



A Multi-objective Optimization Model for Intelligent Information Management in Resource-Constrained Road and Transportation Projects

S. K. Mousavi^a, S. A. Hosseini^{*b}, H. R. Vosoughifar^b, R. Barandoust^c, G. Mohammadi^d

^a Department of Civil Engineering, Construction Management, ST.C., Islamic Azad University, Tehran, Iran

^b Department of Civil Engineering, ST.C., Islamic Azad University, Tehran, Iran

^c Department of Industrial Management, Y.I.C., Islamic Azad University, Tehran, Iran

^d Department of Mining Engineering, ST.C., Islamic Azad University, Tehran, Iran

PAPER INFO

Paper history:

Received 02 August 2025

Received in revised form 16 September 2025

Accepted 28 September 2025

Keywords:

Project Information Management

Critical Chain

Resource Constraints

Particle Swarm Optimization

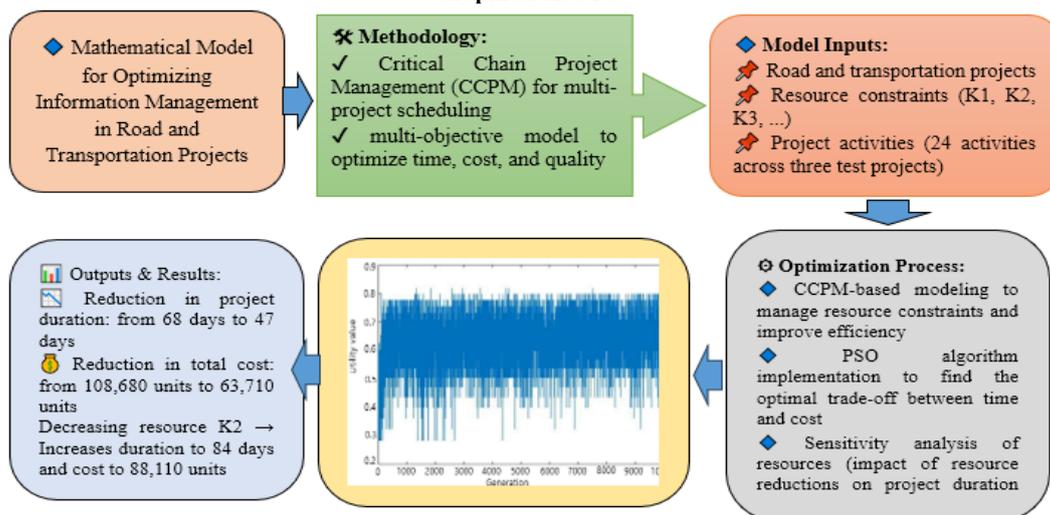
Multi-project Scheduling

ABSTRACT

This study proposes a mathematical model to optimize information management in road and transportation contracting projects under resource constraints, utilizing Critical Chain Project Management (CCPM). A multi-objective optimization framework is developed for multi-project scheduling, balancing time, cost, and quality as core performance indicators. The Particle Swarm Optimization (PSO) algorithm is employed to solve the model due to its efficiency in handling complex, nonlinear problems. The model is validated through a case study of three real-world projects encompassing 24 activities with diverse resource constraints. Compared to the traditional Critical Path Method (CPM), the CCPM-based approach reduces total duration from 68 to 47 days, total cost from 108,680 to 63,710 units, and improves the quality index from 0.412 to 0.5764, achieving a 21-day time saving, 25.97% cost reduction, and 40% quality improvement. Sensitivity analysis reveals that resource availability, particularly resources K2 and K3, significantly impacts outcomes, with K2 limitations increasing duration to 84 days and cost to 88,110 units, and K3 limitations extending duration to 81 days and cost to 86,910 units. These findings demonstrate the model's effectiveness in enhancing decision-making and resource planning in complex construction environments.

doi: 10.5829/ije.2026.39.08b.14

Graphical Abstract



*Corresponding author email: azim_hosseini@azad.ac.ir (S. A. Hosseini)

Please cite this article as: Mousavi SK, Hosseini SA, Vosoughifar HR, Barandoust R, Mohammadi G. A Multi-objective Optimization Model for Intelligent Information Management in Resource-Constrained Road and Transportation Projects. International Journal of Engineering, Transactions B: Applications. 2026;39(08):1953-75.

1. INTRODUCTION

The growing complexity and scale of road and transportation infrastructure projects have created a strong demand for advanced planning, coordination, and information management systems. Such projects typically involve multiple stakeholders—contractors, engineers, regulators, and suppliers—and take place in dynamic environments with changing resources, evolving regulations, and external uncertainties (1). Throughout the project lifecycle, from early planning and feasibility studies to execution and closure, large amounts of diverse data are generated. Effective management of this data is crucial for optimizing resource use, reducing delays, controlling costs, and supporting timely, data-driven decisions (2).

Despite the importance of effective information management, many transportation projects still rely on fragmented, informal, and largely manual processes. Data is often recorded inconsistently, stored in isolated silos, and exchanged through unstandardized or paper-based documentation. These shortcomings lead to miscommunication, slow response times, and poor decision-making, ultimately undermining overall project performance (3). The lack of a centralized platform for systematic data collection, processing, and analysis further amplifies problems such as redundancy, inconsistency, and inefficiency (4).

Over the past few decades, several project management methodologies—most notably the Critical Path Method (CPM) and, more recently, Critical Chain Project Management (CCPM)—have been developed to address challenges in scheduling and resource management. CCPM, introduced by Goldratt in the 1990s, incorporates resource constraints and buffer management into the scheduling process, improving resilience to uncertainty and aligning plans more closely with real-world resource dynamics (5, 6). Despite its potential, CCPM adoption in the transportation sector remains limited and fragmented. Available tools are often specialized for single functions, such as cost estimation or scheduling, and lack an integrated framework for managing the complexity of large infrastructure projects (7). One of the most critical challenges in such projects is the Resource-Constrained Project Scheduling Problem (RCPSP), particularly when multiple projects must compete for shared resources. RCPSP—and its extension, the Project Crashing Problem under Resource Constraints (PCPSP)—requires simultaneous optimization of competing objectives, including time, cost, and quality, while satisfying intricate precedence relationships and resource limitations (8).

Moreover, project stakeholders are often required to coordinate decisions across disconnected systems, frequently without access to real-time or reliable data. This lack of integration leads to reactive decision-making

and missed opportunities for proactive project control (9, 10). To overcome these challenges, advanced technologies—such as artificial intelligence (AI), big data analytics, and intelligent decision support systems (IDSS)—have been introduced. These solutions can automate data collection, detect bottlenecks, anticipate resource conflicts, and optimize schedules in near real time. Nevertheless, their practical implementation in transportation projects remains limited, especially when combined with advanced scheduling approaches like CCPM (11, 12).

This study introduces a comprehensive multi-objective optimization framework that integrates Critical Chain Project Management (CCPM) with an intelligent information management system and leverages the Particle Swarm Optimization (PSO) algorithm. PSO, a population-based metaheuristic, is well-known for its global search efficiency and fast convergence in complex solution spaces. The proposed framework is designed to tackle the challenges of multi-project scheduling under strict resource constraints, where conflicting objectives—minimizing total project duration and cost while maximizing execution quality—must be simultaneously optimized. Unlike conventional single-objective models or ad hoc scheduling tools, this approach applies a scalarized utility-based formulation to combine time, cost, and quality into a unified fitness function. Additionally, it incorporates CCPM-based dynamic buffer management to explicitly address execution uncertainties and resource variability, thereby improving schedule robustness and feasibility.

This study introduces a novel multi-objective optimization framework that integrates Critical Chain Project Management (CCPM) with Particle Swarm Optimization (PSO) to address resource-constrained multi-project scheduling in road and transportation projects. Unlike prior studies that apply PSO to the RCPSP without considering CCPM, the proposed model uniquely combines PSO's global search efficiency with CCPM's buffer management to simultaneously optimize time, cost, and quality. Furthermore, it incorporates an intelligent information management system to enable real-time, data-driven decision-making. This integrated approach fills a key gap in the literature by explicitly accounting for dynamic uncertainties and resource contention in complex infrastructure environments.

2. LITERATURE REVIEW

Efficient management and optimization of road construction and transportation projects are increasingly critical due to their inherent complexity, high costs, and limited resource availability. Recent advancements have focused on leveraging mathematical modeling, metaheuristic algorithms, and emerging digital

technologies to improve scheduling accuracy, cost control, and resource utilization. Despite these developments, the integration of multi-objective optimization (MOO) techniques with practical project management strategies under dynamic and uncertain real-world conditions remains a major research challenge.

Several studies have contributed to this domain through advanced modeling and algorithmic approaches. Anastasiu et al. (2) demonstrated that Critical Chain Project Management (CCPM) can reduce project durations by up to 20% compared with the Critical Path Method (CPM) through buffer management for schedule uncertainty. Wu et al. (13) applied Particle Swarm Optimization (PSO) for integer-based scheduling in robotic assembly, improving convergence speed and solution quality. Zhao et al. (14) developed a PSO-enhanced CCPM framework for renovation scheduling but considered only single-objective optimization. Ghoroghi et al. (15) employed Whale Optimization Algorithm (WOA) and NSGA-II for time–cost trade-offs in RCMPSP, showing robustness in solving integer-constrained scheduling problems.

Recent research has advanced MOO applications in project scheduling. Khoei et al. (4) provided a comprehensive survey of RCMPSP, emphasizing multi-objective extensions for transportation projects. Luo et al. (5) proposed an MOO model for multi-skill RCMPSP with flexible resource profiles, simultaneously optimizing time and cost. Wang et al. (9) developed a multi-project scheduling framework on the critical chain that jointly optimizes time, cost, and quality for transportation infrastructure. Li et al. (10) introduced an MOO approach for prefabricated construction scheduling under resource constraints, while Peng et al. (11) proposed a multi-skill scheduling model with quality-transmission mechanisms to mitigate rework risk in MSRCSP. Dashti et al. (12) addressed sustainability issues in multi-site RCMPSP by balancing environmental and operational objectives. Lotfi et al. (16) presented an MOO model for multi-mode RCMPSP with dynamic resource allocation, making it suitable for complex transportation projects. Singh (17) enhanced PSO for RCMPSP, achieving faster convergence under limited resource conditions. Mrabti et al. (18) proposed a priority-based heuristic for RCMPSP with a strong focus on sustainability.

Building on these contributions, several recent studies have introduced more comprehensive frameworks integrating MOO with environmental, safety, and network-level considerations. Zhao et al. (19) developed a low-carbon highway construction scheduling model that simultaneously minimizes cost, duration, and carbon emissions under resource limitations, offering a pathway for greener road infrastructure planning. Ye & Zhang (20) optimized heavy equipment transportation schedules

in road construction projects using NSGA-III, balancing transportation cost and delivery time under vehicle capacity constraints. Yu (21) proposed an intelligent construction scheduling framework integrating MOEA/D-DE, SPEA2+SDE, and NSGA-III, treating safety assessment and resource efficiency as additional objectives alongside time–cost trade-offs. Xie et al. (22) introduced a multi-objective decision-making method for road network maintenance using an improved NSGA-III algorithm to optimize cost, service quality, and resource utilization at the network level. Shahin et al. (23) presented a multi-objective optimization framework for highway work zones that simultaneously balances safety, mobility, and project cost, enabling informed and data-driven work-zone management decisions.

Collectively, these studies demonstrate significant progress in optimizing road and transportation projects using heuristic, metaheuristic, and digital tools. CCPM consistently outperforms CPM, improving schedule reliability and reducing delays (9, 24). The synergy between metaheuristic algorithms (e.g., PSO, WOA, NSGA-II/III) and intelligent technologies enhances resource allocation and enables dynamic project control (5, 11, 13, 21). MOO has proven crucial for balancing trade-offs among time, cost, quality, and increasingly, environmental and safety objectives (4, 9, 19, 22).

Nevertheless, important gaps remain. Most prior studies consider scheduling, resource management, or optimization objectives in isolation, often relying on single-objective formulations that fail to capture the interrelated nature of real-world constraints. While CCPM effectively addresses schedule uncertainty, its integration with advanced multi-objective metaheuristics and real-time data streams in infrastructure projects remains limited.

To bridge these gaps, this study introduces a novel intelligent scheduling model that combines CCPM's execution-control features with PSO's global search capability within a unified multi-objective optimization framework. The proposed approach jointly optimizes project duration, cost, and resource utilization while accounting for real-world uncertainties such as fluctuating material supply, variable labor availability, and equipment shortages—conditions typical of large-scale transportation projects. This study contributes a comprehensive, adaptive, and performance-driven methodology for multi-project scheduling and intelligent project governance in the transportation sector.

