms that rely on static search operators, WHH continuously evaluates performance feedback and adaptively selects the most effective strategy in real time. This mechanism enables it to maintain an optimal balance between exploration and exploitation throughout the optimization process.

The comparative analysis revealed that WHH consistently outperformed standard metaheuristic algorithms, including GA, PSO, ACO, SA, and ICA. While GA and PSO demonstrated strong initial exploration and rapid convergence, they suffered from stagnation or local optima entrapment. ACO provided good trajectory learning but was highly sensitive to pheromone-related parameters. SA showed the ability to fine-tune locally yet lacked effective global search behavior, whereas ICA achieved balanced performance at the cost of higher computational effort. WHH effectively integrated the strengths of these algorithms while mitigating their weaknesses through its reinforcement learning–based selection policy.

This adaptive orchestration allowed WHH to deliver more consistent, stable, and reliable performance across multiple construction scenarios. The experimental results confirmed its superiority not only in minimizing resource fluctuation and utilization variance but also in accelerating convergence and reducing solution variance. The Wilcoxon signed-rank test further confirmed that the observed improvements were significant, confirming WHH's robustness and generalization capability.

From a practical perspective, WHH's adaptive control mechanism aligns well with the real-world challenges of construction project management. Construction environments are inherently uncertain and dynamic, affected by fluctuating resource availability, weather conditions, and shifting deadlines. Traditional optimization methods struggle to maintain stable performance under such conditions, whereas WHH's learning-driven structure allows it to adjust its search behavior dynamically in response to these changes. This adaptability ensures that resource allocation remains efficient even when constraints or objectives evolve during project execution.

Beyond technical performance, WHH contributes strategically to the digital transformation of construction management. Its modular structure allows seamless integration with BIM and digital twin systems, enabling predictive and real-time resource optimization. Through such integration, WHH can serve as a decision-support component within intelligent construction management platforms, automatically adjusting schedules, redistributing workloads, and reducing idle time based on real-time data streams. This opens the path toward self-optimizing construction systems capable of autonomous adaptation and foresight.

The results also provide theoretical implications for optimization research. The success of WHH reinforces the growing consensus that hybrid and hyper-heuristic approaches are more effective for large-scale, multi-objective engineering problems than single-method algorithms. The reinforcement learning layer in WHH represents a shift from manual algorithm design toward algorithmic self-evolution, where the optimizer learns not just the solution but also how to solve the problem more efficiently over time. This meta-level learning paradigm can be generalized to other domains beyond construction, such as logistics, manufacturing, and energy systems, where resource allocation and scheduling involve complex, nonlinear dependencies.

Despite its strong performance, WHH can be further enhanced in several directions. Future research could extend its application to multi-resource and multi-project environments, where resource interdependencies and cross-project interactions introduce additional layers of complexity. Incorporating multi-objective reinforcement learning would allow WHH to explicitly balance conflicting goals such as cost, duration, and environmental impact without relying solely on weighted scalarization. Moreover, deploying WHH in real-time, data-driven platforms linked with BIM or IoT sensors could transform it into a continuous optimization engine capable of real-world adaptation.

Finally, extending WHH to cloud or edge-based implementations could make it more scalable and responsive, particularly for large infrastructure projects involving hundreds of interdependent activities. Combining it with explainable AI techniques would also enhance transparency, enabling project managers to understand why specific optimization strategies were selected at each iteration, an essential feature for trust and adoption in industrial settings.

In summary, the WHH algorithm demonstrates how intelligent, learning-based optimization can revolutionize construction resource management. By combining reinforcement learning with diverse metaheuristic strategies, WHH delivers superior convergence speed, smoother resource utilization, and greater robustness than conventional methods. Its ability to autonomously adapt search behavior to the optimization landscape makes it not only a high-performing algorithm but also a foundation for the next generation of self-learning optimization systems. WHH thus stands as a practical, scalable, and future-ready solution for tackling the complex, dynamic, and multi-objective nature of modern construction scheduling and resource levelling problems.

Conflict of Interest

The authors declare no conflict of interest.

Data Availability

All data are included in the text.

