Hybrid AHP - TOPSIS and Constraint Programming for Maintenance Prioritisation and Scheduling in Production Systems

Citation for the original published paper (version of record):
Aravind, V., Rajashekarappa, M., Namutebi, A. et al (2026). Hybrid AHP - TOPSIS and Constraint Programming for Maintenance Prioritisation and Scheduling in
Production Systems. IOP Conference Series: Materials Science and Engineering, 1342(1).
http://dx.doi.org/10.1088/1757-899X/1342/1/012068

Hybrid AHP - TOPSIS and Constraint Programming for Maintenance Prioritisation and Scheduling in Production Systems

To cite this article: Vishwas Aravind *et al* 2026 *IOP Conf. Ser.: Mater. Sci. Eng.* **1342** 012068

IOP Conf. Series: Materials Science and Engineering **1342** (2026) 012068 doi:10.1088/1757-899X/1342/1/012068

Hybrid AHP - TOPSIS and Constraint Programming for Maintenance Prioritisation and Scheduling in Production Systems

Vishwas Aravind[1]*, Mohan Rajashekarappa[1]*, Alice Namutebi[2], Roger Burman[2], Marcus Ljung[2], Akshay Bhat[1] and Ebru Turanoglu Bekar[1]
[1]Mechanical Engineering, Chalmers University of Technology, Gothenburg, Sweden
[2]Husqvarna AB, Jönköping, Sweden

*E-mail: vishwasa@chalmers.se, rmohan@chalmers.se

Abstract. Efficient maintenance planning is essential for sustaining productivity and minimizing downtime in industrial production. It ensures equipment reliability, cost efficiency, and stable production flows within modern manufacturing systems. Traditional scheduling approaches often depend on manual expertise and static rules, which limits their scalability in dynamic industrial environments. At the same time, modern manufacturing practices are becoming increasingly complex, as maintenance planning, equipment, production schedules, and resource management are tightly interlinked. Maintaining reliability while reducing costs and downtime poses a significant challenge, especially under Industry 4.0, where digital integration enables more reliable and data-driven decision-making. Existing studies overlook practical constraints like technician availability, cost variations across shifts, and the alignment of maintenance with production schedules and resources. This gap often results in decision-making processes that are either theoretically sound but operationally impractical, or operationally feasible but lacking robust prioritization logic. This paper proposes a hybrid decision support system that integrates the Analytical Hierarchy Process (AHP), the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS), and Constraint Programming to provide a structured approach to maintenance prioritization and planning. The system was implemented as an interactive dashboard and validated through a real-world manufacturing case study. Results show that the AHP-TOPSIS model enables transparent prioritization across risk, cost, and downtime dimensions, while Constraint Programming generates conflict-free schedules that consider technical skills, production cycles, and resource constraints. This integrated approach bridges the gap between expert-driven prioritization and real-time operational planning, offering a scalable and replicable methodology for effective maintenance management.

1. Introduction

Maintenance has evolved from a reactive function into a strategic enabler of productivity and competitiveness that helps in effective planning, reducing costs, mitigating risks and enhancing reliability [1]. Maintenance planning has anticipated to benefit significantly from emerging technologies and improved decision-support systems with the use of data-driven analysis in increasingly digitalized environments [2]. This highlights the growing importance of systematic maintenance prioritization approaches capable of integrating expert judgment with data to support timely and effective maintenance decisions [3]. Recent research has increasingly applied Multi-Criteria Decision-Making (MCDM) techniques, particularly the Analytic Hierarchy Process (AHP) [4] and the Technique for Order Preference by Similarity to Ideal Solution (TOPSIS) [5], to equipment and maintenance prioritization problems [6-10]. These methods are commonly used to rank assets or maintenance actions based on multiple, often conflicting criteria such as criticality, risk, cost, and condition. In parallel, optimization-based approaches such as Goal Programming and Constraint Programming have been widely adopted to address maintenance and operational scheduling under complex resource, precedence, and temporal constraints [11-14]. However, most existing studies treat prioritization and scheduling as separate decision layers. Although some works propose integrated maintenance and scheduling models [16-18], prioritization is typically embedded implicitly within optimization formulations rather than explicitly linked to established MCDM techniques. Consequently, a gap remains in integrated, deployable frameworks that couple MCDM-based prioritization with real-world scheduling for industrial applications.