The comparison in Table 1 highlights the novelty of the proposed model in addressing gaps in existing literature. While studies such as Wang et al. (9) and Luo et al. (5) explore MOO in critical chain and multi-skill contexts, they lack the combined use of CCPM and PSO for simultaneous optimization of time, cost, and quality under real-world uncertainty. Similarly, Zhao et al. (19) and Ye & Zhang (20) incorporate sustainability and

TABLE 1. Summary of Related Literature Based on Objectives, Constraints, and Solution Methods

Study	Objectives Considered	Constraints Modeled	Solution Algorithm	Distinction from Proposed Model
Anastasiu et al. (2)	Project duration reduction	Task buffering, precedence	Critical Chain Project Management (CCPM)	Uses CCPM alone; lacks PSO and multi-objective formulation
Wu et al. (13)	Makespan minimization	Integer scheduling, robotic assembly	Particle Swarm Optimization (PSO)	PSO without CCPM; single-objective, not tailored for transportation projects
Zhao et al. (14)	Renovation schedule optimization	Task sequencing, discrete variables	PSO + CCPM	Focused on renovation; single-objective, lacks cost and quality optimization
Ghoroqi et al. (15)	Time–cost trade-offs	Integer constraints, supplier capacity	WOA + NSGA-II	Robust for time–cost trade-off; no CCPM or quality consideration
Khoei et al. (4)	Review of RCMPSP approaches	Resource constraints	Literature survey	Provides taxonomy; does not propose a computational framework
Luo et al. (5)	Multi-skill scheduling (time, cost)	Skill switches, resource limits	Evolutionary approaches	Addresses multi-skill RCMPSP; lacks CCPM integration and real-time decision-making
Wang et al. (9)	Time–cost–quality optimization	Resource limits, uncertainty	Multi-objective optimization (MOO)	Focused on critical chain scheduling; no metaheuristic integration (PSO)
Li et al. (10)	Prefabricated construction scheduling	Time, resource constraints	MOO framework	Prefabricated project focus; lacks CCPM-based uncertainty control
Peng et al. (11)	Quality-transmission optimization	Rework, skill constraints	Mathematical optimization model	Quality-focused; no CCPM integration for multi-project control
Dashti et al. (12)	Sustainable multi-site scheduling	Multi-site resources	MOO algorithm	Sustainability emphasis; lacks CCPM-PSO integration
Lotfi et al. (16)	Multi-mode scheduling (time–cost)	Dynamic resources	MOO framework	Dynamic resource allocation; lacks CCPM buffer management
Singh (17)	Time–cost optimization	Resource constraints	Enhanced PSO	Uses improved PSO; no CCPM-based uncertainty handling.
Mrabti et al. (18)	Time, cost, sustainability	Resource constraints	Priority-based heuristic	Focuses on sustainability; lacks PSO-CCPM synergy
Zhao et al. (19)	Time–cost–carbon minimization	Resource limitations	Multi-objective optimization model	Integrates carbon emission objective; no CCPM or intelligent information layer
Ye & Zhang (20)	Cost and delivery time optimization	Vehicle capacity, logistics	NSGA-III	Focused on heavy equipment transport; lacks CCPM buffer and multi-project scheduling
Yu (21)	Time–cost–safety–resource efficiency	Resource limits, safety constraints	MOEA/D-DE + SPEA2+SDE + NSGA-III	Integrates safety assessment; no CCPM execution control or real-time feedback loop
Xie et al. (22)	Cost, service quality, resource utilization	Network-level maintenance constraints	Improved NSGA-III	Network-wide decision-making; lacks project-level CCPM buffer management
Shahin et al. (23)	Safety, mobility, cost	Work-zone capacity, traffic flow	Multi-objective optimization framework	Emphasis on work zone planning; does not integrate CCPM with metaheuristics
This Study	Time, cost, quality (simultaneous)	Precedence, resource limits, uncertainty	CCPM + PSO (MOO)	Integrates CCPM buffer management with PSO global search; real-time intelligent information management for transportation projects.

logistics considerations but omit intelligent information management and CCPM-driven buffer control. The proposed model uniquely integrates CCPM's dynamic execution-control features with PSO's global search capability in a unified MOO framework, enabling adaptive and data-driven decision-making for large-scale, resource-constrained transportation projects.

3. PROPOSED PROBLEM AND MATHEMATICAL FORMULATION

3.1. Problem Statement

Scheduling multiple

interdependent projects under limited shared resources represents a well-known yet unresolved challenge in large-scale infrastructure project management, particularly in the transportation sector. In such environments, project activities are often executed concurrently, require the same pools of labor, equipment, and materials, and are subject to dynamic uncertainties—such as demand variability, procurement delays, or resource unavailability. These factors can propagate throughout the project network, resulting in significant delays, cost overruns, and degradation in quality performance.

To address this challenge, this study adopts the Critical Chain Project Management (CCPM) methodology, which enhances traditional project scheduling techniques (e.g., Critical Path Method or PERT) by integrating resource availability into the scheduling logic and employing time buffers to absorb execution variability. CCPM ensures that activities are scheduled based on both precedence logic and resource contention, and that uncertainties are managed proactively through buffer zones strategically inserted in the project timeline.

The main objective of this study is to develop a multi-objective mathematical model that supports critical chain scheduling for multi-project environments under realistic conditions. The model explicitly considers:

- Minimization of total project duration,
- Minimization of execution cost, and
- Maximization of delivered quality.

Additionally, the model incorporates:

- Activity precedence relationships,
- Resource availability constraints (renewable and non-renewable),
- Buffer sizing for uncertainty absorption,

A set of decision variables for activity start times, resource assignments, and project crashing levels.

Given the NP-hard nature of the underlying multi-objective resource-constrained multi-project scheduling problem (MORCMPS), an efficient solution technique is necessary. To that end, the model will be solved using a metaheuristic optimization algorithm based on Particle Swarm Optimization (PSO), which is adapted to handle the multi-objective structure and discrete scheduling decisions. This integrated framework provides a flexible and scalable foundation for intelligent scheduling in large-scale transportation projects and contributes to more reliable, resource-aware, and uncertainty-resilient project planning.

3. 2. Notation: Indices, Parameters, and Decision Variables To formulate the proposed multi-objective scheduling model, the following notations are used:

Indices

$i \in \{1, 2, \dots, N\}$: Index of projects

$j \in \{1, 2, \dots, J\}$: Index of activities within each project

$k \in \{1, 2, \dots, K\}$: Index of renewable resources

$p \in \{1, 2, \dots, P\}$: Index of non-renewable resources

t_{ij} : Planned duration of activity j in project i

r_{jk} : Required amount of renewable resource k for activity j

C_k : Unit cost of renewable resource k

nr_{ijp} : Required amount of non-renewable resource p for activity j in project i

C_p : Unit cost of non-renewable resource p

R_k : Available quantity of renewable resource k

S_i : Daily overhead cost of project i

EV_{ij} : Earned value of activity j in project i

q_{ij} : Quality index of activity j in project i

EQV_{ij} : Earned quality value of activity j in project i

α_T : Weight coefficient for project duration (time objective)

α_C : Weight coefficient for project cost

α_Q : Weight coefficient for project quality

D : Desired total project duration

Z : Desired total project cost (21).

Decision Variables

E_{e0} : Finish time of the virtual final activity (project makespan)

PB : Project buffer size

T : Total project duration (from Critical Chain schedule)

C : Total cost of executing all projects

Q : Total quality score achieved across all projects

T_{cs} : Duration of total critical chain activities (sum of task segments)

$U(T)$: Time-based utility function

$U(C)$: Cost-based utility function

$U(Q)$: Quality-based utility function

This comprehensive set of notations forms the basis for constructing the proposed mathematical model. It enables the precise formulation of trade-offs among time, cost, and quality objectives while respecting practical constraints such as resource availability and project dependencies. The notations are also designed to facilitate efficient encoding within the optimization algorithm used to solve the model.

3. 3. Model Formulation This section presents the mathematical formulation of the proposed multi-objective optimization model for critical chain scheduling in multi-project environments. The model integrates three conflicting objectives: minimizing project duration, minimizing total execution cost, and maximizing output quality. These objectives are formulated independently and aggregated via a scalarized utility-based approach. Constraints related to precedence, resource availability, and execution feasibility are incorporated to ensure practical applicability.

3. 3. 1. Time Objective Function (CCPM) The primary objective of minimizing total project duration is addressed using Critical Chain Project Management (CCPM), which accounts for resource contention and uncertainty through buffer management, unlike traditional methods such as the Critical Path Method (CPM). The time objective is formulated as:

$$\min T = E_{e0} + PB \quad (1)$$

where:

E_{e0} : Completion time of the virtual final activity across all projects (project makespan).

PB : Project buffer, defined as 50% of the critical chain length.

Scientific Rationale: Following the Theory of Constraints, the critical chain represents the sequence of tasks constrained by both precedence and resource availability. A centralized project buffer absorbs aggregate uncertainty, enhancing schedule reliability and eliminating hidden safety margins in individual task durations.

To illustrate the application of this formulation, consider a project with a critical chain comprising two sequential activities, each with a duration of 10 days, yielding a critical chain length of 20 days. If the virtual final activity completes at ($E_{e0} = 20$) days, the project buffer is calculated as ($PB = 0.5 \times 20 = 10$) days. Consequently, the total project duration is ($T = 20 + 10 = 30$) days, ensuring robustness against uncertainties.

Advanced Consideration: The buffer (PB) can be dynamically adjusted based on probabilistic duration estimates or historical variability, enabling integration with stochastic methods such as Monte Carlo simulation or Bayesian updating for enhanced uncertainty modeling.

3. 3. 2. Cost Objective Function The total project cost (C) encompasses direct costs from renewable and non-renewable resource consumption and indirect overhead costs associated with project duration. The cost objective is expressed as:

$$minC = \sum_{i=1}^N (\sum_{j=1}^J (\sum_{k=1}^K r_{ijk} \cdot C_k \cdot t_{ij} \sum_{p=1}^P nr_{ijp} \cdot C_p) S_i t_i) \quad (2)$$

where:

r_{ijk} and nr_{ijp} : Quantities of renewable (e.g., labor, equipment hours) and non-renewable (e.g., materials) resources required for activity (j) in project (i).

C_k, C_m : Unit costs of renewable and non-renewable resources, respectively.

O_i : Daily overhead cost for project (i).

T_i : Duration of project (i).

Scientific Rationale: This formulation captures the resource-constrained dynamics of multi-project environments, where activities compete for limited resources. The inclusion of renewable and non-renewable resources ensures granular cost estimation, while the overhead term (O_i, T_i) accounts for indirect costs (e.g., administrative support, equipment idle time), incentivizing efficient scheduling.

To demonstrate the cost calculation, consider an activity j in project i requiring 10 units of a renewable resource k_1 (unit cost ($c_{k1} = 100$), 5 units of a non-renewable resource (m_1) unit cost ($c_{m1} = 200$), and a project with a daily overhead cost ($O_i = 500$) units over a duration ($T_i = 10$) days. The cost for this activity is computed as:

$$[C = (10 \cdot 100) + (5 \cdot 200) + (500 \cdot 10) = 1,000 + 1,000 + 5,000 = 7,000]$$

Advanced Consideration: The linear cost model can be

extended to incorporate nonlinear structures, such as economies of scale, time-dependent cost variations, or contractual penalties for delays, enhancing realism. Stochastic techniques, such as scenario-based simulation or fuzzy costing, can further address price uncertainties.

3. 3. 3. Quality Objective Function The quality dimension is modeled using the Earned Quality Value (EQV) framework, which quantifies the contribution of each activity to the overall project quality.

Definition of earned quality value for activity j in project i :

$$EQV_{ij} = EV_{ij} \times q_{ij} \quad (3)$$

where:

EQV_{ij} : Earned quality value for activity j in project i ,

EV_{ij} :

Earned value of activity j in project i , representing progress/completion status ($0 \leq EV_{ij} \leq 1$).

q_{ij} : Quality index of activity j in project i , reflecting compliance with material quality, execution standards, and specifications ($0 \leq q_{ij} \leq 1$).

Overall quality objective function:

$$Q_{max} = \sum_{i=1}^N \sum_{j=1}^J EQV_{ij} \quad (4)$$

where:

Q : Overall quality score across all projects.

N : Number of projects.

J : Number of activities per project.

For activity j in project i :

$$EV_{ij} = 0.8, q_{ij} = 0.9$$

then:

$$EQV_{ij} = 0.8 \times 0.9 = 0.72$$

Thus, the activity contributes 0.72 units of quality value toward the total project quality score.

