

A Modular Decision-Support Framework for Maintenance Scheduling Integrating NSGA-II, TOPSIS, and Generative AI

Rafael Calixto Maluf¹[0009-0000-5287-332X], Rafael Araujo Kluska¹[0000-0002-5296-910X],
Gustavo Herold Bender¹[0009-0001-4033-3462], Lucas Camilotti¹[0000-0002-2253-587X] and
Eduardo de Freitas Rocha Loures^{1,2}[0000-0002-1963-6186]

¹ Pontifical Catholic University of Paraná (PUCPR), Curitiba, Brazil

² Federal University of Technology – Paraná (UTFPR), Curitiba, Brazil
rafael.maluf@pucpr.edu.br

Abstract. Efficient asset maintenance is essential for ensuring operational availability and reducing costs in industrial systems. However, traditional Computerized Maintenance Management Systems (CMMS) often function as data repositories rather than decision-support tools. This paper presents a modular computational framework for CMMS that integrates multi-objective optimization, multicriteria decision-making (MCDM), and Generative Artificial Intelligence (GenAI) to automate maintenance scheduling. The approach was tested in a modular environment using a database schema inspired by an industrial CMMS, implemented with PostgreSQL and Docker containers. First, the NSGA-II algorithm is applied to balance conflicting objectives, specifically minimizing asset downtime and maintenance costs. Subsequently, the TOPSIS method ranks the optimized solutions based on strategic criteria, such as asset criticality and operational impact. Finally, the prioritized tasks are processed by an external API integrated with a GenAI model, which automatically generates a weekly schedule assigned to available technicians. The main contribution of this paper is an end-to-end, modular decision-support pipeline that integrates multi-objective optimization (NSGA-II), MCDM ranking (TOPSIS), and a GenAI-based scheduling layer while preserving decision authority in the optimization and ranking stages. Results obtained from a CMMS-inspired dataset demonstrate the framework's viability in prioritizing maintenance tasks and producing balanced, execution-ready agendas.

Keywords: Maintenance Management; CMMS; Multi-objective Optimization; NSGA-II; TOPSIS; Generative AI; Decision-Support.

1 Introduction

Industrial maintenance has evolved from a reactive approach to a strategic function essential for operational efficiency and cost control [1,2]. The transition to Industry 4.0 has accelerated the adoption of digital technologies, including Artificial Intelligence (AI), data-driven analytics, and automated decision pipelines [3,4,5,6].

However, many Computerized Maintenance Management Systems (CMMS) still operate primarily as data repositories, providing limited decision-support capabilities and weak integration with advanced analytics in practice [2,6,7].

A major challenge in maintenance management is balancing conflicting objectives, such as minimizing asset downtime while controlling operational costs [1,8,9]. Multi-objective optimization algorithms, particularly NSGA-II, are widely adopted for generating trade-off solutions in such scenarios because they can explore a diverse Pareto set without collapsing objectives into a single weighted score [10,11]. Since multi-objective methods typically return a set of non-dominated alternatives rather than a single answer, multicriteria decision-making (MCDM) techniques such as TOPSIS are often applied to rank the optimized solutions using strategic criteria (e.g., criticality, cost, and operational impact) and produce an actionable priority order [12,13,14].

Despite these advances, there is still a practical gap between quantitative prioritization and real-world execution. In many CMMS-centered workflows, optimization and ranking outputs are not directly translated into feasible task assignments, which keeps scheduling dependent on manual effort or rigid rules and limits scalability [2,6,7]. Recent progress in AI-based scheduling and decision-support systems has shown that intelligent modules can bridge this “last mile” by mapping prioritized tasks into human-executable plans under operational constraints [5]. In this context, a Generative AI component can act as a controlled interface layer that transforms structured, pre-ranked maintenance outputs into a readable weekly agenda, while preserving the priorities defined by the optimization and MCDM stages and reducing subjectivity in allocation [4,15].