Therefore, this study addresses that gap by integrating AHP and TOPSIS with Constraint Programming into a unified, hybrid decision support methodology. The approach is implemented as an interactive dashboard developed and validated using a real-world manufacturing use case, which aims to create a data-driven framework for maintenance prioritization and scheduling under real production constraints. Key contributions include: (i) A replicable prioritisation framework for risk, cost and downtime trade-offs. (ii) A constraint-based scheduling algorithm that integrates production and resource realities, and (iii) An industry validated dashboard that demonstrates operational feasibility.

The remainder of this paper is structured as follows: Section 2 provides a brief overview of related work regarding maintenance prioritisation and scheduling. Section 3 details the proposed methodology and implementation. Section 4 presents the key findings and results, while Section 5 summarizes a discussion of the broader implications of the study. Finally, Section 6 concludes the paper and outlines future research.

2. Related Work

2.1 Maintenance prioritisation with AHP-TOPSIS

Maintenance prioritization has been widely studied as a MCDM problem due to the need to balance technical, economic, safety, and operational factors [1]. Traditional prioritization approaches often relied on single-criterion indicators such as failure frequency or downtime; however, these methods have been shown to be insufficient for capturing the multidimensional nature of maintenance decision-making [9]. As a result, MCDM techniques have been used in maintenance to help decision-makers take an action based on criteria conflicting with each other [17]. Among MCDM techniques, the AHP is one of the most established, providing a systematic way to translate subjective evaluations into a coherent quantitative decision-making model [17]. AHP is particularly useful where expert judgement plays a crucial role in allowing decision-makers to express preferences through pairwise comparisons [4]. These comparisons are then used to calculate relative weights for each criterion. AHP is simple to apply and easy to understand, which makes it especially popular in industry-focused studies [4].

AHP has successfully been applied as a prioritisation tool, particularly in the context of maintenance decision making, to support selection and ranking of maintenance strategies based on expert judgement [6]. Maletic et al. [1] demonstrates its ability to evaluate and compare maintenance policies in paper a manufacturing industry. Their work shows that AHP not only improves the transparency of decision making but also allows managers to trace the rationale behind selected maintenance policies, which is vital for continuous improvement initiatives. Further, Maletić et al. [6] and Al-Najjar and Alsyouf [7] extend the use of AHP to machine criticality assessment and selection of efficient maintenance approaches, confirming its applicability across various industrial sectors. AHP has been used to assess equipment criticality by incorporating criteria such as failure consequences, safety impact, maintenance cost, and operational importance [9]. While AHP offers transparency, ease of interpretation, and a structured foundation for weighting criteria, it is limited in its ability to generate rankings among multiple alternatives, and its reliance on expert judgment introduces subjectivity, which can affect the consistency and robustness of the results, particularly in large-scale systems [8]. To overcome this limitation, many studies have proposed hybrid MCDM frameworks combining AHP with the TOPSIS [8, 10, 18]. In these AHP–TOPSIS approaches, AHP is typically employed to compute the relative importance of criteria, while TOPSIS is used to rank maintenance alternatives. TOPSIS quantifies closeness of each alternative to an ideal solution, using geometric distances to measure performance trade-offs [5]. The alternative closest to the ideal and furthest from the negative ideal is considered the most favourable [5]. This approach is intuitive and easy to compute, especially when the criteria weights are already known, which is why it is often used in combination with AHP and has been widely adopted in recent years for maintenance prioritization across different domains [10, 18]. Therefore, in this study, we integrate AHP and TOPSIS using the strengths of both methods, i.e., AHP systematically captures expert judgment and determines criterion importance, while TOPSIS uses these weights alongside performance data to produce more robust rankings. Such prioritization outcomes are particularly valuable when supporting resource-constrained maintenance scheduling, where task ranking and resource allocation depend directly on a reliable and well-differentiated priority order, as is often the case in real-world manufacturing environments.

