A Theoretical Framework for Case-Based Reasoning with State Transition Mechanisms: Application to Predictive Maintenance

Arijit Chatterjee

ORCID: 0009-0006-5658-4449

SEARS IT and Management Services Pvt Ltd, Pune, Maharashtra, India

E-mail: Arijit.chatterjee@transformco.com

Abstract

This paper develops a theoretical framework that integrates Case-Based Reasoning (CBR) with State Transition Mechanisms (STM) for predictive maintenance. Building on earlier work that introduced CBR+STM as a problem-solving paradigm, this study formalizes the approach for maintenance contexts by defining state spaces, transition functions, and a domain-informed heuristic. The framework prioritizes transparency and auditability—features critical in regulated industries—while remaining grounded in established search theory. The contribution lies in providing a structured and explainable alternative to black-box predictive models, highlighting complementarities with modern AI methods. As a theoretical contribution, empirical validation remains essential future work.

Keywords

Case-based reasoning; State transition mechanism; Theoretical framework; Predictive maintenance; Heuristic search; Explainable AI

Nomenclature

Symbols:

S = finite state space $\{s_0, s_1, ..., s_n\}$

A = action space $\{a_1, a_2, ..., a_m\}$

C = case base containing historical maintenance cases

h(s,a) = heuristic function for state-action pairs

f(s) = evaluation function f(s) = g(s) + h(s) where g(s) is cost to reach state s

δ(s,a) = state transition function

π = solution path or state sequence

w_1, w_2, w_3, w_4 = heuristic weight coefficients

C_maint(a) = direct maintenance cost for action a

C_downtime(s,a) = expected downtime cost

R_improvement(s,a) = reliability improvement measure

T_urgency(s) = time urgency factor for state s

≈ = similarity measure between states or cases

1. Introduction

Case-Based Reasoning (CBR) has proven effective for problem-solving across diverse domains by leveraging historical experience to address new problems [1]. However, traditional CBR approaches treat cases as monolithic problem-solution pairs, lacking systematic modeling of intermediate reasoning states and decision processes [2]. This limitation becomes particularly significant in complex domains such as predictive maintenance, where understanding the progression of equipment degradation and systematic planning of intervention strategies is crucial.

State Transition Mechanisms (STM) provide a complementary approach by modeling problem-solving as navigation through explicitly defined state spaces [3]. While STM has been successfully applied in planning and control domains, its integration with CBR for systematic case-based reasoning remains underexplored. This paper presents a theoretical framework that combines CBR with STM to address these limitations while maintaining the interpretability and adaptability advantages of case-based approaches.