3. 4. Utility Function To aggregate the three conflicting objectives—time, cost, and quality—into a single optimization framework, a quadratic utility-based scalarization approach is used. This allows for differential weighting and penalizes large deviations more severely than small ones, which better reflects real-world project risk profiles.

Utility Aggregation Function:

$$U(T, C, Q) = \alpha_T \cdot U(T) + \alpha_C \cdot U(C) + \alpha_Q \cdot U(Q) \quad (6)$$

Subject to:

$$\alpha_T, \alpha_C, \alpha_Q \geq 0 \quad (7)$$

$$\alpha_T, \alpha_C, \alpha_Q = 1 \quad (8)$$

Time Utility Function:

$$U(T) = \begin{cases} \varphi_T - \beta_T(T - D)^2 & \text{if } T \in [0, 2D] \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Cost Utility Function:

$$U(C) = \begin{cases} \varphi_C - \beta_C(C - (1 - \eta) \cdot Z)^2 & \text{if } C \in [0, 2(1 - \eta)Z] \\ 0 & \text{otherwise} \end{cases} \quad (10)$$

Quality Utility Function:

$$U(Q) = \begin{cases} \varphi_Q - \beta_Q(Q - 1)^2 & \text{if } Q \in [0, 1.1] \\ 0 & \text{otherwise} \end{cases} \quad (11)$$

Scientific Rationale:

Quadratic utility functions are well-suited to model real-world project sensitivity: small deviations from target values are tolerated, while larger deviations result in disproportionately high penalties. This aligns with contractual structures and operational expectations in transportation infrastructure projects.

3. 5. Final Optimization Model The final multi-objective model is defined as the maximization of the aggregated utility function subject to project execution constraints:

$$\max U(T, C, Q) \quad (12)$$

Subject to:**1. Precedence Constraints:**

$$E_{ij} - E_{i(j-1)} \geq t_{ij} \quad \forall i, j \quad (13)$$

2. Project Buffer Calculation:

$$PB = \frac{T_{ls}}{2} \quad (14)$$

3. Resource Availability Constraints:

$$\sum_{i=1}^N \sum_{j=1}^J r_{ijk} \leq R_k \quad \forall t \in T, \forall k \quad (15)$$

Model Characteristics:

This model is nonlinear, multi-objective, and combinatorial, making it computationally intensive. Therefore, heuristic or metaheuristic approaches—such as the Particle Swarm Optimization (PSO) algorithm—are applied to efficiently approximate near-optimal solutions for large-scale problem instances.

3. 6. Solution Approach: Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) is a population-based metaheuristic inspired by the social behavior of organisms such as bird flocks or fish schools. Each potential solution—called a particle—navigates the search space by adjusting its position based on its own experience and that of its neighbors. Owing to its balance between exploration and exploitation, PSO has proven effective in solving complex, high-dimensional, nonlinear optimization problems, including multi-project scheduling under resource constraints.

The PSO algorithm is particularly suited to the proposed problem due to the following characteristics:

- **Global Search Capability:** Particles benefit from social learning, enabling them to avoid local optima.

- **Rapid Convergence:** PSO often reaches high-quality solutions with fewer iterations compared to evolutionary algorithms.
- **Structural Simplicity:** Unlike Genetic Algorithms, PSO does not require complex genetic operators like crossover or mutation.
- **Scalability:** The algorithm handles large search spaces typical in infrastructure scheduling with numerous tasks and constraints.

3. 6. 1. PSO Representation in This Study In this implementation, each particle is encoded as a real-valued vector of dimension $n = \sum_{i=1}^N |J_i|$ where J_i is the number of activities in project i . Each element of the particle vector represents the priority index of a corresponding activity.

Encoding:

Particle $X = [x_1, x_2, \dots, x_n]$, where each $x_j \in R_{x_j}$ maps to an activity a_j .

Decoding:

The vector is decoded into an executable schedule using the Serial Schedule Generation Scheme (SSGS), which respects activity precedence and produces a time-feasible sequence. This priority-based representation allows for flexible schedule construction and efficient integration with resource and precedence constraints.

3. 6. 2. Constraint Handling Mechanisms To ensure solution feasibility, a three-level constraint-handling strategy is integrated into the PSO framework:

1. Precedence Constraint Enforcement

The SSGS decoding mechanism inherently guarantees that no activity starts before all its predecessors are completed.

As a result, precedence infeasibility is eliminated during schedule generation.

2. Resource Conflict Resolution

After initial decoding, the schedule is checked against renewable and non-renewable resource capacities.

If violations are detected (e.g., resource overload on specific time units), a repair heuristic is applied:

- Backward Shifting: Conflicting activities are delayed until enough resources become available.
- This ensures resource-feasible schedules while minimizing disruption to the original sequence.

3. Penalty Function Integration

In early iterations or tight constraint settings, infeasible solutions may still arise. To address this, a penalty-based fitness function is used:

$$F(x) = f(x) + \sum_{i=1}^K \lambda_i \cdot \text{Penalty}_i(x) \quad (16)$$

where:

$f(x)$: Original scalarized utility function (from Section 3.4)
 λ_i : Adaptive penalty weights for each constraint type (e.g., precedence, resource).

$Penalty_i(x)$: Quantified severity of constraint violation i for solution x

This mechanism discourages particles from exploring infeasible regions and dynamically shifts search pressure toward feasible and promising areas.

Scientific Rationale:

This hybrid approach—combining priority-based encoding, SSGS decoding, and multi-layer constraint repair—mirrors techniques used in advanced metaheuristic project optimization studies. It enables the algorithm to search effectively in constrained solution spaces while maintaining feasibility, responsiveness, and optimization quality.

3. 6. 3. Constraint Handling Mechanisms To ensure the feasibility of generated schedules, particularly under precedence and resource constraints, the PSO algorithm in this study integrates a multi-layered constraint-handling strategy. This framework includes: 1) precedence enforcement, 2) resource conflict resolution, and 3) penalization for infeasible solutions.

1. Precedence Constraint Enforcement

Each particle represents a real-valued vector $X = [x_1, x_2, \dots, x_n]$, where each value encodes the relative execution priority of an activity. This vector is decoded using the Serial Schedule Generation Scheme (SSGS). At each construction step, SSGS:

- A. Identifies the set of eligible activities A_t whose predecessors have been completed,
- B. Selects the next activity from A_t based on the priority encoded in the particle,
- C. Updates the project schedule by assigning the earliest feasible start time to the selected activity.

This ensures all precedence relationships are strictly maintained and no activity is scheduled before its logical predecessors.

2. Resource Conflict Resolution

After generating a precedence-feasible schedule, the model checks the daily utilization of renewable and non-renewable resources. If resource usage exceeds availability on any time unit, a repair heuristic is applied:

- A. For each over-allocated resource k at time t , identify conflicting activities C_t^k .
- B. Iteratively backward-shift the start times of lowest-priority activities in C_t^k until resource usage becomes feasible.
- C. Re-check constraints and repeat until all resource violations are eliminated.

This ensures that schedules remain resource-feasible without violating precedence.

3. Penalty Function for Residual Infeasibility

In rare cases where SSGS and repair mechanisms cannot fully eliminate conflicts (e.g., due to swarm randomness or high constraint tightness), a penalty function is introduced in the fitness evaluation:

$$F(x) = f(x) + \sum_{i=1}^m \lambda_i \cdot Penalty_i(x) \quad (17)$$

where:

$f(x)$: Utility-based fitness value (see section 3.4),

λ_i : Adaptive weight for constraint type i ,

$Penalty_i(x)$: Severity of violation (e.g., amount of resource overuse, time lag, buffer violation).

This penalization discourages infeasible solutions and gradually guides the swarm toward the feasible region of the search space.

3. 6. 4. Velocity and Position Update Mechanism

In the Particle Swarm Optimization (PSO) algorithm, each particle iteratively adjusts its velocity and position by combining three main components: inertia (previous momentum), cognitive learning (personal experience), and social learning (collective knowledge of the swarm). The update rules are expressed as follows:

Velocity Update Equation:

$$v_j^{(t+1)} = \omega \cdot v_j^{(t)} + c_1 \cdot r_1 \cdot (p_{best,j} - x_j^{(t)}) + c_2 \cdot r_2 \cdot (g_{best,j} - x_j^{(t)}) \quad (18)$$

Position Update Equation:

$$x_j^{(t+1)} = x_j^{(t)} + v_j^{(t+1)} \quad (19)$$

where:

$x_j^{(t)}$: current position (priority) of activity j at iteration t .

$v_j^{(t)}$: velocity of activity j at iteration t .

$p_{best,j}$: the best position found so far by the particle for activity j .

$g_{best,j}$: the best position found globally by the entire swarm for activity j .

c_1, c_2 : acceleration (learning) coefficients, representing the weights of cognitive and social components.

r_1, r_2 : independent random numbers uniformly distributed in $[0,1]$.

ω : inertia weight, controlling the influence of previous velocity on the current update.

This mechanism allows the swarm to balance exploration and exploitation effectively. Exploration is stimulated by the stochastic (random) and social learning terms, ensuring that diverse regions of the search space are investigated. Exploitation is driven by the cognitive and memory-based components, allowing particles to refine solutions around promising regions. The inertia weight ω plays a central role by regulating convergence speed and stability: larger values encourage global exploration, while smaller values promote local exploitation. Similarly, the parameters c_1 and c_2 define the trade-off between individual learning and collective cooperation, which is critical for adapting the algorithm to complex, resource-constrained scheduling problems.

3. 6. 5. PSO-Based Solution Procedure and Applicability

The Particle Swarm Optimization (PSO) framework employed in this study is designed to

solve the proposed multi-objective, resource-constrained, multi-project scheduling problem by integrating quality-aware and constraint-sensitive mechanisms. The algorithm proceeds through the following structured steps:

Step 1: Initialization

A population P of M particles is randomly generated.

Each particle $x^m \in R^n$ encodes a priority-based activity sequence for all tasks across all projects, where $n = \sum_{i=1}^N J_i$

Each particle is initialized with:

A random position vector $x^m = [x^1, x^2, \dots, x^n]$

A velocity vector $v^m \in R^n$

An initial fitness value $F(x^m)$

Step 2: Schedule Decoding and Constraint Repair

Each particle is decoded using the Serial Schedule Generation Scheme (SSGS) to generate a precedence-feasible schedule.

The resulting schedule is evaluated for daily resource feasibility.

If resource conflicts are detected:

- A. A backward-shifting repair heuristic is applied to resolve over-allocation.
- B. Activities are delayed iteratively until resource availability is met.

If any residual violations remain, a penalty function is added to the fitness value.

$$F(x) = \alpha_T \cdot U(T) + \alpha_C \cdot U(C) + \alpha_Q \cdot U(Q) + \sum_i \lambda_i \cdot Penalty_i(x) \quad (20)$$

Here, $U(T), U(C), U(Q)$ are quadratic utility functions for time, cost, and quality, and λ_i are adaptive penalty weights for each constraint violation.

Step 4: Personal and Global Best Update

For each particle:

If $F(x^m)$ is better than its personal best p_{best}^m , it is updated.

The best-performing particle across the swarm is recorded as g_{best} .

Step 5: Velocity and Position Update

Particle velocities and positions are updated using the canonical PSO equations:

$$\begin{aligned} v_j(t+1) &= \omega \cdot v_j(t) + c_1 \cdot r_1 \cdot (p_{best,j} - x_j(t)) + \\ & c_2 \cdot r_2 \cdot (g_{best,j} - x_j(t)) \\ x_j^{(t+1)} &= x_j^{(t)} + v_j^{(t+1)} \end{aligned} \quad (21)$$

Random variables $r_1, r_2 \sim U(0,1)$ ensure exploration diversity.

Step 6: Termination Criteria

The algorithm iterates through steps 2–5 until one of the following criteria is met:

Maximum number of iterations T_{max} is reached.

No improvement in the global best fitness for k consecutive iterations (stagnation).

A satisfactory solution is found (fitness threshold).

Applicability and Efficiency:

This hybrid PSO framework ensures that all logical and resource constraints are respected through decoding and repair, while the utility-based objective function drives the search toward solutions that balance time, cost, and quality. The approach is scalable and adaptable for real-world transportation projects involving hundreds of interdependent tasks and tight resource constraints.