This paper proposes a modular computational framework that integrates multi-objective optimization (NSGA-II), multicriteria decision-making (TOPSIS), and an AI-driven scheduling layer to support decision-making in asset maintenance. The system was validated using a simulated environment based on the structure of a CMMS-like data model, demonstrating the feasibility of automating the flow from task prioritization to weekly staff scheduling. The remainder of this paper is organized as follows: Section 2 describes the research methodology adopted to design and validate the proposed approach. Section 3 presents the resulting modular computational framework and its main components. Section 4 reports a case application using a synthetic dataset and discusses the observed outcomes. Section 5 concludes the paper and outlines directions for future work.

2 Methodology

This work followed an applied, artifact-oriented research workflow to design and validate a modular computational framework for maintenance planning. The study began with a targeted theoretical and technical review covering maintenance

engineering and CMMS workflows, Industry 4.0-oriented maintenance systems, multi-objective optimization, multicriteria decision making, and AI-assisted scheduling concepts, which informed the design requirements and the modular decomposition adopted in the proposed solution [2,4,6,7]. Based on this grounding, the research procedures were organized into four main steps: (i) definition of the maintenance decision context and minimum CMMS-like data requirements (work orders, costs, estimated durations, criticality, and technician availability) [2,6,7]; (ii) implementation of a multi-objective optimization stage to generate trade-off alternatives between cost control and downtime reduction [1,9,10,11]; (iii) implementation of an MCDM ranking stage to convert the Pareto set into a single actionable priority order [12,13,14]; and (iv) integration of the ranked outputs with an AI-based scheduling layer to generate an execution-ready weekly agenda under operational constraints, while preserving the priorities defined by the previous stages [4,15].

To ensure replicability, the environment was containerized using Docker and backed by a PostgreSQL database structured to emulate typical CMMS records, including assets, work orders, and technician availability [2,6]. The optimization module was implemented in Python using NSGA-II due to its widespread adoption and robustness in multi-objective scheduling contexts [10,11], with binary decision variables and objective functions defined as the minimization of maintenance cost and downtime [1,9]. In the experimental setup, NSGA-II was configured with a population size of 150 individuals and executed for 150 generations, with a fixed random seed to ensure reproducibility [11]. The optimization stage produces a Pareto set of non-dominated solutions, meaning that no solution can improve one objective without worsening at least one other. Since the optimization stage returns multiple efficient but not naturally ordered alternatives, TOPSIS was implemented as a transparent ranking mechanism to convert the Pareto set into a single operational priority order suitable for automated decision pipelines [13]. Solutions were evaluated using three criteria, namely asset criticality (benefit, since higher criticality increases intervention urgency), cost (cost), and downtime (cost), with Min–Max normalization and weights set to reflect typical industrial prioritization (criticality 0.5, cost 0.3, downtime 0.2) [14,16].

Validation was performed through a simulated weekly maintenance scenario designed to reflect realistic work-order heterogeneity and scheduling constraints. The framework was assessed by verifying the feasibility of the generated agenda (e.g., no technician exceeding 8 hours/day), the preservation of the TOPSIS-derived priority order during allocation, and the consistency of the resulting schedule with criticality and operational impact considerations [8]. This validation strategy demonstrates the end-to-end viability of converting CMMS-like inputs into ranked decisions and a conflict-free weekly schedule within a modular architecture.

3 Results

The main result of this research is a modular computational framework that converts CMMS-like maintenance data into an execution-ready weekly schedule through a structured pipeline. The framework comprises four independent modules: (i) data environment, (ii) multi-objective optimization, (iii) multicriteria ranking, and (iv) AI-based scheduling, which are connected through well-defined interfaces. This modular architecture supports replication across different maintenance contexts and enables each module to be improved, replaced, or extended independently. For example, alternative optimization or ranking techniques can be adopted, or the scheduling layer can be enhanced, without disrupting the remaining workflow, as summarized in Fig. 1. As a result, the pipeline separates decision generation (optimization and ranking) from execution (scheduling), improving transparency and maintainability.