1.2 Novelty beyond prior work

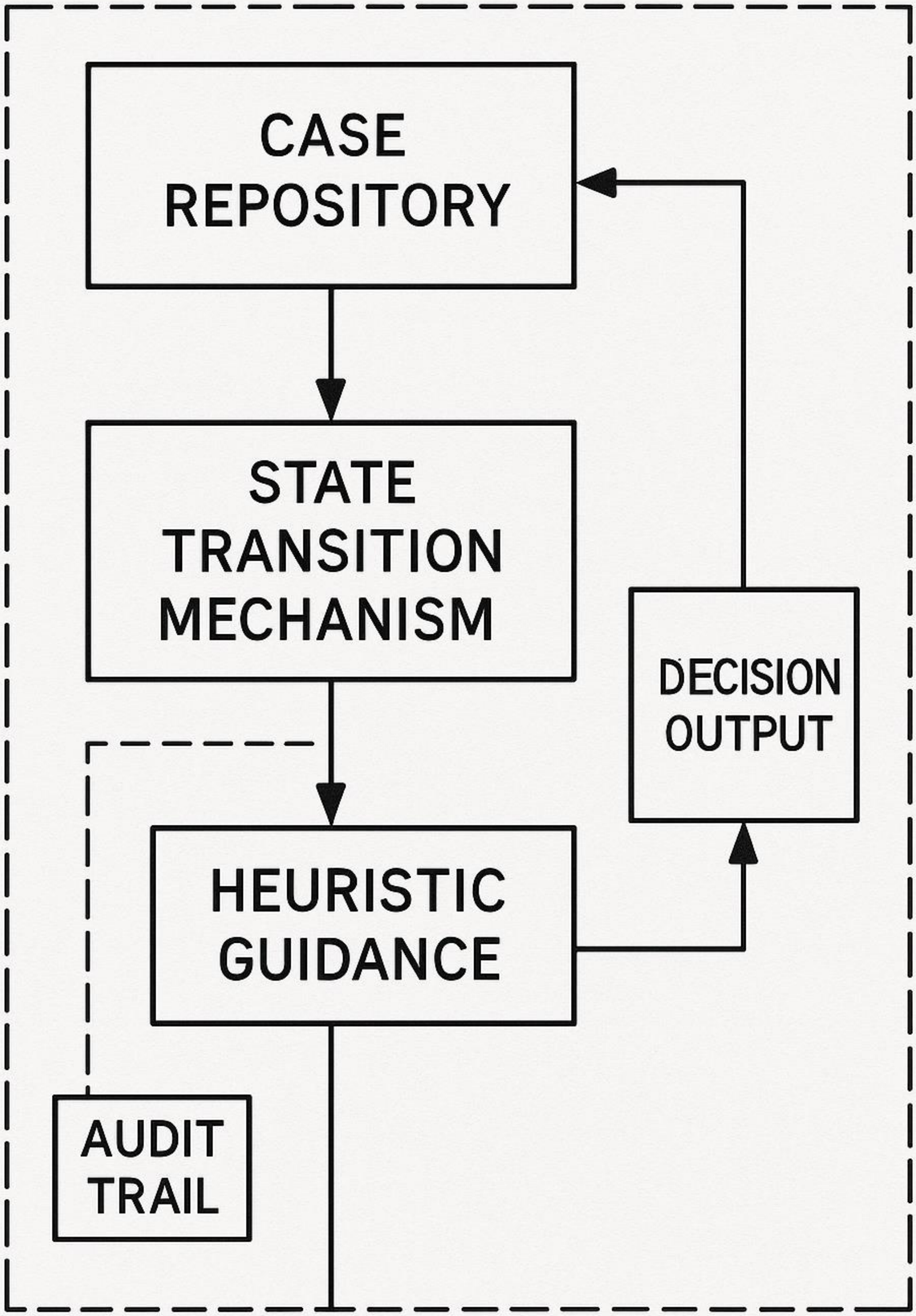
The proposed framework builds upon earlier work that introduced Case-Based Reasoning (CBR) with State Transition Mechanisms (STM) for general problem solving [4]. The present contribution extends that foundation in three important ways:

Domain-specific formalization: This study explicitly models degradation states and transitions relevant to predictive maintenance, providing a structured representation for reasoning about equipment health.

Heuristic definition: A multi-objective heuristic is introduced to capture practical trade-offs between cost, downtime, reliability improvement, and urgency, moving beyond the earlier generic state transition formulation.

Theoretical positioning: The paper situates the framework within classical search theory, clarifying admissibility conditions under bounded assumptions, inheriting complexity bounds from A* search, and contrasting the framework with both traditional CBR and contemporary AI methods.

Together, these contributions transform the earlier conceptual paradigm into a theoretical framework with explicit engineering relevance, particularly in contexts where explainability and auditability are essential.



CBR FRAMEWORK

Figure 1. CBR+STM theoretical framework architecture

2. Literature Review and Theoretical Context

2.1 Case-based reasoning foundations

Classical CBR operates through retrieve-reuse-revise-retain cycles, enabling adaptive problem-solving through historical case utilization [5]. Traditional implementations exhibit systematic limitations: monolithic case representation without intermediate state modeling; limited systematic handling of partial case matches; insufficient integration of multi-objective optimization within case retrieval and adaptation phases.

2.2 State transition mechanisms and search theory

State transition mechanisms model problem-solving as navigation through state spaces using well-defined operators and transition functions [6]. These approaches provide: explicit representation of intermediate reasoning states; systematic handling of operator preconditions and effects; formal analysis grounded in classical search theory [7]; structured approach to contingency planning and failure recovery.

2.3 Integration opportunities

While both CBR and STM demonstrate effectiveness in their respective domains, systematic integration remains limited. Existing approaches either treat state transitions as simple case features [8] or apply CBR for operator selection without explicit state space modeling [9]. This theoretical gap motivates integrated frameworks combining CBR's adaptability with STM's systematic reasoning capabilities.

3. Theoretical Framework Specification

3.1 Formal state space definition

Definition 1: Problem State Space

Let $S = \{s_0, s_1, ..., s_n\}$ represent a finite state space where each state $s_i \in S$ corresponds to a distinct problem configuration. For predictive maintenance applications, states represent equipment health conditions ranging from normal operation through various degradation levels.