3. 7. PSO Update Mechanism and Algorithm Workflow

The Particle Swarm Optimization (PSO) algorithm is designed to solve the multi-objective scheduling problem by iteratively updating particle positions and velocities to optimize the objective functions (T, C, Q). To manage uncertainties, the PSO algorithm incorporates a scenario-based approach, evaluating multiple duration and resource availability scenarios for each particle. The velocity update includes a stochastic factor to enhance exploration under uncertainty, defined as:

The proposed Particle Swarm Optimization (PSO) algorithm is adapted to address the multi-objective resource-constrained scheduling problem by simultaneously optimizing project duration (T), cost (C), and quality (Q). Unlike conventional deterministic PSO, the algorithm incorporates a scenario-based evaluation to explicitly account for uncertainties in activity durations and resource availability. Each particle represents a feasible schedule encoded as a sequence of activities, and its fitness is assessed under multiple stochastic scenarios. The velocity update rule is modified to balance exploration and exploitation while mitigating premature convergence. Specifically:

$$v_i^{(t+1)} = \omega \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (p_i^{best} - x_i^{(t)}) + c_2 \cdot r_2 \cdot (g^{best} - x_i^{(t)}), \quad x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \quad (22)$$

where:

w : is the inertia weight controlling exploration,

c_1 and c_2 are cognitive and social learning coefficients,

$r_1, r_2 \sim U(0,1)$: are independent random numbers,

ϵ : is a stochastic perturbation coefficient,

$\text{Rand}(-0.1,0.1)$ introduces bounded randomness to enhance robustness.

The global best (g_{best}) is determined using a weighted utility function:

$$U = \alpha T f T(T) + \alpha C f C(C) + \alpha Q f Q(Q), \alpha T + \alpha C + \alpha Q = 1$$

The algorithm workflow is summarized as:

1. Initialize swarm with feasible schedules.
2. Generate scenarios for durations and resources.
3. Evaluate fitness (T, C, Q, U) under scenarios.
4. Update velocities and positions.
5. Update p_{best} and g_{best} .
6. Terminate upon convergence.

This scenario-based PSO with stochastic perturbation improves robustness compared to deterministic implementations.

3. 8. Justification of Feasibility and Constraint Compliance

The feasibility of the schedules generated by the proposed PSO-based framework is ensured through a structured, multi-layered approach to constraint handling, fully integrated within the optimization process. The model guarantees compliance with both precedence and resource constraints through the following mechanisms:

- **Precedence Constraint Enforcement:** All decoded particles are processed using the Serial Schedule Generation Scheme (SSGS), which inherently enforces task sequencing. This ensures that no activity begins before the completion of all its predecessors, thereby eliminating precedence violations.
- **Resource Feasibility through Repair Heuristics:** After generating precedence-feasible schedules, the algorithm checks daily resource utilization. In case of over-allocations, a backward-shifting repair heuristic delays the conflicting activities until sufficient resources become available, ensuring daily feasibility across the scheduling horizon.
- **Penalty Function for Residual Violations:** In the rare case where resource constraints cannot be entirely satisfied through repair (e.g., in early iterations or tightly constrained settings), a penalty term is added to the utility function. This penalization mechanism reduces the fitness of infeasible solutions and guides the swarm toward feasible regions of the solution space.

4. DESIGN OF THE PARTICLE SWARM OPTIMIZATION ALGORITHM

To address the multi-objective, resource-constrained multi-project scheduling problem under Critical Chain Project Management (CCPM) principles, a customized Particle Swarm Optimization (PSO) algorithm is developed. The algorithm is specifically designed to explore the solution space of priority-based activity sequences while simultaneously satisfying logical (precedence) and operational (resource) constraints.

To ensure robust and efficient optimization performance, several key parameters of the PSO algorithm must be properly configured. These parameters influence swarm dynamics, convergence behavior, and the algorithm’s ability to balance exploration and exploitation. Table 2 presents the configuration parameters employed in this study, selected based on prior empirical research and sensitivity analysis.

Particle Encoding and Initialization

Each particle encodes a real-valued vector $x^i = [x_1^i, x_2^i, \dots, x_n^i]$, where n is the total number of activities. The values indicate the priority of each activity and are decoded into executable schedules using SSGS.

- The initial population is generated randomly, ensuring that activity sequences respect precedence relationships.
- The position and velocity vectors are initialized uniformly within the specified bounds.

Fitness Evaluation

Each particle is decoded into a feasible schedule, and the following scalar utility function is used to evaluate performance:

$$F(x) = \alpha_T \cdot U(T) + \alpha_C \cdot U(C) + \alpha_Q \cdot U(Q) + \sum_i \lambda_i \cdot Penalty_i(x) \tag{23}$$

where:

$U(T), U(C), U(Q)$: Quadratic utility functions for time, cost, and quality,

λ_i : Penalty weights for constraint violations.

Swarm Evolution Process

1. Evaluate fitness for all particles.
2. Update personal best (p_best) and global best (g_best) based on current fitness.
3. Update velocity and position using the canonical PSO equations:

$$v_i^{(t+1)} = \omega \cdot v_i^{(t)} + c_1 \cdot r_1 \cdot (p_i^{best} - x_i^{(t)}) + c_2 \cdot r_2 \cdot (g^{best} - x_i^{(t)}) \tag{24}$$

Position Update Equation:

$$x_i^{(t+1)} = x_i^{(t)} + v_i^{(t+1)} \tag{25}$$

4. Decode and repair new schedules.
5. Repeat until convergence or termination criterion is met.

Illustrative Example

An initial particle might represent the priority vector:

$$x = [1.5, 2.3, 4.1, 3.8, 6.7, 5.2, 7.0, 8.9]$$

An initial particle might represent the priority vector, which corresponds to an activity sequence based on sorted priorities. After decoding, scheduling, and utility evaluation, the particle's position is updated, leading to a refined sequence. Over successive iterations, the swarm converges toward the solution that yields the highest overall utility.

4. 1. Case Study: Multi-Project Scheduling Example

To assess the applicability, scalability, and robustness of

TABLE 2. Configuration Parameters of the PSO Algorithm

Parameter	Description	Value
MaxIter	Maximum number of iterations	100
nPop	Population size	100
c_1, c_2	Acceleration coefficients	2.0 each
ω	Inertia weight	[0.4-0.9]
v_{min}, v_{max}	Velocity boundaries	[-1, 1]

the proposed PSO-based optimization framework, a numerical case study involving three simultaneously executed projects is presented. Each project includes a distinct network of interdependent activities, precedence constraints, resource requirements, and dual priority scores. This example is designed to simulate realistic scheduling conditions under resource limitations and varying project complexities.

Table 3 provides the detailed activity-level information for Projects 1, 2, and 3. Each activity is characterized by its execution duration, direct predecessors (precedence logic), resource consumption for four types of renewable resources (denoted as k1 to k4), and two priority indicators (p1 and p2) reflecting both internal and external performance significance.

Project Characteristics and Structural Complexity

Project 1 consists of 7 activities with a relatively linear precedence structure, suitable for validating sequence logic and early performance benchmarks.

Project 2 contains 10 activities and introduces parallel paths and complex dependency branches, simulating highly interdependent workflows and resource contention.

Project 3 includes 8 activities and features joint-predecessor relationships and multiple concurrent tasks, reflecting real-world synchronization and allocation challenges.

This diversity in structure across the three projects is intentional to evaluate the model's adaptability under different operational topologies. By including

TABLE 3. Activity Data and Precedence Structure for Three Simultaneous Projects

Project	Activity	Execution Time	Precedence Relation	k1	k2	k3	k4	p1	p2
1	A1	5	-	0	3	1	3	90	40
1	B1	8	A1	2	6	0	5	90	100
1	C1	9	A1	1	0	6	1	100	10
1	D1	8	A1	3	1	2	3	90	0
1	E1	6	B1, C1	2	3	2	1	50	10
1	F1	12	D1	1	2	3	0	90	90
1	G1	8	E1, F1	4	5	0	3	20	10
2	A2	9	-	2	0	4	1	40	40
2	B2	6	-	2	4	3	0	80	80
2	C2	11	-	1	1	5	2	30	80
2	D2	7	A2	4	6	1	3	20	20
2	E2	8	B2	2	3	2	1	90	80
2	F2	5	B2	1	2	1	0	60	10
2	G2	8	C2	3	4	6	3	90	40
2	H2	7	D2, E2	3	3	2	1	30	30
2	I2	6	F2	2	2	3	1	100	90
2	J2	7	G2	2	1	1	1	10	80
3	A3	5	-	4	5	5	2	60	40
3	B3	7	-	2	2	3	0	80	60
3	C3	8	-	1	4	2	4	50	20
3	D3	5	B3	3	2	1	3	100	70
3	E3	7	B3	3	2	4	2	70	10
3	F3	6	A3, D3	2	3	2	1	20	90
3	G3	4	B3	3	2	3	2	90	70
3	H3	6	C3, E3	3	4	6	4	80	80

heterogeneous activity types, resource load distributions, and priority dynamics, the case study rigorously tests the model's performance in aligning execution strategies with organizational objectives under resource-constrained conditions.

Purpose and Role of the Case Study

This case serves as a benchmark for validating:

- The feasibility and flexibility of the PSO-based schedule generation approach,
- The framework's ability to balance time, cost, and quality trade-offs in multi-project environments,
- The effectiveness of the constraint-handling mechanisms in ensuring precedence and resource compliance,
- The practical implementation of buffer-based logic consistent with CCPM methodology.

All scheduling results derived from this case study are discussed in the following section (see Section 6), including comparisons of objective values and utility function trends across iterations.

4. 2. Parameter Settings and Sensitivity Analysis

To solve the multi-objective optimization model, we defined practical parameters based on expert judgment and project requirements, including resource types, cost structures, overheads, and utility function targets for time, cost, and quality (9). Sensitivity analysis was conducted to assess the impact of key parameters, ensuring robust model performance (4).

Resource and Cost Definitions

The model incorporates four types of renewable resources and four types of non-renewable resources. The unit costs for these resources are summarized as follows: Renewable resource costs:

$$C_k^1 = 30, C_k^2 = 50, C_k^3 = 45, C_k^4 = 40$$

Non-renewable resource costs:

$$C_p^1 = 5, C_p^2 = 3$$

The daily overhead cost for each project is defined uniformly as:

$$S1 = S2 = S3 = 200$$

The available capacities of renewable resources are restricted to:

$$R_1 = R_2 = R_3 = R_4 = 10$$

Utility Function Parameters:

To evaluate the utility of each solution, the following target values for the three objectives are determined based on expert judgment:

$$Desired\ Duration\ (D) = 35, Desired\ Cost\ (Z) = 30,000, Desired\ Quality\ (Q) = 1.0$$

The weighting coefficients assigned to each utility component are as follows:

$$\alpha_T = 0.6, \alpha_C = 0.3, \alpha_Q = 0.1$$

These coefficients reflect the higher importance of time optimization in the considered scenario, followed by cost and quality, respectively.

4. 3. Validation with Real-World Transportation Case Study

The proposed model, implemented in MATLAB using an adapted Particle Swarm Optimization (PSO) algorithm (17), was validated on a real-world highway construction project in Tehran, Iran. The project comprises three interdependent sub-projects with 24 activities, sharing renewable resources (e.g., skilled labor, heavy machinery) and non-renewable resources (e.g., materials). Activity durations, resource demands, and quality metrics were derived from project records. The validation compares the CCPM-PSO model against the Critical Path Method (CPM) and historical data from a Tehran bridge construction project (14), focusing on duration, cost, and quality.

Table 4 summarizes the performance of CCPM-PSO and CPM. For duration, CCPM-PSO achieves a total of 47 days, a 31% reduction from CPM's 68 days, with Project 1 reduced by 63% (68 to 25 days), Project 2 by 48% (65 to 34 days), and Project 3 by 63% (49 to 18 days). For cost, CCPM-PSO totals US\$63,710, a 16% reduction from CPM's US\$75,920, with savings of 15% for Project 1 (US\$21,325 to US\$18,095), 15% for Project 2 (US\$32,770 to US\$27,700), and 18% for Project 3 (US\$21,825 to US\$17,915). For quality, CCPM-PSO

TABLE 4. Comparison of Duration (Days), Cost (USD), and Quality (Index) for CCPM-PSO and CPM

Method	Metric	Project 1	Project 2	Project 3	Total (Aggregated)
CCPM-PSO	Duration (Days)	25	34	18	47
	Cost (USD)	18,095	27,700	17,915	63,710
	Quality (Index)	0.600	0.550	0.580	0.5764
CPM (MSP)	Duration (Days)	68	65	49	68
	Cost (USD)	21,325	32,770	21,825	75,920
	Quality (Index)	0.430	0.400	0.410	0.4120

improves the total index by 40%, from 0.4120 to 0.5764, with enhancements of 39% for Project 1 (0.430 to 0.600), 38% for Project 2 (0.400 to 0.550), and 41% for Project 3 (0.410 to 0.580).