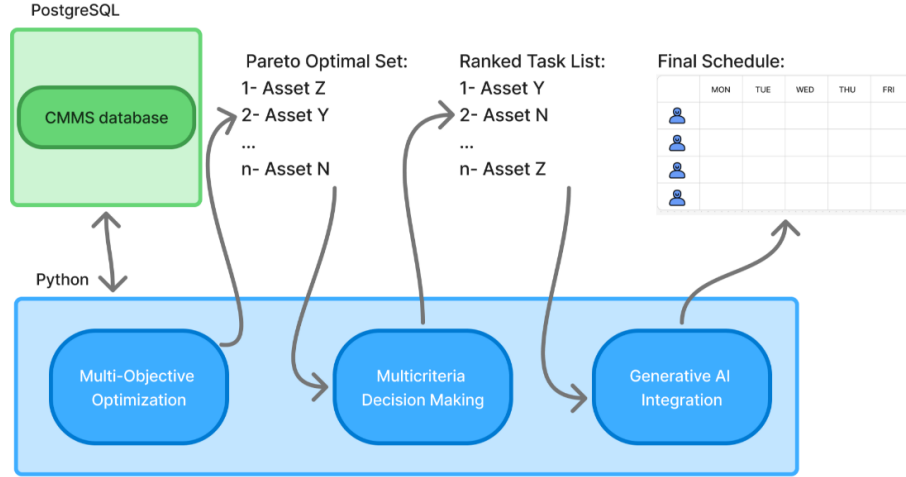


Fig. 1. Workflow of the proposed framework integrating Multi-Objective Optimization (NSGA-II), Multicriteria Decision Making (TOPSIS), and Generative AI for maintenance scheduling.

3.1 Data Environment and CMMS-like Simulation

The first module establishes a CMMS-like data layer that stores and provides structured maintenance records required by the downstream decision stages. The database schema captures core entities such as assets, work orders, and technician availability, including attributes used in planning and prioritization (e.g., estimated duration, material/labor cost, and criticality level) [2,7]. The module outputs a standardized dataset representing the current maintenance backlog and available workforce capacity, ensuring that subsequent modules operate on consistent and reproducible inputs. By relying on containerized services and a relational database,

the framework supports repeatable experimentation and deployment-oriented execution [6].

3.2 Multi-Objective Optimization Module (NSGA-II)

The second module generates a set of candidate maintenance selections by explicitly modeling the trade-off between operational cost control and downtime minimization [1,9]. Using NSGA-II, the module produces a Pareto-optimal set of non-dominated alternatives rather than a single solution, which is appropriate for maintenance planning problems where objectives are inherently conflicting [10,11]. The output of this module is a reduced set of high-quality alternatives that excludes dominated choices and narrows the decision space for the ranking stage. This design ensures that the framework preserves diverse solutions ranging from low-cost/higher-downtime scenarios to higher-cost/lower-downtime scenarios without collapsing the problem into a single weighted objective [11].

3.3 Multicriteria Ranking Module (TOPSIS)

The third module converts the Pareto set into an actionable priority order suitable for operational use. Since Pareto-optimal alternatives are not naturally ordered, TOPSIS is applied as a transparent ranking mechanism that evaluates each alternative against decision criteria and selects those closest to the positive ideal solution [13]. TOPSIS was selected for its interpretability and auditability, producing a single deterministic ranking that can be directly operationalized by the downstream scheduling layer. In the proposed framework, asset criticality is treated as a benefit criterion, while cost and downtime are treated as cost criteria, enabling the ranking process to reflect strategic maintenance priorities [12,14]. The output is a single ranked list of tasks (or solution choices) that can be consumed directly by the scheduling module, providing a deterministic and auditable decision layer between optimization and execution [13,16].