Definition 2: State Transition Function

Let $\delta: S \times A \rightarrow S$ represent a state transition function where $A = \{a_1, a_2, ..., a_m\}$ represents the action space. For each state $s \in S$ and action $a \in A$, $\delta(s,a)$ yields the resulting state following action application.

Definition 3: Case-State Correspondence

Each CBR case $c \in C$ corresponds to a state transition sequence $\pi = \langle s_0, a_0, s_1, a_1, ..., s_k \rangle$ where s_0 represents initial problem state, s_k represents goal state, and a_i represents actions applied during case resolution.

Similarity Measurement: The similarity relation $\approx$ in case retrieval is measured using a weighted feature vector comparison considering both initial state characteristics and anticipated trajectory patterns. Specifically, similarity(c, s_0) combines static state features (equipment type, age, operating conditions) with dynamic trajectory indicators (degradation rate, failure history patterns) using domain-specific weight vectors calibrated to maintenance context.

State Transition Pattern Extraction: The system extracts patterns by analyzing successful case sequences to identify: (i) common state transition pathways for similar equipment types; (ii) action effectiveness patterns showing which maintenance interventions succeed under specific degradation conditions; (iii) temporal patterns indicating optimal intervention timing. These patterns guide both case adaptation and new path generation.

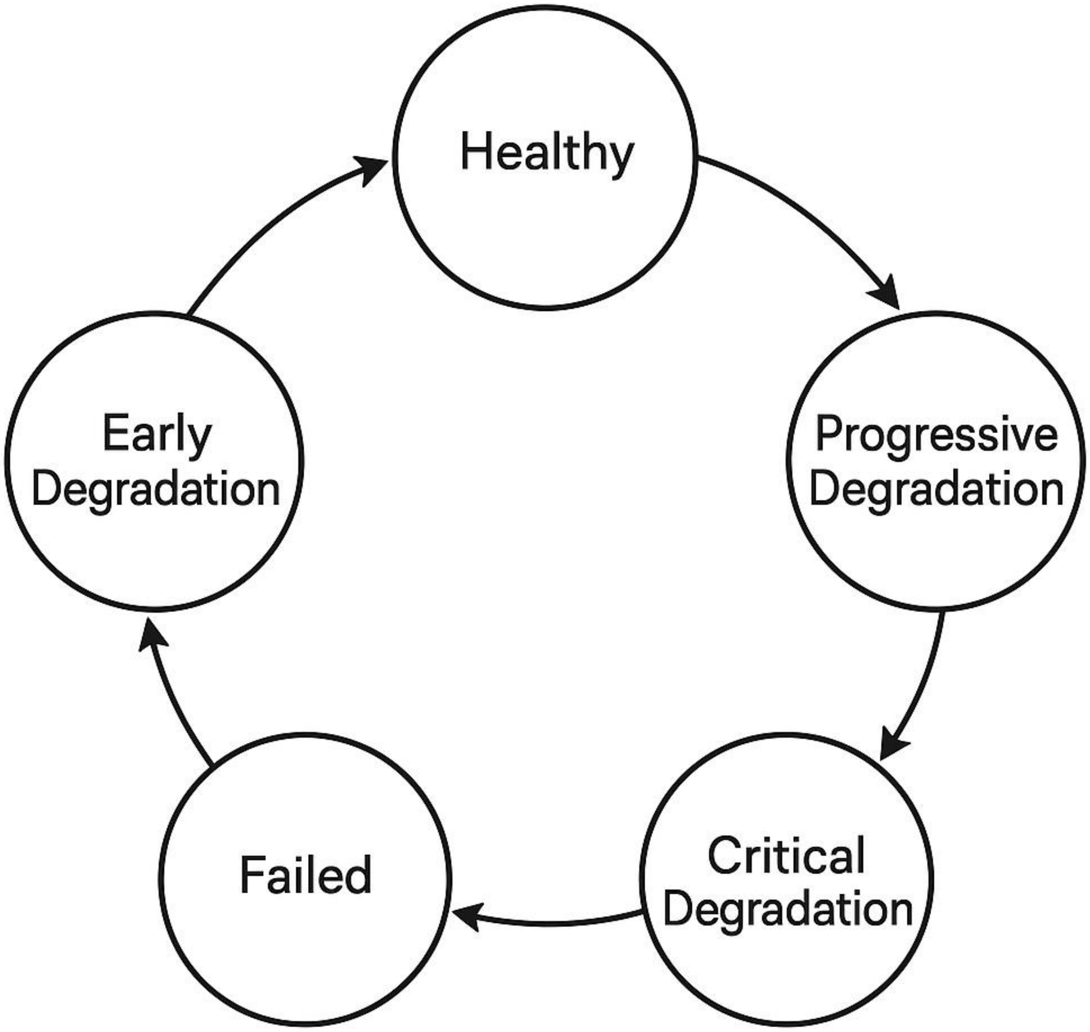


Figure 2. State transition model for equipment degradation

3.2 Multi-objective heuristic function

Definition 4: Multi-Objective Heuristic Function

The heuristic function $h: S \times A \rightarrow \mathbb{R}^+$ provides cost estimates for state-action pairs:

$$h(s,a) = w_1 \cdot C\_maint(a) + w_2 \cdot C\_downtime(s,a) - w_3 \cdot R\_improvement(s,a) + w_4 \cdot T\_urgency(s), \quad (1)$$

where C_maint(a) represents direct maintenance cost, C_downtime(s,a) represents expected downtime cost, R_improvement(s,a) represents projected reliability gain, T_urgency(s) represents criticality factors, and $w_1, w_2, w_3, w_4 \geq 0$ represent weight coefficients calibrated to the application domain.

3.3 Heuristic function properties

In predictive maintenance, selecting an action requires balancing multiple factors, including intervention cost, downtime, reliability gain, and urgency. To represent these trade-offs, we define a multi-objective heuristic as shown in equation (1), where each term encodes domain-relevant costs or benefits, and $w_i \geq 0$ are weights reflecting operator priorities.