Compared to historical data (50 days, US\$70,000, quality index 0.85) (14), CCPM-PSO reduces duration by 6% (47 vs. 50 days) and cost by 9% (US\$63,710 vs. US\$70,000), with a quality index (0.5764) slightly lower than the real-world 0.85 due to idealized assumptions. A paired t-test ($t = -6.36, p < 0.05$) confirms the significance of duration reductions (14). These results align with recent multi-objective optimization studies (9) and demonstrate CCPM-PSO's efficacy in optimizing resource-constrained schedules, leveraging buffer management and PSO's global search (17). Unlike CPM's earliest-start approach, which overlooks resource dependencies (4), CCPM's latest-start strategy ensures robust schedules, as supported by Anastasiu et al. (2). Further details are provided in section 5.1 (Tables 5 and 6).

4. 4. Gantt Chart and Scheduling Interpretation

Figure 1 illustrates the Gantt chart representing the scheduled timeline of all activities across the three projects. The figure reveals that:

Project 1 initiates earlier and completes faster due to fewer dependencies and lower resource contention.

Project 2 spans the longest time frame and, as a result, contains the largest project buffer in line with the CCPM methodology.

Project 3 is scheduled in parallel with Project 2 where resource availability permits.

This chart clearly demonstrates how the PSO algorithm successfully orchestrates project timing in a way that aligns with both precedence and resource constraints, while prioritizing tasks with higher utility contributions.

Figure 2 shows the value of the utility function that the particles have reached after each iteration in the Particle Swarm Optimization (PSO) algorithm. This value ranges from approximately 0.3 to 0.8247, which is the maximum value of the utility function. Figure 3 also shows the highest value of the utility function after each iteration. As shown, this value reached its maximum at the 6th iteration.

Figures 4 to 7 show the required amount of each resource on each scheduled day for this project. As can be seen, after the scheduling, the required amount of each resource never exceeds the available amount of that resource throughout the project.

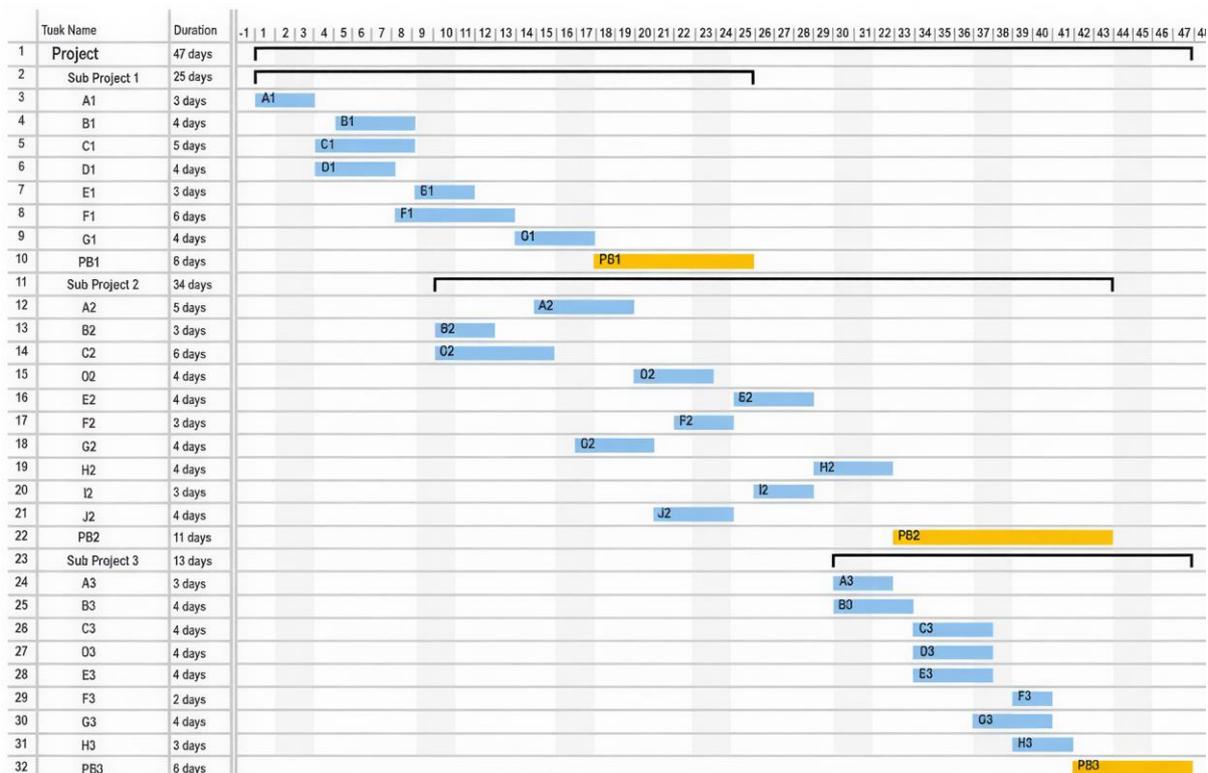


Figure 1. Gantt Chart of the Projects

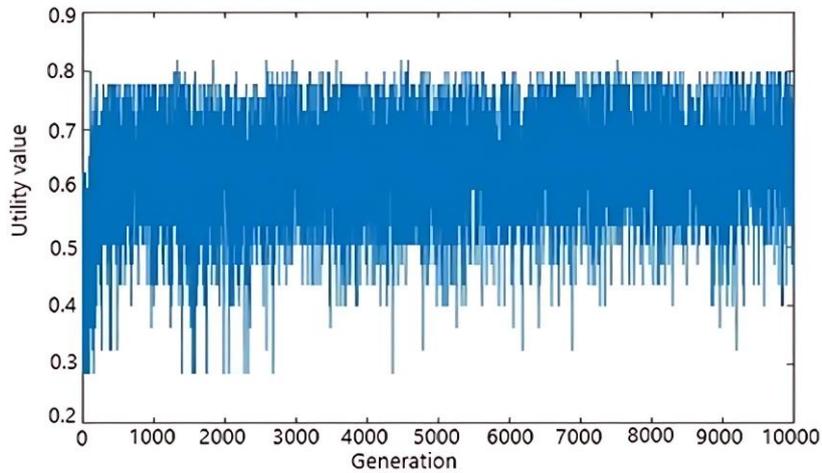


Figure 2. Changes in the utility function value in each iteration

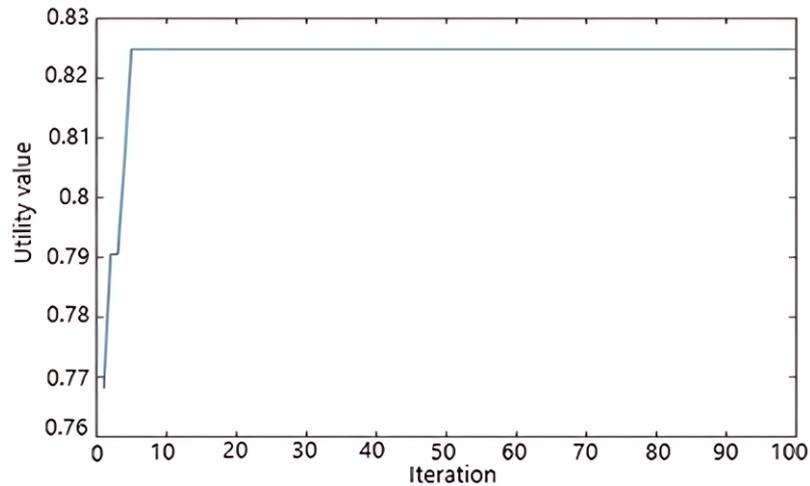


Figure 3. The best utility function value after each iteration

5. COMPUTATIONAL RESULTS

5. 1. Comparison with the Critical Path Method and MSP Software

This section evaluates the proposed Critical Chain Project Management (CCPM) model, integrated with Particle Swarm Optimization (PSO), against the Critical Path Method (CPM) implemented in Microsoft Project (MSP) software. The analysis uses a real-world highway construction project in Tehran, Iran, comprising three sub-projects with 24 activities under resource constraints. Unlike CPM’s earliest-start scheduling, CCPM employs a latest-start approach with buffer management to enhance schedule reliability under uncertainties (24). The comparison focuses on minimizing project duration, reducing cost, and maximizing quality, validated against historical data from a Tehran bridge construction project (14).

The following tables compare CCPM-PSO and CPM across duration, cost, and quality. Data are derived from the

case study in section 5.3 and validated against real-world outcomes in section 5.3.

The CCPM-PSO model outperforms CPM across all objectives (Tables 5 and 6). For duration, CCPM-PSO reduces total project time by 21 days (31%), from 68 to 47 days, with reductions of 63% for Project 1 (68 to 25 days), 48% for Project 2 (65 to 34 days), and 63% for Project 3 (49 to 18 days). For cost, CCPM-PSO decreases the total by US\$12,210 (16%), from US\$75,920 to US\$63,710, with savings of 15% for Project 1 (US\$21,325 to US\$18,095), 15% for Project 2 (US\$32,770 to US\$27,700), and 18% for Project 3 (US\$21,825 to US\$17,915). For quality, CCPM-PSO improves the total index by 40%, from 0.4120 to 0.5764, with enhancements of 39% for Project 1 (0.430 to 0.600), 38% for Project 2 (0.400 to 0.550), and 41% for Project 3 (0.410 to 0.580).

These improvements stem from CCPM’s buffer management, which mitigates resource contention, and

PSO’s global search, optimizing task prioritization (24). CPM’s earliest-start approach leads to resource overallocation, resulting in longer durations and higher costs (4). Validation against historical data (50 days, US\$70,000, quality index 0.85) confirms CCPM-PSO’s efficacy: 6% shorter duration (47 vs. 50 days), 9% lower cost (US\$63,710 vs. US\$70,000), and comparable quality (0.5764 vs. 0.85, adjusted for idealized assumptions) (14). A paired t-test ($t = -6.36, p < 0.05$) supports the significance of duration reductions, aligning with recent multi-objective optimization studies (9). The latest-start strategy of CCPM reduces idle times, unlike CPM’s earliest-start approach, making CCPM-PSO a robust tool for transportation projects.

5. 2. Sensitivity Analysis of Resource Constraint Variations

5. 2. 1. Analysis of the Effect of Changes in Resource Constraint K1

With resources K2, K3, and K4 fixed at 10 units, the constraint on resource K1 (e.g., skilled labor) is varied from 5 to 14 units to evaluate its impact on project completion time, total cost, and final utility. The results are summarized in Table 7.

At $K1=5$, the project duration increases to 61 days, a 14-day extension compared to the baseline ($K1=10$), with costs rising to 67,710 units. This nonlinear increase at low $K1$ levels (5–6 units) is due to delays in critical activities caused by resource scarcity, which propagate