2.2 Maintenance scheduling with Constraint Programming

Unlike heuristic or linear programming methods, Constraint Programming defines problems through a set of variables, domains and constraints, allowing highly flexible modelling of real-world dependencies such as limited technician availability, overlapping maintenance windows and resource conflicts [19, 20]. Constraint Programming has gained recognition for its ability to generate conflict free schedules while respecting production constraints particularly in an industrial environment [21]. Manzani et al. [22] applied Constraint Programming to maintenance scheduling and demonstrate its strength in handling non-linear and interdependent constraints that traditional optimisation models often overlook. More recent research highlights the abilities of Constraint Programming to be scalable and adaptable for a more dynamic environments, where production load and maintenance windows change constantly [13-14].

It should be noted that by integrating Constraint Programming with results from AHP and TOPSIS, this study extends these earlier contributions in the literature by transforming prioritised maintenance tasks into executable maintenance schedules. The hybrid framework thereby enables a smooth transition from existing prioritisation to operational feasibility. Together, forming a robust framework that can address both decision level and execution level challenges in modern maintenance planning.

3. Methodology

The proposed hybrid framework integrates two components: (1) a prioritisation stage using AHP-TOPSIS to rank maintenance tasks, and (2) a scheduling stage using Constraint Programming to allocate tasks within production constraints. Figure 1 illustrates the sequential steps of the proposed hybrid framework.

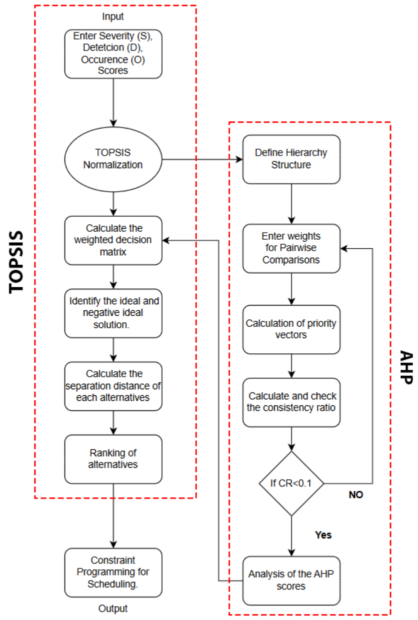


Figure 1. Proposed hybrid AHP-TOPSIS and Constraint Programming framework. Outlines in red distinguish TOPSIS and AHP methods.

As depicted in Figure 1, the steps involved in the AHP are described below [4].

Step 1 - Define the hierarchical structure): The hierarchy is constructed with the goal at the top (e.g. optimal maintenance prioritisation), followed by the criteria and sub-criteria levels, and finally the alternatives at the bottom for policy selection.

Step 2 - Pairwise comparison of criteria: Each pair of criteria is compared with respect to their relative importance using a 1-9 scale.

Step 3 - Normalise the comparison matrix: Each element is divided by the sum of its column, and the average of the rows is calculated to get the priority vector, representing the weights of the criteria.

Step 4 - Check consistency ration: AHP includes a consistency check to ensure logical consistency in judgments.

Step 5 - Finalise criteria weights: If the consitencey ration is within an acceptable range, the derived priority vector is used as the criteria weights for TOPSIS. Otherwise, the comparisons must be revised.

As depicted in Figure 1, the TOPSIS follow the steps outlined below [5].

Step 1: Construct the decision matrix to capture quantitative performance of maintenance tasks across all criteria.

Step 2: Construct a normalised decision matrix.

Step 3: Calculate the weighted decision matrix.

Step 4: Determine ideal and negative-ideal solutions.

Step 5: Calculate the separation distance of each competitive alternative from the ideal and negative ideal solution.

Step 6: Calculate relative closeness and rank the alternatives accordingly.

Using Constraint Programming has several advantages in maintenance scheduling. It allows for the integration of a wide variety of constraints within a single framework, enabling the model to replicate complex real-world scenarios. The constraints in this algorithm are flexible and can be dynamically adjusted based on changing operational conditions. Furthermore, the model can be tailored to focus on specific objectives, such as minimising cost, reducing downtime, or addressing high-risk components, depending on the organisation's priorities. In industrial maintenance planning, several practical constraints must be considered. Table 1 summarizes the typical constraints commonly encountered in such environments.