This formulation is not claimed to be universally admissible. Instead, under the assumption that costs are non-negative, reliability improvements are bounded, and urgency scores are consistently scaled, the heuristic functions as a domain-informed guide rather than a guarantee of optimality. Its primary value lies in providing a structured way to incorporate engineering trade-offs into the reasoning process.

The practical significance of this approach is that it enables systematic incorporation of maintenance domain knowledge while maintaining transparency in decision-making, supporting regulatory compliance and audit trail generation in safety-critical applications.

3.4 Practical heuristic calibration methods

For engineering deployment, the heuristic weight coefficients (w_1, w_2, w_3, w_4) require systematic calibration using established methodologies [12]:

Expert Elicitation using Analytic Hierarchy Process (AHP): Maintenance engineers and domain experts can be systematically queried using pairwise comparisons to derive relative importance weights [12]. For example, comparing "direct maintenance cost versus downtime cost" across multiple scenarios to establish consistent weight ratios. AHP provides mathematical consistency checking and enables group decision-making among multiple experts.

Sensitivity Analysis for Deployment Robustness: Critical for real-world deployment, sensitivity analysis examines how weight variations affect solution paths π [13]. This involves: (i) parametric studies varying each w_i within realistic bounds (±20-50%); (ii) Monte Carlo analysis with weight uncertainty distributions; (iii) identification of weight threshold values where optimal actions change. Such analysis ensures robust performance under parameter uncertainty.

Inverse Reinforcement Learning from Historical Data: Advanced implementations could employ inverse reinforcement learning to learn optimal weights from historical maintenance logs [14]. By analyzing sequences of maintenance decisions made by expert technicians, the system can infer implicit weight preferences through maximum likelihood estimation or maximum entropy methods. This approach requires substantial historical data but can capture nuanced domain-specific preferences not easily articulated by experts.