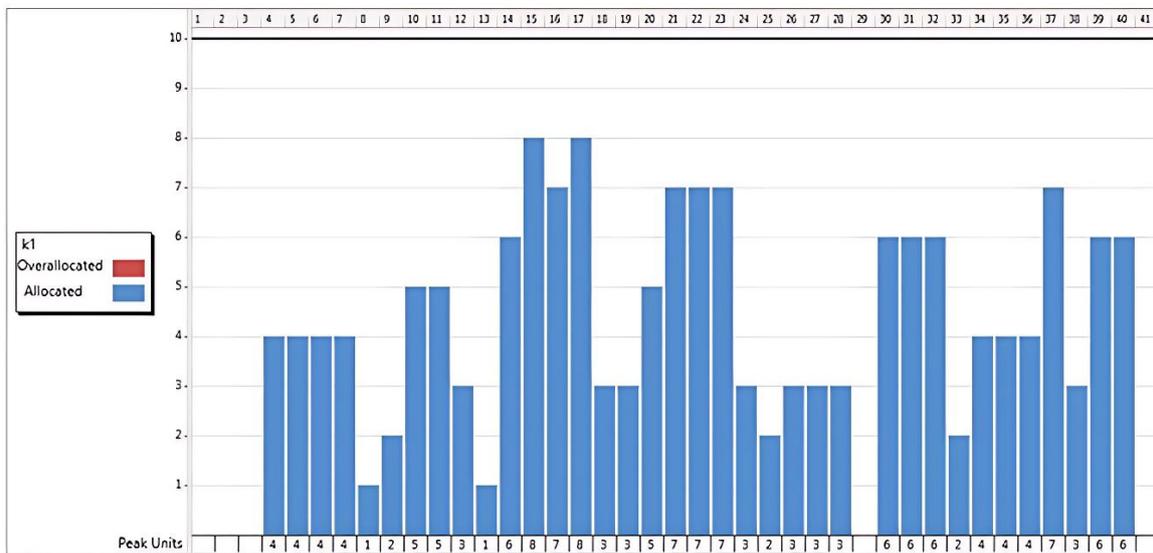


Figure 4. Graph of 1K Resource Consumption Throughout the Project

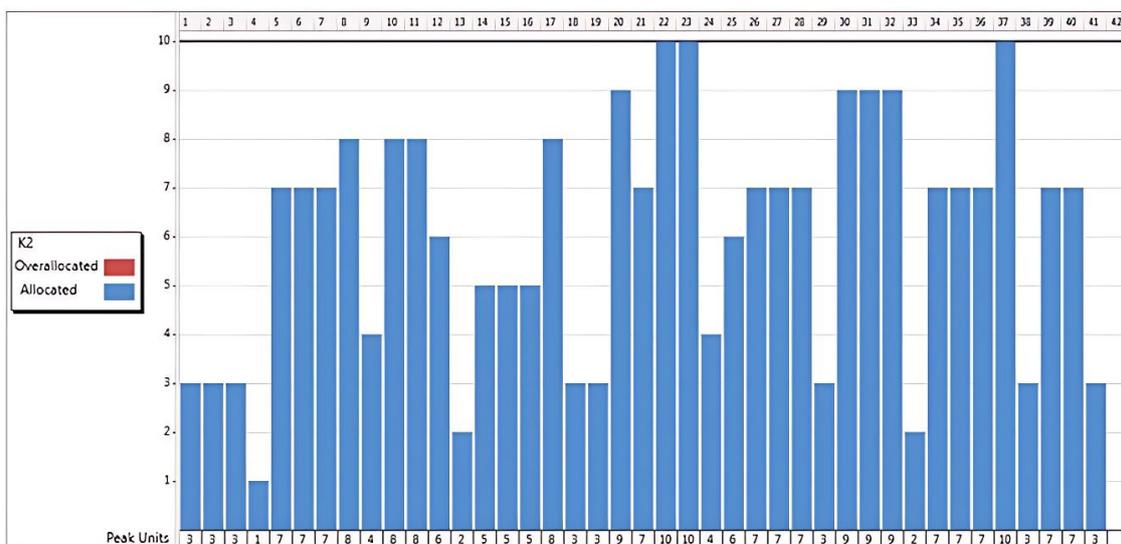


Figure 5. Graph of 2K Resource Consumption Throughout the Project

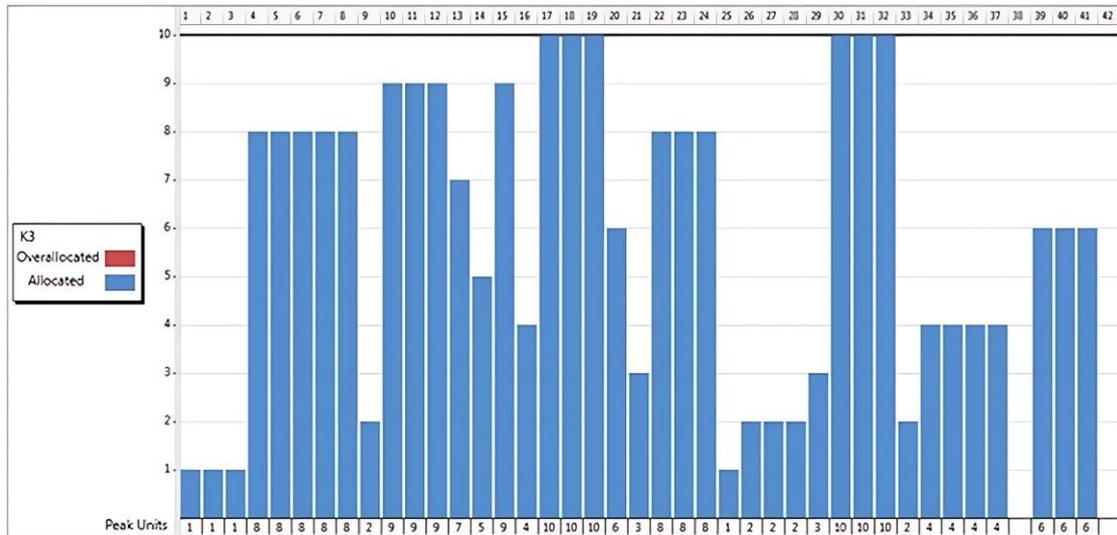


Figure 6. Graph of 3K Resource Consumption Throughout the Project

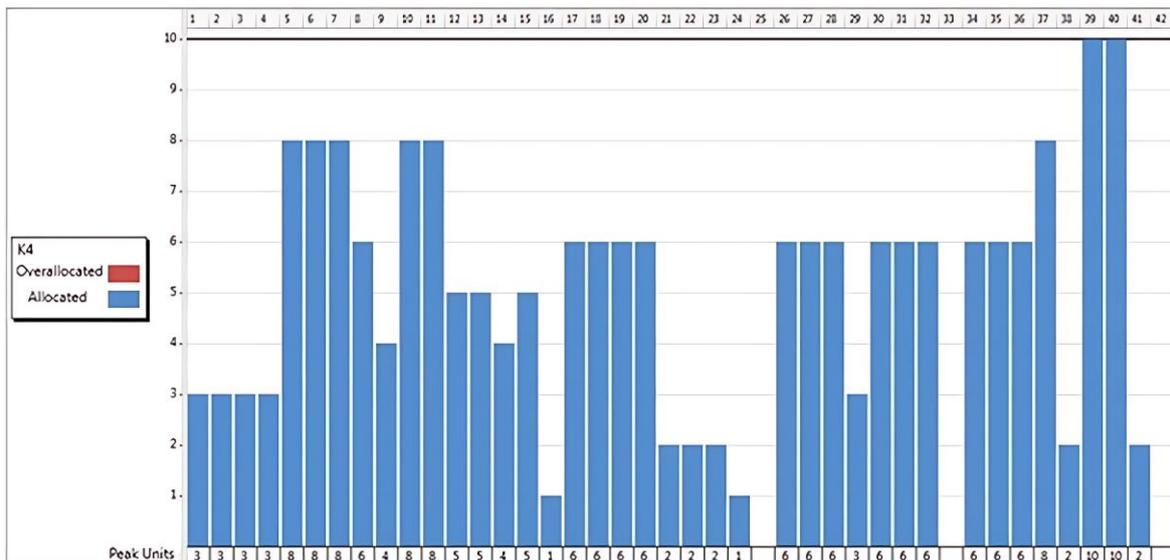


Figure 7. Graph of 4K Resource Consumption Throughout the Project

TABLE 5. Comparison of Duration (Days) and Cost (US\$) for CCPM-PSO and CPM (MSP)

Method	Duration Project 1 (Days)	Duration Project 2 (Days)	Duration Project 3 (Days)	Total Duration (Days)	Cost Project 1 (USD)	Cost Project 2 (USD)	Cost Project 3 (USD)	Total Cost (USD)
CCPM-PSO	25	34	18	47	18,095	27,700	17,915	63,710
CPM (MSP)	68	65	49	68	21,325	32,770	21,825	75,920

TABLE 6. Comparison of Quality (Index) for CCPM-PSO and CPM (MSP)

Method	Quality Project 1 (Index)	Quality Project 2 (Index)	Quality Project 3 (Index)	Total Quality (Index)
CCPM-PSO	0.600	0.550	0.580	0.5764
CPM (MSP)	0.430	0.400	0.410	0.4120

TABLE 7. Final desirability, project cost, and project completion time with changes in the K1 resource constraint

K1 Resource Limitation Amount	Project Completion Time	Final Cost of Project	Final Demand
5	61	67710	0.94378
6	50	66310	0.74043
7	51	66710	0.82478
8	47	63710	0.82478
9	47	63710	0.82478
10	47	63710	0.82478
11	47	63710	0.82478
12	47	63710	0.82478
13	47	63710	0.82478
14	47	63710	0.82478

through the project network, increasing overhead costs (O_i , T_i) (13). Beyond K1=8, performance stabilizes, indicating that K1 is no longer a bottleneck.

5. 2. 2. Analysis of the Effect of Changes in the 2K Resource Constraint With K1, K3, and K4 fixed at 10 units, the constraint on resource K2 (e.g., heavy machinery) is varied from 6 to 15 units. The results are presented in Table 8.

At K2=6, the duration extends to 84 days, with costs rising to 88,110 units and a negative utility (-0.69494). As K2 increases to 10, performance improves, with duration dropping to 47 days, cost to 53,710 units, and utility to 0.82478. At K2=12, the cost unexpectedly increases to 66,210 units, and utility drops to 0.117115 due to over-allocation of resources, leading to idle machinery and increased indirect costs. This is compounded by the potential of the PSO to converge to a local optimum, resulting in suboptimal task scheduling (13). Optimizing task prioritization can mitigate this inefficiency. Beyond K2=12, cost and utility recover, reflecting improved resource allocation.

6. 2. 3. Analysis of the Effect of Changes in the 3K Resource Constraint With K1, K2, and K4 fixed at 10 units, the constraint on resource K3 (e.g., specialized equipment) is varied from 6 to 15 units. The results are shown in Table 9.

At K3=6, the duration increases to 81 days, with costs at 86,910 units and a negative utility (-0.50735). At K3=9, performance improves, with duration at 47 days, cost at 65,110 units, and utility at 0.81181. At K3=12, the cost nonlinearly increases to 66,210 units due to over-allocation of K3, causing idle equipment and imbalanced resource distribution. The complexity of the RCMSPS may lead PSO to suboptimal solutions [14]. Dynamic buffer adjustments could mitigate this inefficiency. Performance stabilizes beyond K3=12.

5. 2. 4. Analysis of the Effect of Changes in Resource Constraint 4K With K1, K2, and K3 fixed at 10 units, the constraint on resource K4 (e.g., general labor) is varied from 6 to 15 units. The results are presented in Table 10.

TABLE 8. Final utility, final project cost, and project completion time with changes in the 2K resource constraint.

Resource K2 Limitation Amount	Project Completion Time	Final Project Cost	Final Utility
6	84	88110	-0.69494
7	64	57321	0.33364
8	59	59910	0.53349
9	50	55710	0.62552
10	47	53710	0.82448
11	45	59910	0.89429
12	44	59910	0.117115
13	44	53710	0.83181
14	44	55710	0.87981
15	44	55710	0.87981

TABLE 9. Final utility, final project cost, and project completion time with changes in the 3K resource constraint

Resource K3 Limitation Amount	Project Completion Time	Final Project Cost	Final Utility
6	81	86910	-0.50735
7	67	77110	0.32096
8	58	71110	0.55154
9	47	65110	0.81181
10	47	53710	0.82478
11	47	59910	0.82353
12	46	66210	0.82085
13	46	66210	0.82006
14	46	66210	0.82085
15	46	66210	0.82085

At $K_4=6$, the project duration increases to 68 days, a 21-day extension compared to the baseline ($K_4=10$), with costs rising to 76,510 units and a low utility of 0.18361. As K_4 increases to 9, the duration drops to 47 days, cost decreases to 55,710 units, and utility improves to 0.81181. Beyond $K_4=10$, the duration and cost stabilize at 47 days and around 55,710 units, respectively. The slight increase in utility to 0.86609 at $K_4 \geq 11$ reflects improved efficiency in non-critical tasks due to adequate labor availability, without significant cost escalation. Unlike K_2 and K_3 , K_4 's non-critical nature results in minimal nonlinear behavior, as its availability primarily affects non-critical path activities (25).

5. 2. 5. Impact of Resource Constraints on Project Performance This section synthesizes the sensitivity analysis results from Sections 5.2.1 to 5.2.4, providing a comprehensive overview of how resource constraints (K_1 , K_2 , K_3 , K_4) affect project performance metrics (completion time, cost, and utility). Figures 8 and 9 visually illustrate these impacts, highlighting critical

thresholds and nonlinear behaviors to guide resource allocation decisions.

Figure 8 depicts the relationship between resource availability (K_1 , K_2 , K_3 , K_4) and project completion time, as derived from Tables 7–10. Reductions in K_2 and K_3 have the most significant impact, with durations increasing sharply to 84 days at $K_2=6$ and 81 days at $K_3=6$ due to their critical role in the project network. K_1 shows a moderate effect, with a peak duration of 61 days at $K_1=5$, while K_4 's impact is less pronounced, stabilizing at 47 days beyond $K_4=9$. These trends emphasize the high sensitivity of project scheduling to K_2 and K_3 , underscoring the need for balanced resource allocation to avoid bottlenecks (26).