3.4 AI-based Scheduling Module and Interface Layer

The final module translates the ranked outputs into an execution-ready weekly agenda. The module receives a structured, pre-ranked list of maintenance tasks along with operational constraints (e.g., technician availability and a maximum daily workload of 8 hours) and produces a structured schedule representation (JSON) mapping tasks to technicians and time slots [5]. Importantly, the scheduling layer does not modify the priorities produced by NSGA-II and TOPSIS; instead, it acts as an allocation interface that operationalizes the ranked plan under constraints, supporting consistency and reducing subjectivity in assignment decisions [4,15]. This constrained role is aligned with trustworthiness concerns in AI-assisted maintenance settings, as the AI component is used to operationalize decisions rather than to redefine them [4].

Overall, the resulting framework provides an end-to-end, modular path from CMMS-like data to a conflict-free weekly plan. Its architecture supports future extensions such as integration with real CMMS deployments, IoT-driven updates for dynamic rescheduling, and alternative optimization or ranking methods depending on domain constraints.

4 Case Application and Discussion

To assess the proposed framework, the system was executed on a synthetic, CMMS-inspired dataset designed to represent a typical week of industrial maintenance demand. The evaluation focused on architectural and functional feasibility rather than on quantitative performance gains, assessing whether the pipeline (i) preserves end-to-end consistency from CMMS-like inputs to an execution-ready weekly schedule, (ii) respects operational constraints (e.g., technician availability and workload limits), and (iii) produces a deterministic, auditable prioritization-to-scheduling flow. The dataset included heterogeneous work orders with attributes such as estimated repair time, material and labor costs, and criticality levels, reflecting the operational information commonly handled in CMMS-like environments [2,7].

The case workflow started with the multi-objective optimization stage which aims to retain non-dominated alternatives under a cost–downtime trade-off formulation [9]. In practical terms, the optimization filters out options that are inefficient because they present higher cost without offering proportional benefits in downtime reduction, thereby narrowing the decision space for the subsequent ranking stage. Table 1 illustrates this filtering effect and shows examples of work orders that were retained in the Pareto set versus work orders that were discarded as dominated, consistent with NSGA-II-based multi-objective baselines reported in the literature [10,11].

Table 1. Optimization results show the filtering of dominated solutions by NSGA-II.

Work Order	Specialty	Cost	Duration (h)	Optimization Status
WO005	Industrial	\$ 5,200.00	15.0	Selected (Pareto Front)
WO006	Boilermaker	\$ 850.00	3.0	Selected (Pareto Front)
WO099	General	\$ 4,800.00	14.0	Discarded (Dominated)
WO003	Hydraulics	\$ 2,100.00	8.0	Selected (Pareto Front)
WO088	Painting	\$ 1,500.00	10.0	Discarded (Dominated)

As observed, tasks such as WO099 were discarded because they were dominated by better alternatives, which reduced the search space for the decision-making module and improved the tractability of the downstream prioritization step [11].

After optimization, TOPSIS was applied to transform the remaining alternatives into a single actionable priority order. Since Pareto-optimal outputs are not inherently ranked, TOPSIS provides a transparent mechanism to order candidates according to weighted business criteria (e.g., asset criticality and operational impact), which is consistent with decision-support approaches that combine multi-objective optimization with MCDM ranking in maintenance contexts [12,14]. Table 2 presents the resulting ranking and defines the order in which tasks are inserted into the scheduling stage.

Table 2. Final ranking of maintenance tasks after TOPSIS processing.

Rank	Work Order	Specialty	Priority Logic	Impact Factor
1	WO007	Welding	High Criticality	Structural Safety
2	WO005	Industrial	High Impact	Major Production Line
3	WO003	Hydraulics	Efficiency	Fluid Leak Risk
4	WO006	Boilermaker	Quick Win	Fast Repair

Table 2 also illustrates the practical logic of the framework: even long tasks may be ranked highly when their operational impact is significant, while safety-related interventions are prioritized early to reduce risk exposure, aligning the ranking behavior with structured TOPSIS decision schemes used in group and engineering contexts [13,16]. It is worth noting that this table displays the top-ranked results from the complete dataset, which includes high-priority tasks (e.g., WO007) that were not present in the random sample shown in Table 1.