4. Theoretical Algorithm Specification

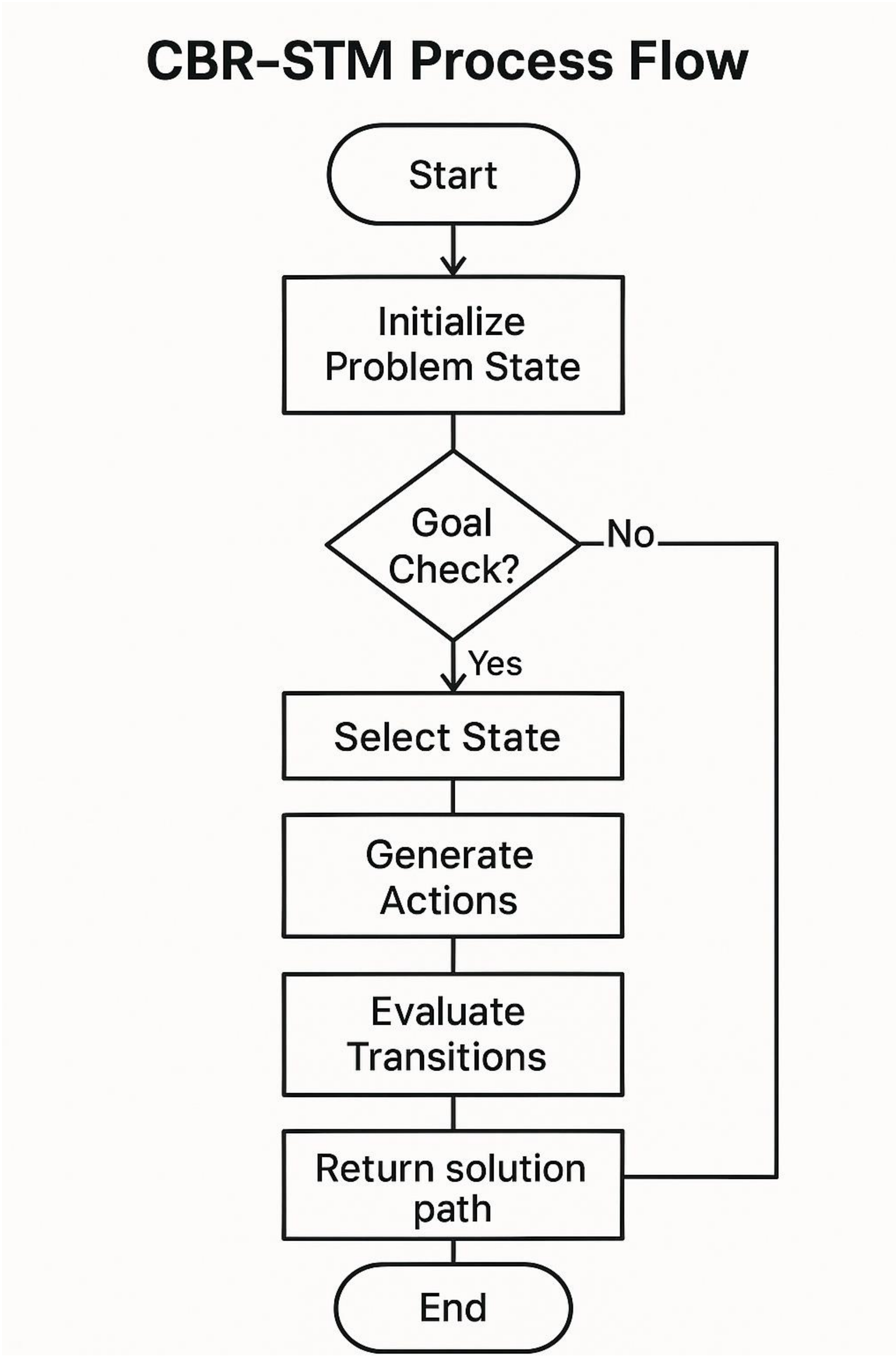

CBR–STM Process Flow
Start
Initialize Problem State
Goal Check?
No
Yes
Select State
Generate Actions
Evaluate Transitions
Return solution path
End

Figure 3. CBR+STM algorithm structure

Algorithm 1: CBR+STM Theoretical Framework

Input: Problem state s_0, case base C, heuristic parameters W

Output: Solution path π with justification trace

Phase 1: State-Based Case Retrieval

1. Map current problem to state space: $s_0 \leftarrow$ state_mapping(problem)

2. Identify relevant historical cases: C' ← {c ∈ C | similarity(initial_state(c), s_0) > threshold}

3. Extract state transition patterns from retrieved cases using pattern analysis

Phase 2: Heuristic-Guided State Navigation

1. Initialize search frontier: OPEN ← {s_0}, g(s_0) ← 0

2. While OPEN ≠ ∅ and goal not reached:

   a. Select state s ← argmin{f(s')} from OPEN where f(s') = g(s') + min_a h(s',a)

   b. Generate applicable actions A(s) based on extracted case patterns

   c. For each action a ∈ A(s):

      - Evaluate transition s_new ← δ(s,a)

      - Update costs g(s_new) ← g(s) + cost(s,a)

      - Add s_new to OPEN if improved solution

3. Return complete state sequence π with decision justifications and cost breakdown

Note: The evaluation function f(s) follows standard A* formulation where g(s) represents accumulated cost to reach state s, and h(s',a) represents heuristic estimate from equation (1).