Table 1. Summary of common operational and resource constraints in industrial maintenance.

Constraints	Descriptions
Schedule window	Maintenance tasks can ideally be performed when the machine or tool is idle, leaving us with specific windows of downtime in the production schedule to perform these tasks [23].
Resource availability	The availability of spare parts, technicians, tools and other resources could often be limited and must be considered during planning a maintenance schedule and does not exceed the available capacity during any given period [24].
Interdependencies	Some tasks may sometimes depend on other tasks to be performed beforehand, and the duration of these tasks may also vary depending on other constraints and increasing complexity [23].
Distribution of work	Distributing maintenance tasks across available timeframes ensures that technicians or shifts are overworked, reducing stress on technicians and improving resource utilisation [25].

3.1 Data collection

Data were collected directly from the case company's operations to develop a realistic and context-specific framework. The study focused on a single manufacturing cell comprising eight interdependent machines. To ensure that the prioritisation and scheduling models reflected the actual operational context, data were collected from multiple sources.

- Interviews were conducted with maintenance engineers. These sessions provided valuable insights into machine functionality, failure modes, and the significance of various components within the production-flow.
- Historical work orders and maintenance logs were examined. These documents offered detailed records of past maintenance activities, including part replacements, downtime,

task frequencies, and downtime durations. Analysing this data allowed for the estimation of average downtime associated with specific component failures.

- Component-level cost data were obtained from the company's procurement and inventory systems. These included the cost of spare parts and, in some cases, estimated labour requirements. The cost information was essential in assessing the economic implications of maintenance scheduling decisions.

To enable structured and scalable prioritisation, all subcomponents were mapped and categorised to a high-level component category. This grouping was necessary to simplify and aggregate the sub-components while maintaining clarity in the maintenance plan. The classification was based on several factors, including the terminology in part descriptions, the functional role of the subcomponent, and common industry references. Subcomponents sharing similar roles or failure behaviours were grouped under the same functional category.

Given that downtime was the primary factor influencing task prioritisation in this study, a structured categorisation of downtime events was essential. The categorisation was performed inductively based on historical logs, followed by validation through internal expert discussions. The aim was to convert qualitative downtime severity into a weighted numerical format suitable for prioritisation in the decision support system. Downtime is classified into three categories with corresponding weights: minor stoppages of less than 2 hours are assigned a weight of "3", major downtimes of 2 to 5 hours are assigned a weight of "6", and critical downtimes exceeding 5 hours are assigned a weight of "8". These weights were chosen according to three main criteria [26]. First, fairness ensures that each severity category contributes appropriately to the total priority score, preventing overemphasis on rare but severe events. Second, interpretability aligns the weighting with the intuitive understanding of technicians and planners regarding downtime severity. Finally, responsiveness maintains a simple yet sufficiently refined scale, allowing the model to distinguish effectively between tasks of varying complexity without becoming overly complicated.

3.2 Dashboard design and implementation

A user-friendly dashboard was developed using the Streamlit framework in Visual Studio Code. This dashboard served as the central interface through which maintenance engineers and planners could interact with the prioritisation and scheduling models. The decision-support dashboard developed in this study integrates all analytical components into a cohesive user interface. Its primary function is to serve as a bridge between the technical logic underpinning prioritisation and scheduling and the real-world decisions made by maintenance planners.

The dashboard includes a pairwise comparison interface for AHP inputs, allowing users to enter their judgements directly into a matrix. This interface instantly computes priority weights and checks for consistency using Saaty's consistency ratio metric [4], providing feedback if the inputs deviate from acceptable thresholds. This ensures that users maintain logical coherence in their decision-making without needing to recalculate manually. Another notable feature is the risk input interface, which lets users manually enter S, O and D values that contribute to the overall RPN score. By doing so, the system accommodates expert judgement and context-specific insights, enabling risk evaluations to reflect current operational realities rather than static historical data. Once the weights for AHP are calculated, the component rankings via TOPSIS are generated and the dashboard presents a ranked list of components in a dynamic table. This list provides full visibility into how each component scored against all criteria, and users can explore which factors influenced a component's prioritisation. For scheduling, the dashboard integrates a backend call to the Z3 solver [27]. Upon initiating the scheduling process, users receive a detailed output specifying which tasks are scheduled in which week, day, and shift. These outputs are clearly formatted and easy to interpret, aiding maintenance planning at the tactical level. Finally, the dashboard features several visualisation tools, including bar charts and summary tables. These visuals enhance user understanding of scheduling patterns, technician distribution, and overall task criticality, supporting faster and more informed decision-making. The dashboard is designed primarily for internal maintenance planners and operations engineers. By integrating multiple decision layers into a single tool, it enhances transparency, reduces manual planning errors, and supports timely, data-driven decisions. The entire process is displayed in an intuitive dashboard that is tailored to partner company's needs.