The final step integrated the ranked outputs with the AI-based scheduling module. The API received the structured, pre-ranked list and generated a weekly agenda by allocating tasks to technicians while enforcing feasibility constraints, such as not exceeding a daily 8-hour workload per technician, which reflects common operational scheduling limits in maintenance environments [8,11]. Fig. 2 shows the resulting schedule and indicates that the allocation adhered to workload constraints, producing a conflict-free plan ready for deployment [5].



Fig. 2. Final weekly maintenance schedule generated by the AI module.

The distribution pattern—higher concentration of tasks at the beginning of the week and gaps in subsequent days—reflects the prioritization effect of the ranking stage, which pushes high-criticality work orders earlier in the plan. This behavior is aligned with integrated production–maintenance scheduling perspectives found in intelligent industrial systems, where early mitigation of high-impact issues is often desirable [5].

Beyond feasibility, this case highlights the practical contribution of the proposed pipeline when contrasted with traditional CMMS-based planning. In conventional settings, maintenance decisions are frequently performed manually or based on simple priority rules, which can lead to fragmented analysis and limited consideration of conflicting objectives [2,7].

From a managerial perspective, the framework can support decisions such as which work orders should be prioritized first, how to balance asset criticality against cost and downtime, and how to distribute the prioritized workload across available technicians while respecting operational constraints. In addition, by making trade-offs explicit and converting them into a deterministic ranking and schedule, the pipeline can help managers identify bottlenecks, the concentration of high-impact interventions, and specialty-specific workload imbalances that might otherwise remain obscured in conventional CMMS-based planning.

These case observations are consistent with recent literature indicating that multi-objective optimization is suitable for handling conflicting criteria in maintenance decision problems, and that combining optimization with structured ranking can produce actionable plans [9,10,11]. In addition, this work advances beyond theoretical formulations by implementing the end-to-end workflow within a containerized environment backed by a relational database, strengthening practical reproducibility and enabling deployment-oriented experimentation [6,5].

Furthermore, the case supports the flexibility of the modular architecture. Because each component operates as an independent service, isolated customization becomes feasible; for instance, the scheduling layer can be replaced or enhanced with additional domain constraints without disrupting the upstream optimization and ranking logic, consistent with Industry 4.0-oriented maintenance system evolution and integrated intelligent manufacturing architectures [6,5].

Finally, in this case, feeding the AI module with a pre-ranked structured dataset produced by TOPSIS reduced the risk of unreliable or ungrounded outputs. Under this design, the AI acts strictly as an allocation interface rather than a decision-maker, preserving decision authority within the mathematical optimization and ranking stages and aligning with trustworthiness and reliability concerns in AI-assisted maintenance systems [4,15].

5 Conclusion

This paper presented a framework for Asset Maintenance Management that integrates Multi-Objective Optimization, Multicriteria Decision Making, and an AI-based scheduling layer. The modular architecture, built on Docker, Python, and PostgreSQL, allowed for the simulation of a real-world CMMS environment, validating the technical feasibility of the proposal.

The main contribution of this work is the establishment of a seamless workflow from data extraction to schedule generation. By using NSGA-II, the system ensures that trade-offs between cost and downtime are mathematically optimized. The subsequent application of TOPSIS aligns these technically optimal solutions with strategic business goals. Finally, the GenAI integration addresses the "last mile" challenge, automating the translation of complex prioritized lists into human-readable assignments for technicians. Due to its modular and system-agnostic design, the proposed framework can be adapted to different CMMS platforms and to a wide range of asset-intensive industrial contexts with limited architectural changes. In practical applications, this adaptation mainly depends on mapping the company's maintenance database structure and configuring decision criteria, weights, and operational constraints according to the organization's maintenance priorities and workforce characteristics.