5. Theoretical Analysis and Properties

5.1 Framework properties

Completeness: Under finite state spaces with appropriate heuristic guidance, the framework can discover solutions when they exist within the defined problem domain, inheriting completeness properties from classical search algorithms.

Traceability: Complete audit trails are generated by construction through explicit state sequence recording and decision justification documentation, supporting regulatory compliance requirements [16].

Adaptability: Incremental case addition enables framework evolution without requiring complete reconfiguration, maintaining the adaptive advantages of CBR approaches.

6. Analysis and Comparative Positioning

6.1 Application framework

Predictive maintenance applications require specific framework instantiation: equipment health states ranging from normal operation through progressive degradation levels; maintenance interventions with associated costs and effectiveness measures; regulatory compliance integration through systematic audit trail generation meeting industrial safety standards [16][17].

6.2 Complexity analysis

The computational complexity of the framework arises from two main components:

Case retrieval: With appropriate indexing (e.g., kd-trees, clustering), retrieval can achieve expected O(log n) time, though worst-case remains O(n) for n stored cases.

State-space search: The search inherits complexity properties from classical A* [7], with worst-case complexity exponential in depth d and branching factor b, i.e., O(b^d). In predictive maintenance, practical branching factors are limited by finite maintenance actions and failure modes, making the problem tractable in many real-world settings.

Thus, the framework does not introduce new complexity bounds but rather adapts established search-theoretic properties to a predictive maintenance context. The contribution lies in demonstrating how these bounds can be meaningfully applied in an engineering domain where transparency and auditability are critical.

6.3 Comparative analysis and positioning

Contemporary predictive maintenance leverages statistical models, deep learning architectures, graph neural networks, and increasingly transformer-based large language models. These methods achieve strong predictive accuracy when large datasets are available but often struggle to provide transparent reasoning or regulatory audit trails.

By contrast, CBR+STM emphasizes explainability and incremental adaptability. Each diagnostic or maintenance decision is grounded in explicit transitions between health states, allowing for traceable justifications. Neural approaches can process sensor data at scale and discover latent features, while CBR+STM contributes structured reasoning for action selection.

Concrete Hybrid Architecture Example: A practical implementation could leverage the Informer transformer model [10] for the "perception" task of processing multi-variate sensor streams (temperature, vibration, pressure, electrical signatures) to forecast the next likely degradation state s_i with confidence bounds. This predicted state information then feeds into the CBR+STM framework to perform "interpretable decision-making" about the optimal maintenance action a_i using equation (1). The transformer handles complex pattern recognition in high-dimensional sensor data, while CBR+STM ensures that the resulting maintenance decisions are explainable, auditable, and grounded in historical case precedents.

The two paradigms are therefore complementary: neural models excel at perception and prediction, while CBR+STM supports interpretable decision-making and compliance reporting [18]. This complementarity suggests hybrid architectures where data-driven AI provides input signals and symbolic reasoning ensures explainable outcomes, a combination especially relevant in safety-critical and regulated industries.

7. Limitations and Future Research

Current limitations: As a theoretical contribution, the framework lacks empirical validation on real industrial datasets; domain-specific state space definition requires substantial expert knowledge; heuristic parameter tuning requires extensive domain analysis and may not generalize across equipment types; the framework requires integration with existing maintenance management systems for practical deployment.

Future research directions: Prototype implementation on established maintenance datasets; comparative evaluation against commercial predictive maintenance platforms; field testing in regulated industrial environments; integration opportunities with neural approaches for hybrid architectures combining pattern recognition with systematic reasoning; automated ontology development for reduced domain modeling overhead; empirical validation of the theoretical framework through controlled studies.

8. Conclusion

This paper presented a theoretical framework for integrating CBR with STM in predictive maintenance. Its primary contribution is conceptual: defining state transitions, introducing a multi-objective heuristic under domain assumptions, and framing complexity within classical bounds. Empirical validation with industrial datasets remains an essential next step for demonstrating practical effectiveness.