4. Results

The dashboard was implemented using the Streamlit library to provide an interactive interface that enables users to input data for algorithmic analysis, prioritization, and scheduling of maintenance tasks. Its design was guided by stakeholder requirements to ensure usability, relevance, and alignment with operational needs. Upon launching the web application, as shown in Figure 2, the user is greeted with a simple dropdown menu to select a 'Global Primary Criterion' to choose between 'Risk', 'Downtime' and 'Cost' depending on the background of the user. This forms the initial basis for how weights are assigned during the AHP pairwise comparison. In the 'Pairwise Comparison' section, the users input their expertise about relative importance of these criteria.

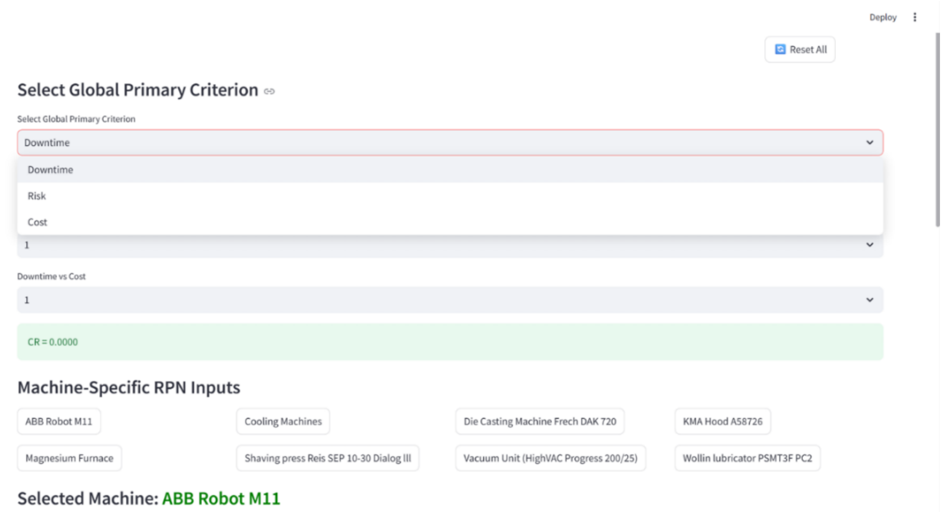


Figure 2. Select global primary criterion and machine-specific RPN inputs in the dashboard.

The dashboard dynamically generates additional comparison drop-downs, as shown in Figure 3, based on the prioritisation values assigned by the user, while ensuring the resulting 'CR' remains below 10% or 0.10 ensuring the expert input remains consistent. The automatic generation of comparisons removes the burden on users to perform complex calculations to ensure the consistency ratio < 0.10. The risk scoring module is a key feature where users can customise S, O and D levels for each machine. These values are used in RPN calculations and can be adjusted on a scale from 'Minor' to 'Critical', enabling flexible evaluation of operational risks, as shown in Figure 4 and Figure 5. Once RPN scores are finalised, users can trigger the TOPSIS algorithm.