At the current stage, this study focuses on architectural and functional feasibility of the proposed pipeline in a controlled setting using a CMMS-inspired dataset, without pursuing organizational validation in real operational environments. Future work will therefore focus on real-world deployment using industrial CMMS logs, including the mapping of company-specific data structures and the configuration of decision criteria, ranking weights, and operational constraints according to each maintenance context. This next stage will also include user-oriented evaluation with maintenance planners and supervisors, as well as comparative assessment against current manual scheduling practices. In addition, an economic feasibility analysis will be conducted to quantify potential cost reductions and efficiency gains compared to manual scheduling. Finally, predictive capabilities will be explored by extending the AI layer with maintenance analytics to support early failure anticipation and proactive work-order generation.

References

1. Peres, R.C., Lima, G.B.A.: Proposal of a model for maintenance cost control focusing on the application of balanced indicators. In: Proc. of Industrial Engineering Conf., pp. 1–10 (2008).
2. Shankar, L., Singh, C.D., Singh, R.: Bridging the gap: Challenges and solutions for implementing CMMS in manufacturing industries. *Int. J. Ind. Eng. Manag.* 8(3), 1–12 (2019).

3. Oliveira, P.V.S., Santos, L.F., Ferreira, M.P.: Artificial intelligence in industrial process automation and its impacts. *Revista de Economia Mackenzie* 21(1), 162–182 (2024).
4. Ucar, A., Karakose, M., Kırımça, N.: Artificial intelligence for predictive maintenance applications: Key components, trustworthiness, and future trends. *Applied Sciences* 14, 1–25 (2024).
5. Mota, B., Silva, A., Lima, R., Costa, A.P.C.S.: Review of manufacturing integration between production, maintenance and quality artificial intelligence systems. *Journal of Industrial Information Integration* 40, 100910 (2025).
6. Shaheen, B.W., Németh, I.: Integration of maintenance management system functions with Industry 4.0 technologies and features—A review. *Processes* 10, 1–20 (2022).
7. Ismail, Z.A.: The requirements for maintenance management systems (MMS) at Malaysian polytechnic: a case study. *Journal of Quality in Maintenance Engineering* 28(2), 291–304 (2022).
8. Tavassoli, L.S., Sakhavand, N., Fazeli, S.S.: Integrated preventive maintenance scheduling model with redundancy for cutting tools on a single machine. *Technology & Applied Science Research* 10, 1–10 (2020).
9. Ma, Y., Zhang, W., Branke, J.: Multi-objective optimisation of multifaceted maintenance strategies for wind farms. *Journal of the Operational Research Society* 74(5), 1362–1377 (2023).
10. Cacereno, A., Greiner, D., Galván, B.J.: Multi-objective optimum design and maintenance of safety systems: An in-depth comparison study including encoding and scheduling aspects with NSGA-II. *Mathematics* 9(15), 1–20 (2021).
11. Rahimi, I., Kianfar, F., Poursalehi, N., Abbasi, M.: Scheduling by NSGA-II: Review and bibliometric analysis. *Processes* 10(1), 98 (2022).
12. Mohammed, B., Nouredine, R., Nouredine, M.: Using PROMETHEE and TOPSIS for prioritizing maintenance actions. In: *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pp. 1–10. IEOM Society International (2020).
13. Shih, H.S., Shyur, H.J., Lee, E.S.: An extension of TOPSIS for group decision making. *Mathematical and Computer Modelling* 45(7–8), 801–813 (2007).
14. Wang, C.N., Hsueh, M.H., Tran Thi, D.O., Le, T.D.M., Dinh, Q.T.: Optimal maintenance strategy selection for oil and gas industry equipment using a combined AHP–TOPSIS approach: A case study. *Processes* 13(5), 1–18 (2025).
15. Souza, A.C.A., Silva, P.R.N.: Application of artificial intelligence in reliability engineering and predictive maintenance: A case study in the mining industry. *Revista Ibero-Americana de Humanidades, Ciências e Educação* 10(10), 3646–3659 (2024).
16. Khan, A., Maity, K.: Application of MCDM-based TOPSIS method for the selection of optimal process parameters in turning of pure titanium. *Benchmarking* 24(7), 2009–2021 (2017).