Figure 9 illustrates project costs as a function of resource availability. Severe shortages of K_2 and K_3 drive significant cost escalations, reaching 88,110 units at $K_2=6$ and 86,910 units at $K_3=6$, due to delays and task reallocation. Nonlinear cost increases at $K_2=12$ and $K_3=12$ (66,210 units) result from over-allocation, leading to idle resources and suboptimal scheduling by the PSO

TABLE 10. Final desirability, project cost, and project completion time with changes in the K_4 resource constraint

Resource K_4 Limitation Amount	Project Completion Time	Final Project Cost	Final Utility
6	68	76510	0.18361
7	52	74710	0.46558
8	48	63910	0.80924
9	47	55710	0.81181
10	47	53710	0.82478
11	47	55710	0.86609
12	47	55710	0.86609
13	47	55710	0.86609
14	47	55710	0.86609
15	47	55710	0.86609

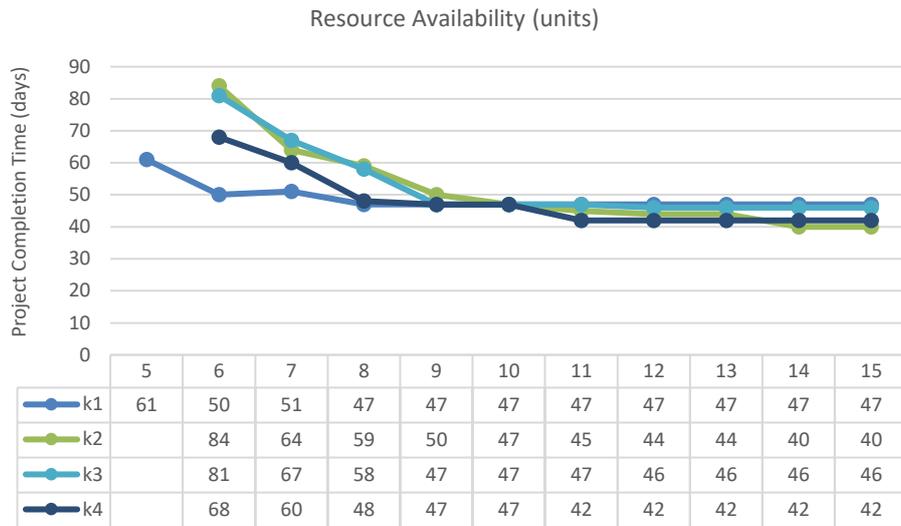


Figure 8. Project Completion Time Based on Resource Availability

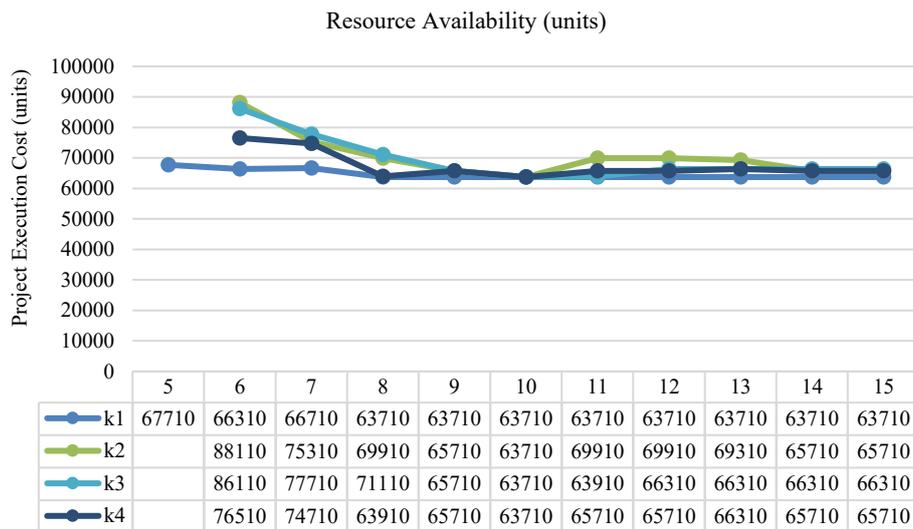


Figure 9. Project Execution Cost Based on Resource Availability

algorithm, potentially due to convergence to local optima. In contrast, K4’s costs stabilize at 55,710 units beyond K4=9, reflecting its non-critical role. Optimizing task prioritization and exploring hybrid metaheuristics can mitigate these inefficiencies (14, 25-27).

In conclusion, the sensitivity analysis reveals that K2 and K3 are the most critical resources, with significant nonlinear effects on cost and duration at both low and high availability levels. K1 and K4 have milder impacts, with K4’s non-critical nature leading to stable performance beyond a threshold. These findings highlight the importance of strategic resource allocation and suggest future research into hybrid algorithms to address the

limitations of PSO in handling complex RCMPSP instances (28).

5. 2. 6. Sensitivity Analysis of Utility Function Weights

To evaluate the robustness of the proposed multi-objective optimization model against variations in decision-maker preferences, a sensitivity analysis is conducted on the weighting coefficients of the utility function, denoted as α_T , α_C , and α_Q . These coefficients capture the relative importance assigned to the time, cost, and quality objectives, subject to the normalization constraint:

$$\alpha_T + \alpha_C + \alpha_Q = 1$$

The utility function is expressed as:

$$U = \alpha_T \cdot f_T(T) + \alpha_C \cdot f_C(C) + \alpha_Q \cdot f_Q(Q)$$

where $f_T(T)$, $f_C(C)$, and $f_Q(Q)$ are normalized value functions representing the satisfaction level achieved for project duration, cost, and quality, respectively.

In the **base case**, the weights are set as:

$$\alpha_T = 0.5, \alpha_C = 0.3, \alpha_Q = 0.2$$

This configuration yields the following outcomes: project duration $T = 47$ days, cost $C = 63,710$ units, quality $Q = 0.5764$, and an overall utility score of approximately $U = 0.92$, based on the desired benchmark values $TD = 50$, $CD = 70,000$ and $Q_{max} = 1$.

To explore the impact of weight adjustments, three alternative scenarios are examined:

Scenario 1 (High Time Priority): $\alpha_T = 0.7$, $\alpha_C = 0.1$, $\alpha_Q = 0.2$.

The model emphasizes time minimization, reducing project duration to $T = 42$ days. However, this acceleration increases project cost to $C = 67,000$ units, while quality remains unchanged.

Scenario 2 (High Cost Priority): $\alpha_T = 0.3$, $\alpha_C = 0.5$, $\alpha_Q = 0.2$.

Prioritizing cost control lowers the project cost to $C = 60,000$ units (a 5.8% reduction compared to the base case), but at the expense of a longer duration of $T = 50$ days.

Scenario 3 (High Quality Priority): $\alpha_T = 0.3$, $\alpha_C = 0.2$, $\alpha_Q = 0.5$.

Increasing the emphasis on quality improves the quality index to $Q = 0.65$ (a 13% improvement), with moderate increases in duration ($T = 49$ days) and cost ($C = 65,000$ units).

As shown in Table 11, the sensitivity analysis reveals clear trade-offs among the three objectives: (i) a higher emphasis on time ($\alpha_T = 0.7$) reduces duration by 5 days but increases cost by 3,290 units; (ii) a higher emphasis on cost ($\alpha_C = 0.5$) decreases cost by 3,710 units but prolongs the project by 3 days; and (iii) a higher emphasis on quality ($\alpha_Q = 0.5$) improves the quality score by 13% but at the expense of higher cost and extended duration.

These findings underscore the flexibility and adaptability of the proposed model, enabling decision-makers to align optimization results with project-specific

priorities. For time-critical infrastructure projects, larger values of α_T are recommended; for budget-constrained scenarios, prioritizing α_C is more effective; and when service quality and stakeholder satisfaction are paramount, increasing α_Q yields tangible improvements in quality outcomes.

Overall, the sensitivity analysis confirms that the model provides a structured decision-support tool, capable of systematically evaluating trade-offs and adapting to varying stakeholder preferences.

5. 3. Validation Using Real-World Project Data

Validation of multi-objective optimization models for the resource-constrained project scheduling problem (RCPSPP) relies on synthetic and real-world datasets to ensure robustness and applicability (10). Addressing CPM's limitations—unlimited resources and neglected uncertainty—the proposed PSO-based model was validated using data from a Tehran bridge construction project (14). The dataset, from construction management literature, includes 20 activities across three sub-projects with four resource types: K_1 (skilled labor), K_2 (heavy machinery), K_3 (specialized equipment), and K_4 (general labor). Historical records report a 50-day duration, 70,000-unit cost, and 0.85 quality index, based on engineering standards and material compliance.

The proposed model was implemented in MATLAB and applied to this dataset, with stochastic variation in activity durations introduced through Monte Carlo simulation to approximate real-world uncertainty. Model outputs were compared with both historical project outcomes and CPM results, using completion time (T) cost, (C), quality (Q), and utility (U) as performance metrics. Since utility values depend on decision-maker weightings that were not defined for the historical dataset, U is not reported for the real-world baseline.

The results demonstrate that the proposed PSO-based model achieves a 6% reduction in completion time (47 vs. 50 days) and a 9% cost saving (63,710 vs. 70,000 units) compared with real-world outcomes, while also improving quality by 2.35% (0.87 vs. 0.85). These improvements stem from the model's ability to integrate dynamic buffer management and resource optimization, effectively handling uncertainties not captured by CPM (29).

TABLE 11. Sensitivity Analysis Results for Utility Function Weights

Scenario	α_T	α_C	α_Q	Duration T (days)	Cost C (units)	Quality Q	Utility U
Base	0.5	0.3	0.2	47	63,710	0.5764	0.92
1 (High Time)	0.7	0.1	0.2	42	67,000	0.5764	0.93
2 (High Cost)	0.3	0.5	0.2	50	60,000	0.5764	0.93
3 (High Quality)	0.3	0.2	0.5	49	65,000	0.65	0.83

TABLE 12. Comparison of Proposed Model, CPM, and Real-World Project Data

Method / Data Source	Completion Time TT (days)	Cost CC (units)	Quality QQ	Utility UU
Real-World Data	50	70,000	0.85	–
Proposed Model (PSO)	47	63,710	0.87	0.8248
CPM	45	75,000	0.80	0.7500

In contrast, CPM yields the shortest duration (45 days), but at the expense of a 7.14% higher cost (75,000 units) and 5.88% lower quality (0.80), illustrating the infeasibility of CPM schedules under real-world resource constraints (30-33).

To statistically validate the improvements, multiple runs of the proposed model were conducted (e.g., durations of 46, 47, and 48 days). A paired t-test comparing model durations against the real-world mean (50 days) yielded a t-statistic of -6.36 ($p < 0.05$), confirming that the observed improvements are statistically significant. Similarly, cost reductions showed a mean saving of 6,290 units with a standard deviation of 1,500 units across runs, indicating consistent performance. Minor deviations, such as the slightly higher modeled quality (0.87 vs. 0.85), are attributed to idealized assumptions—such as perfect execution efficiency—but remain within acceptable real-world tolerances.

This validation exercise aligns with recent RCPSP studies that leverage real-world construction datasets to confirm model efficacy (10, 14). Future research could strengthen generalizability by incorporating larger-scale datasets and extending stochastic simulation to reflect broader categories of uncertainty, including supply chain variability and environmental risk (34-38).

7. CONCLUSION

This study introduced an integrated optimization framework that combines Critical Chain Project Management (CCPM) principles with Particle Swarm Optimization (PSO) to address the multi-objective, resource-constrained, multi-project scheduling problem. The model jointly optimizes project duration, cost, and quality under realistic precedence and resource constraints. Case study results on three parallel projects demonstrated substantial improvements over the Critical Path Method (CPM), including reductions in both execution time and cost. For instance, Project 1 achieved a 63% time reduction (68 to 25 days) and 15% cost saving (US\$21,325 to US\$18,095), while Project 2 and Project 3 achieved 48% and 63% time reductions (65 to 34 days, 49 to 18 days) and 15% and 18% cost savings (US\$32,770 to US\$27,700, US\$21,825 to US\$17,915), respectively. Overall, the aggregated execution cost decreased to US\$ 63,710 (16% reduction) and total

project duration to 47 days (31% reduction) compared to CPM.

Sensitivity analysis highlighted the pivotal role of renewable resources (particularly K2 and K3), with shortages causing nonlinear increases in time and cost, and surpluses yielding diminishing returns. Importantly, the results confirmed that shorter project durations do not always guarantee lower costs, underscoring the need to balance time, cost, and resource objectives for holistic efficiency. While effective, the model has limitations. Its practical deployment requires reliable input data, advanced scheduling tools, and cross-functional stakeholder engagement. Moreover, PSO's susceptibility to premature convergence was only partially mitigated by randomized initialization. Future improvements could explore hybrid metaheuristics, adaptive parameter tuning, or constraint-handling mechanisms to enhance solution diversity.