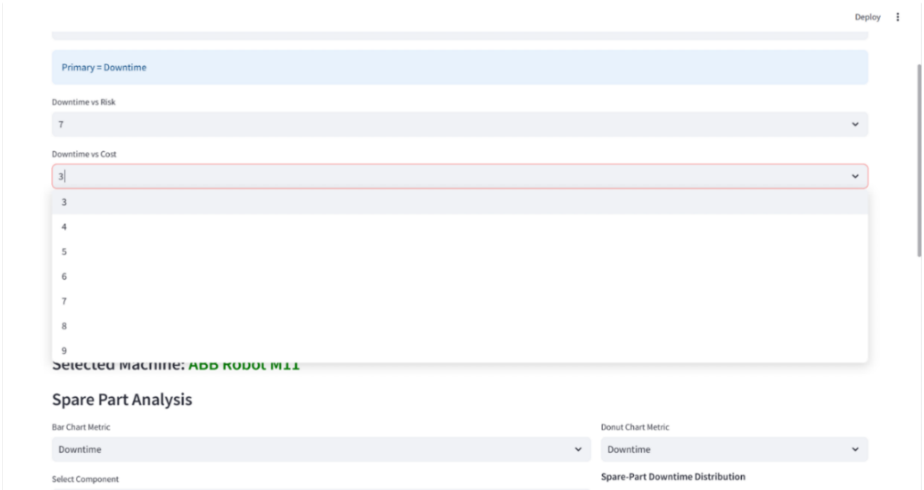


Figure 3. Pairwise Comparison Section with dynamically generated dropdowns.

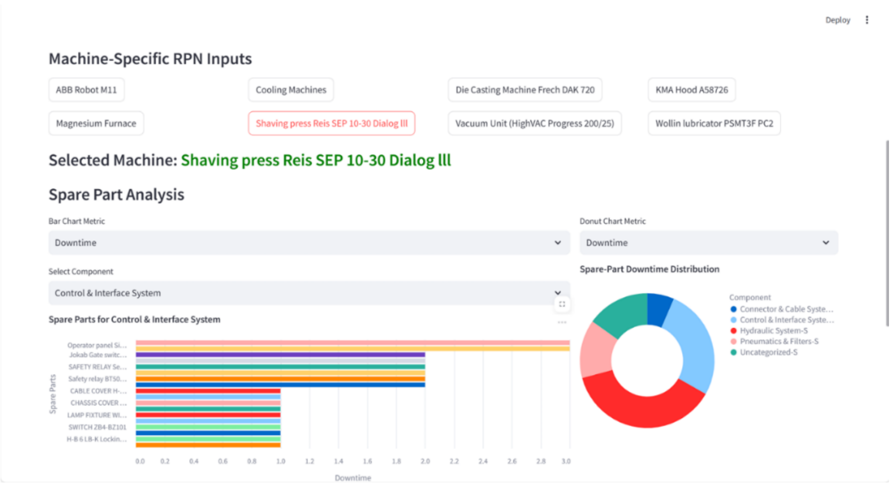


Figure 4. An overview of machine spare parts analysis for informed decision making.

The algorithm ranks subcomponents based on criticality and presents a sorted list with closeness coefficients, offering transparency on how each component performed across the criteria. As Tzeng and Huang [17] emphasise, MCDM methods act as bridges between domain knowledge and quantitative logic. This dashboard embodies those principles through a robust, transparent, and scalable tool for modern industrial maintenance planning.

IOP Conf. Series: Materials Science and Engineering **1342** (2026) 012068 doi:10.1088/1757-899X/1342/1/012068

The dashboard integrates a backend Z3-based constraint programming algorithm [27] that converts prioritised tasks into a feasible maintenance schedule aligned with production constraints. Upon clicking 'Calculate', the model evaluates multiple parameters, such as shifts, holiday periods, technician availability, downtime windows, production load, and other constraints to generate an optimised task schedule organised by component, day, shift, and technicians, Figure 6.