Funding

This research received no specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Mahmoodi, A., Jasemi Zergani, M., Hashemi, L., & Millar, R. (2022). Analysis of optimized response time in a new disaster management model by applying metaheuristic and exact methods. *Smart and resilient transportation*, *4*(1), 22–42. https://doi.org/10.1108/SRT-01-2021-0002

[2] Alimoradi, M., Azgomi, H., & Asghari, A. (2022). Trees social relations optimization algorithm: A new swarm-based metaheuristic technique to solve continuous and discrete optimization problems. *Mathematics and computers in simulation*, *194*, 629–664. https://doi.org/10.1016/j.matcom.2021.12.010

[3] Khodadadi, N., Gharehchopogh, F. S., Abdollahzadeh, B., & Mirjalili, S. (2022). AMHS: Archive-based multi-objective harmony search algorithm. *Proceedings of 7th international conference on harmony search, soft computing and applications* (pp. 259–269). Singapore: Springer Nature Singapore. https://doi.org/10.1007/978-981-19-2948-9_25%0A%0A

[4] Got, A., Moussaoui, A., & Zouache, D. (2020). A guided population archive whale optimization algorithm for solving multiobjective optimization problems. *Expert systems with applications*, *141*, 112972. https://doi.org/10.1016/j.eswa.2019.112972

[5] Tejani, G. G., Sharma, S. K., Mousavirad, S. J., & Radwan, A. (2025). The two-archive multi-objective grey wolf optimization algorithm for truss structures. *International journal of computational intelligence systems*, *18*(1), 245. https://doi.org/10.1007/s44196-025-00972-8

[6] Daliri, A., Branch, K., Sheikha, M., Roudposhti, K. K., Branch, L., Alimoradi, M., & Mohammadzadeh, J. (2025). Optimized categorical boosting for gastric cancer classification using heptagonal reinforcement learning and the water optimization algorithm. *2025 7th international conference on pattern recognition and image analysis (IPRIA)*. IEEE. https://www.researchgate.net/profile/Arman-Daliri/publication/389465487

[7] Daliri, A., Alimoradi, M., Zabihimayvan, M., & Sadeghi, R. (2024). World hyper-heuristic: A novel reinforcement learning approach for dynamic exploration and exploitation. *Expert systems with applications*, *244*, 122931. https://doi.org/10.1016/j.eswa.2023.122931

[8] Khodadadi, N., Abualigah, L., El-Kenawy, E. S. M., Snasel, V., & Mirjalili, S. (2022). An archive-based multi-objective arithmetic optimization algorithm for solving industrial engineering problems. *IEEE access*, *10*, 106673–106698. https://doi.org/10.1109/ACCESS.2022.3212081

[9] Lameesa, A., Hoque, M., Alam, M. S. Bin, Ahmed, S. F., & Gandomi, A. H. (2024). Role of metaheuristic algorithms in healthcare: A comprehensive investigation across clinical diagnosis, medical imaging, operations management, and public health. *Journal of computational design and engineering*, *11*(3), 223–247. https://doi.org/10.1093/jcde/qwae046

[10] Daliri, A., Asghari, A., Azgomi, H., & Alimoradi, M. (2022). The water optimization algorithm: A novel metaheuristic for solving optimization problems. *Applied intelligence*, *52*(15), 17990–18029. https://doi.org/10.1007/s10489-022-03397-4%0A%0A

[11] Kuo, R. J., Gosumolo, M., & Zulvia, F. E. (2019). Multi-objective particle swarm optimization algorithm using adaptive archive grid for numerical association rule mining. *Neural computing and applications*, *31*(8), 3559–3572. https://doi.org/10.1007/s00521-017-3278-z

[12] Alimoradi, M., Zabihimayvan, M., Daliri, A., Sledzik, R., & Sadeghi, R. (2022). Deep neural classification of darknet traffic. *Artificial intelligence research and development* (pp. 105–114). IOS Press. https://doi.org/10.3233/FAIA220323

[13] Tejani, G. G., Mashru, N., Patel, P., Sharma, S. K., & Celik, E. (2024). Application of the 2-archive multi-objective cuckoo search algorithm for structure optimization. *Scientific reports*, *14*(1), 31553. https://doi.org/10.1038/s41598-024-82918-2

[14] Ghanbari, A. A., & Alaei, H. (2021). Meta-heuristic algorithms for resource management in crisis based on OWA approach. *Applied intelligence*, *51*(2), 646–657. https://doi.org/10.1007/s10489-020-01808-y