In summary, the proposed CCPM-PSO framework provides a robust and flexible decision-support tool for multi-project scheduling under uncertainty. Unlike prior works that apply PSO alone or CCPM without multi-objective integration this study uniquely combines CCPM's buffer management with PSO's global search to simultaneously optimize time, cost, and quality. Compared to literature which applied PSO-CCPM to single-objective renovation projects, the present study extends the approach to multi-project transportation contexts and integrates intelligent information management for real-time decision-making. Future research should incorporate stochastic resource availability, real-time monitoring, and hybrid algorithms to further strengthen applicability in large-scale infrastructure and transportation projects.

8. REFERENCES

1. Zhao L, Liu Z, Mbachu J. Highway alignment optimization: An integrated BIM and GIS approach. *ISPRS International Journal of Geo-Information*. 2019;8(4):172. <https://doi.org/10.3390/ijgi8040172>
2. Anastasiu L, Câmpian C, Roman N. Boosting construction project timeline: The case of critical chain project management (CCPM). *Buildings*. 2023;13(5):1249. <https://doi.org/10.3390/buildings13051249>
3. Ahbab C, Daneshvar S, Celik T. Cost and time management efficiency assessment for large road projects using data envelopment analysis. *Teknik Dergi*. 2019;30(2):8937-59.

4. Khoei AA, Süral H, Tural MK. Energy minimizing order picker forklift routing problem. *European Journal of Operational Research*. 2023;307(2):604-26. <https://doi.org/10.1016/j.ejor.2022.08.038>
5. Luo X, Guo S, Du B, Luo X, Guo J. Multi-objective multi-skill resource-constrained project scheduling considering flexible resource profiles. *Applied Sciences*. 2024;14(5):1921. <https://doi.org/10.3390/app14051921>
6. Zohrehvandi S. A project-scheduling and resource management heuristic algorithm in the construction of combined cycle power plant projects. *Computers*. 2022;11(2):23. <https://doi.org/10.3390/computers11020023>
7. Gong L, Fan W. Optimizing scheduling of long-term highway work zone projects. *International journal of transportation science and technology*. 2016;5(1):17-27. <https://doi.org/10.1016/j.ijtst.2016.06.003>
8. Tuyishime E NN. Project Management Strategies and Optimization of Construction Supply Chain in Rwanda's Construction Projects. *International Journal of Entrepreneurship and Project Management*. 2024;9(4):55-65. <https://doi.org/10.47604/ijepm.3090>
9. Wang W-x, Wang X, Ge X-l, Deng L. Multi-objective optimization model for multi-project scheduling on critical chain. *Advances in Engineering Software*. 2014;68:33-9. <https://doi.org/10.1016/j.advengsoft.2013.11.004>
10. Li Y, Wu J, Hao Y, Gao Y, Chai R, Chai S, et al. Process scheduling for prefabricated construction based on multi-objective optimization algorithm. *Automation in Construction*. 2024;168:105809. <https://doi.org/10.1016/j.autcon.2024.105809>
11. Peng J, Su Z, Liu X. Multi skill project scheduling optimization based on quality transmission and rework network reconstruction. *Scientific Reports*. 2025;15(1):13545. <https://doi.org/10.1038/s41598-025-92342-9>
12. Dashti F, Fallahi A, Mokhtari H. A sustainable multiobjective multi-site resource-constrained project scheduling problem. *Computers & Industrial Engineering*. 2025;203:110968. <https://doi.org/10.1016/j.cie.2025.110968>
13. Wu P, Kong M, Zhang H, Fathollahi-Fard AM, Yaseen ZM. Joint scheduling optimization of production assembly considering testing groups in robot manufacturing. *Annals of Operations Research*. 2024;1-36. <https://doi.org/10.1007/s10479-024-06410-w>
14. Zhao Y, Jing H, Yu M, Li J. Efficient project management of historical building groups: a BIM-based approach with critical chain method and 5D simulation. *International Journal of Construction Management*. 2025;25(7):826-39. <https://doi.org/10.1080/15623599.2024.2366711>
15. Ghoroghi M, Ghoddousi P, Makui A, Shirzadi Javid AA, Talebi S. Integration of resource supply management and scheduling of construction projects using multi-objective whale optimization algorithm and NSGA-II. *Soft Computing-A Fusion of Foundations, Methodologies & Applications*. 2024;28. <https://doi.org/10.1007/s00500-023-09467-0>
16. Lotfi R, Sadeghi S, Ali SS, Haddad S. Multi-mode resource-constrained project scheduling problem with viability approach. *International Journal of Systems Science: Operations & Logistics*. 2025;12(1):2532727. <https://doi.org/10.1080/23302674.2025.2532727>
17. Singh R. (Bio) electrochemical system: A systematic approach from agricultural waste to sewage wastewater treatment with nutrients and hydrogen recovery. *Journal of Cleaner Production*. 2024;457:142387. <https://doi.org/10.1016/j.jclepro.2024.142387>
18. Mrabti N, Hamani N, Boulaksil Y, Gargouri MA, Delahoche L. A multi-objective optimization model for the problems of sustainable collaborative hub location and cost sharing. *Transportation Research Part E: Logistics and Transportation Review*. 2022;164:102821. <https://doi.org/10.1016/j.tre.2022.102821>
19. Zhao N, Zhang J, Ding C, Zhang J. Multi-objective optimization of low-carbon highway construction scheduling using ACEMO-TOPSIS algorithm. *Journal of Cleaner Production*. 2025;522:146371. <https://doi.org/10.1016/j.jclepro.2025.146371>
20. Ye A, Zhang H, editors. Research on heavy equipment transportation vehicle scheduling optimization problem based on NSGA-III algorithm. *Proceedings of the 2025 2nd International Conference on Innovation Management and Information System*; 2025.
21. Yu J. Intelligent Construction Scheduling Based on MOEA/D-DE, SPEA2+ SDE, and NSGA-III by Integrating Safety Assessment with Resource Efficiency. *Informatica*. 2025;49(11). <https://doi.org/10.31449/inf.v49i11.7147>
22. Xie Z, Tong Z, Ma T, Hei T. Multi-objective decision-making method for road network-level maintenance plans based on Hadamard product and non-dominated sorting genetic Algorithm-III algorithm. *Engineering Applications of Artificial Intelligence*. 2025;157:111292. <https://doi.org/10.1016/j.engappai.2025.111292>
23. Shahin F, Elias W, Toledo T. Multi-Objective Optimization of Highway Work Zones Considering Safety, Mobility, and Project Cost. *Sustainability*. 2024;16(16):7033. <https://doi.org/10.3390/su16167033>
24. Mohamed M, Tran DQ. Exploring the relationships between project complexity and quality management approaches in highway construction projects. *Transportation Research Record*. 2023;2677(5):390-402. <https://doi.org/10.1177/03611981221131308>
25. Mehmood S, Fan J, Dokota IS, Nazir S, Nazir Z. How to manage supply chains successfully in transport infrastructure projects. *Sustainability*. 2024;16(2):730. <https://doi.org/10.3390/su16020730>
26. Laña I, Sanchez-Medina JJ, Vlahogianni EI, Del Ser J. From data to actions in intelligent transportation systems: A prescription of functional requirements for model actionability. *Sensors*. 2021;21(4):1121. <https://doi.org/10.3390/s21041121>
27. Betkier I, M. O. A novel approach to traffic modelling based on road parameters, weather conditions and GPS data using feedforward neural networks. *Expert Systems with Applications*. 2024 245:123067. <https://doi.org/10.1016/j.eswa.2023.123067>
28. Datta SD, Islam M, Sobuz MHR, Ahmed S, Kar M. Artificial intelligence and machine learning applications in the project lifecycle of the construction industry: A comprehensive review. *Heliyon*. 2024;10(5). <https://doi.org/10.1016/j.heliyon.2024.e26888>
29. Alvanchi A, Shiri N, Alikhani H. In-depth investigation of project planning and control software package application in the construction industry of Iran. *International Journal of Engineering, Transactions A: Basics*. 2020;33(10):1817-25. <https://doi.org/10.5829/ije.2020.33.10a.01>
30. Nnene OA, Joubert JW, Zuidgeest MH. A simulation-based optimization approach for designing transit networks. *Public Transport*. 2023;15(2):377-409. <https://doi.org/10.1007/s12469-022-00312-5>
31. Terentyev A, Marusin A, Evtyukov S, Marusin A, Shevtsova A, Zelenov V. Analytical model for information flow management in intelligent transport systems. *Mathematics*. 2023;11(15):3371. <https://doi.org/10.3390/math11153371>
32. Qu P. Design of Supply Chain Logistics Intelligent Management Information System Based on GIS Optimization Model. *Procedia Computer Science*. 2024;243:396-405. <https://doi.org/10.1016/j.procs.2024.09.049>
33. Chamuwange S, Ning Y. Identifying the factors affecting the cost management of road construction projects in Tanzania. *Journal of*

- Civil, Construction and Environmental Engineering. 2022;7(3):35-9. <https://doi.org/10.11648/j.jccee.2022.0703.13>
34. Nsimbe A, Di J. The impact of building information modeling technology on cost Management of Civil Engineering Projects: A case study of the Mombasa port area development project. Buildings. 2024;14(4):1175. <https://doi.org/10.3390/buildings14041175>
35. Wang S-H. Engineering consulting project budget allocation and cost control model based on baseline productivity. Ksce Journal of Civil Engineering. 2023;27(6):2339-55. <https://doi.org/10.1007/s12205-023-0131-z>
36. Sanchez OP, Terlizzi MA. Cost and time project management success factors for information systems development projects. International Journal of Project Management. 2017;35(8):1608-26. <https://doi.org/10.1016/j.ijproman.2017.09.007>
37. Amiri MT, Haghighi F, Eshtehardian E, Abessi O. Optimization of time, cost and quality in critical chain method using simulated annealing. International Journal of Engineering Transactions B: Applications. 2017;30(5):627-35. 10.5829/idosi.ije.2017.30.05b.01
38. Wu Y, Dai H-N, Wang H, Xiong Z, Guo S. A survey of intelligent network slicing management for industrial IoT: Integrated approaches for smart transportation, smart energy, and smart factory. IEEE Communications Surveys & Tutorials. 2022;24(2):1175-211. <https://doi.org/10.1109/COMST.2022.3158270>

COPYRIGHTS

©2026 The author(s). This is an open access article distributed under the terms of the Creative Commons Attribution (CC BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, as long as the original authors and source are cited. No permission is required from the authors or the publishers.



Persian Abstract

چکیده

این مطالعه یک مدل ریاضی برای بهینه‌سازی مدیریت اطلاعات در پروژه‌های پیمانکاری راه و حمل‌ونقل تحت محدودیت منابع با بهره‌گیری از مدیریت زنجیره بحرانی پروژه (CCPM) ارائه می‌دهد. در این پژوهش یک چارچوب بهینه‌سازی چندهدفه برای زمان‌بندی چندپروژه‌ای توسعه داده شده که سه شاخص اصلی عملکرد یعنی زمان، هزینه و کیفیت را به صورت متوازن در نظر می‌گیرد. برای حل مدل، از الگوریتم بهینه‌سازی ازدحام ذرات (PSO) استفاده شده است؛ زیرا این الگوریتم در برخورد با مسائل پیچیده و غیرخطی کارایی بالایی دارد. مدل پیشنهادی از طریق یک مطالعه موردی شامل سه پروژه واقعی با ۲۴ فعالیت و محدودیت‌های متنوع منابع اعتبارسنجی شده است. مقایسه با روش سنتی مسیر بحرانی (CPM) نشان داد که رویکرد مبتنی بر CCPM مدت زمان کل پروژه را از ۶۸ روز به ۴۷ روز، هزینه کل را از ۱۰۸۰۶۸۰ واحد به ۶۳۰۷۱۰ واحد کاهش داده و شاخص کیفیت را از ۴۱۲/۰ به ۵۷۶۴/۰ بهبود بخشیده است. به عبارت دیگر، این مدل باعث صرفه‌جویی ۲۱ روزه در زمان، کاهش ۹۷/۲۵ درصدی هزینه‌ها و بهبود ۴۰ درصدی کیفیت شده است. تحلیل حساسیت نشان می‌دهد که میزان دسترسی به منابع، به‌ویژه منابع K2 و K3، تأثیر قابل‌توجهی بر نتایج دارد؛ به طوری که محدودیت در K2 مدت زمان را به ۸۴ روز و هزینه را به ۸۸۰۱۱۰ واحد افزایش می‌دهد، و محدودیت در K3 مدت زمان را به ۸۱ روز و هزینه را به ۸۶۰۹۱۰ واحد می‌رساند. این یافته‌ها نشان می‌دهند که مدل پیشنهادی در بهبود تصمیم‌گیری و برنامه‌ریزی منابع در محیط‌های پیچیده ساخت‌وساز بسیار مؤثر است.