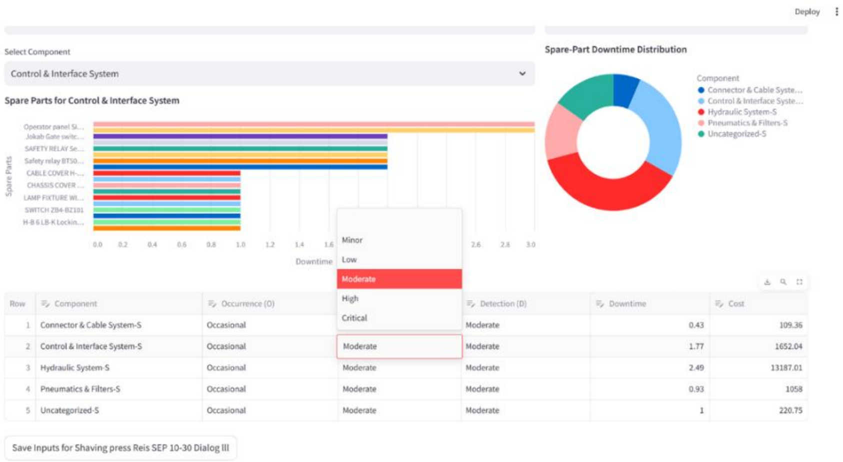
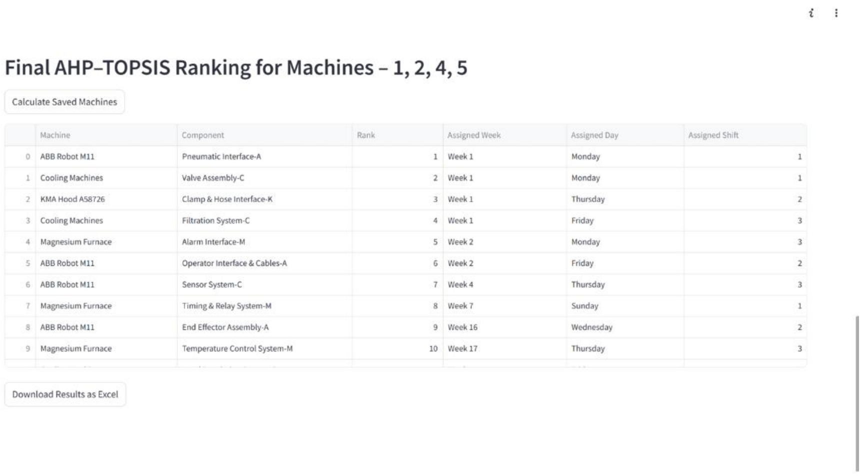


Figure 5. Machine specific RPN inputs and spare part analysis interface.



	Machine	Component	Rank	Assigned Week	Assigned Day	Assigned Shift
0	ABB Robot M11	Pneumatic Interface-A	1	Week 1	Monday	1
1	Cooling Machines	Valve Assembly-C	2	Week 1	Monday	1
2	KMA Hood A58726	Clamp & Hose Interface-K	3	Week 1	Thursday	2
3	Cooling Machines	Filtration System-C	4	Week 1	Friday	3
4	Magnesium Furnace	Alarm Interface-M	5	Week 2	Monday	3
5	ABB Robot M11	Operator Interface & Cables-A	6	Week 2	Friday	2
6	ABB Robot M11	Sensor System-C	7	Week 4	Thursday	3
7	Magnesium Furnace	Timing & Relay System-M	8	Week 7	Sunday	1
8	ABB Robot M11	End Effector Assembly-A	9	Week 16	Wednesday	2
9	Magnesium Furnace	Temperature Control System-M	10	Week 17	Thursday	3

Figure 6. Generated maintenance schedule from Constraint Programming algorithm.

5. Discussion

The integration of AHP-TOPSIS and Constraint Programming demonstrated clear advantages over conventional approaches. A systematic and scalable approach to optimizing maintenance activities is to prioritize tasks according to their criticality, which was achieved using the hybrid framework. This allows components to be ranked based on failure risk, downtime impact, costs, and labor requirements, employing pairwise comparisons through AHP and evaluation via TOPSIS. Once the weights are established from AHP, TOPSIS algorithm computes the closeness coefficient of each component, maintenance task to an ideal solution. These ratings are then presented in a ranked format that enables the user to identify critical components and areas that require immediate attention. This demonstrates that MCDM techniques can support structured and informed decision-making in maintenance planning. While prior studies have individually applied and discussed AHP or TOPSIS as maintenance prioritisation techniques, this study presents a novel integration with a Constraint Programming algorithm. This hybrid combination not only allows for a nuanced task ranking but also translates this ranking into realistic operationally feasible scheduling, helping in data-driven decision making, bridging the gap in existing literature [3].