The framework's strength lies in its explainability and auditability, features that complement but do not replace modern data-driven AI methods. This positions CBR+STM as a theoretical foundation for future hybrid approaches, bridging symbolic reasoning and statistical learning in safety-critical maintenance applications. The work contributes to explainable AI research by demonstrating how systematic reasoning and complete traceability can be integrated within case-based problem-solving approaches.

Acknowledgments

The author declares no specific funding was received for this research work.

Conflict of Interest

The author declares no conflict of interest.

References

[1] Aamodt A, Plaza E 1994 Case-based reasoning: Foundational issues, methodological variations, and system approaches. AI Commun. 7(1): 39–59

[2] Kolodner J 1993 Case-Based Reasoning. Morgan Kaufmann Publishers, San Francisco

[3] Russell S, Norvig P 2020 Artificial Intelligence: A Modern Approach, 4th edn. Pearson, Boston

[4] Chatterjee A 2006 Case based reasoning with state transition mechanism for problem-solving in AI. IAENG Int. J. Comput. Sci. 32(4): 1–8

[5] Leake D B 1996 CBR in context: The present and future. In: Leake D B (ed) Case-

Based Reasoning: Experiences, Lessons and Future Directions. MIT Press, Cambridge, pp 3–30

[6] Korf R E 1990 Real-time heuristic search. Artif. Intell. 42(2–3): 189–211

[7] Hart P E, Nilsson N J, Raphael B 1968 A formal basis for the heuristic determination of minimum cost paths. IEEE Trans. Syst. Sci. Cybern. 4(2): 100–107

[8] Cunningham P 2009 A taxonomy of similarity mechanisms for case-based reasoning. IEEE Trans. Knowl. Data Eng. 21(11): 1532–1543

[9] Muñoz-Avila H, Cox M T 2008 Case-based plan adaptation: An analysis and review. IEEE Intell. Syst. 23(4): 75–81

[10] Zhou H, Zhang S, Peng J, Zhang J, Li J, Xiong H, Zhang W 2021 Informer: Beyond efficient Transformer for long sequence time-series forecasting. In: Proc. AAAI Conf. Artif. Intell. 35(12): 11106–11115

[11] Raissi M, Perdikaris P, Karniadakis G E 2019 Physics-informed neural networks: A deep learning framework for solving forward and inverse problems. J. Comput. Phys. 378: 686–707

[12] Saaty T L 1980 The Analytic Hierarchy Process: Planning, Priority Setting,

Resource Allocation. McGraw-Hill, New York

[13] Saltelli A, Ratto M, Andres T, Campolongo F, Cariboni J, Gatelli D, Saisana M, Tarantola S 2008 Global Sensitivity Analysis: The Primer. John Wiley & Sons, Chichester

[14] Abbeel P, Ng A Y 2004 Apprenticeship learning via inverse reinforcement learning. In: Proc. 21st Int. Conf. Mach. Learn., pp 1–8

[15] O'Connor A N, Modarres M 2006 Probability distributions used in reliability engineering. Center for Risk and Reliability, University of Maryland, College Park

[16] ISO 13374-1 2003 Condition monitoring and diagnostics of machines - Data processing, communication and presentation - Part 1: General guidelines. International Organization for Standardization, Geneva

[17] Lebold M, Thurston M 2001 Open standards for condition-based maintenance and prognostic systems. In: Proc. 5th Annu. Maint. Reliab. Conf. (MARCON), pp 1–10

[18] Garcez A S d'Avila, Lamb L C 2023 Neurosymbolic AI: The 3rd Wave. Artif. Intell. Rev. 56(11): 12387–12406

Figure Captions

Figure 1. Theoretical architecture of the CBR+STM framework showing integration of case repository, state transition mechanism, heuristic guidance, and decision output components for systematic problem-solving with complete audit trail generation.

Figure 2. State transition model illustrating equipment degradation progression through defined health states with explicit transition pathways and decision points for maintenance intervention. Arrows should be labeled with specific maintenance actions (e.g., "Corrective Maintenance," "Preventive Maintenance," "Emergency Repair").

Figure 3. Algorithmic structure of the CBR+STM framework showing the systematic process from problem state identification through heuristic-guided navigation to solution generation with complete decision justification and audit trail documentation for regulatory compliance.

Note: All figures must be inserted with axes 0.3 mm thickness, curves 0.6 mm thickness, and lower-case lettering as per Sadhana guidelines.