Additionally, a user-friendly decision support system was developed as an interactive dashboard built using Streamlit [28]. The dashboard integrates Z3 based Constraint Programming algorithm [27] that logically prioritises, and schedules maintenance tasks based on real world constraints such as shift limits, overtime, resource availability, costs, downtime and production schedule. The dashboard also translates historical data into interactive graphs, allowing users to identify trends. After expert inputs are provided, the dashboard generates an optimised schedule at the click of a button, ensuring that user intuition is included in a structured decision-making process.

Theoretically, this study highlights the significance of the hybrid MCDM approach for strategic maintenance scheduling and planning. The model shows how a hybrid MCDM model, along with Constraint Programming scheduling to improve both strategic and operational aspects of scheduling maintenance activities in a dynamic environment. Practically, the decision support system was designed for scalability and ease of use. Moreover, the integration of an interactive dashboard enables intuitive use by both technical and non-technical personnel, empowering maintenance teams to make timely, data-driven decisions. The results align with the human centric digitalisation framework proposed by Rajashekarappa et al. [29], which stresses the integration of operator expertise with digital tools for developing condition-based maintenance.

It should also be noted that while the proposed hybrid framework demonstrates clear practical value, this study is subject to certain limitations including its reliance on historical risk data and the manual input of severity, occurrence, and detection scores. In addition to this, the validation was conducted using a single industrial case study within a specific manufacturing context. Although this allowed for a realistic evaluation of the framework under real production constraints, the results may not be directly generalizable to all industrial settings. Differences in production layouts, maintenance policies, workforce structures, and asset characteristics may influence both prioritization outcomes and scheduling feasibility. Consequently, the findings should be interpreted as indicative rather than universally representative. Nevertheless, the proposed framework has been intentionally designed with transferability in mind. The modular structure of the methodology allows each component, e.g., AHP-based weighting, TOPSIS-based ranking, and Constraint Programming-based scheduling to be adapted independently to different industrial contexts. Criteria definitions, weighting schemes, and constraint formulations can be modified to reflect alternative production systems, maintenance strategies, or organisational priorities without altering the underlying framework. This flexibility enables the approach to be applied to a wide range of manufacturing environments. Therefore, future research can focus on validating the framework across multiple industrial cases and sectors to further assess its robustness and generalizability. Applying the approach in different operational contexts would enable comparative analysis and provide deeper insights into how contextual factors influence prioritization and scheduling outcomes. Such extensions would strengthen the empirical foundation of the framework and support its broader adoption as a scalable decision-support tool for industrial maintenance planning.

6. Conclusion

This study proposed a framework successfully integrates a hybrid MCDM approach, combining AHP and TOPSIS with Constraint Programming to generate maintenance schedules that are both strategically prioritized and operationally feasible. By using both user inputs and quantitative machine data, the system enables maintenance teams to make informed, data-driven decisions that account for risk, cost, technician availability, and production cycles. Moving beyond predominantly reactive maintenance practices, such decision support system built upon an interactive dashboard enhances transparency and introduces a structured prioritization process that balances technical criticality with operational constraints. Through AHP, operators can formalize their expertise and preferences using weights and pairwise comparisons, while TOPSIS translates these inputs into a ranked list of tasks based on proximity to an ideal solution. The Constraint Programming algorithm then transforms these rankings into feasible maintenance schedules that respect downtime windows, resource availability, and cost considerations. By combining structured prioritization with actionable scheduling, this study contributes a scalable, replicable framework that supports data-driven maintenance management and has the potential for broader application across complex production systems.

Acknowledgements

This study is based on the Master's thesis conducted by the first author at Chalmers University of Technology. The authors gratefully acknowledge the support of the Advanced and Innovative Digitalization Program, funded by VINNOVA, through the research project TPdM - Trustworthy Predictive Maintenance (Grant No. 2022-01710). The study was carried out within the Production Area of Advance at Chalmers University of Technology. The authors also extend their appreciation to the personnel from the Maintenance department of the case company for their valuable support and assistance with data and domain expertise.

References

[1] Maletić D, Maletić M and Gomišček B 2014 Int. J. Prod. Res. 52 174–188
[2] Bokrantz J, Skoogh A, Berlin C and Stahre J 2017 Int. J. Prod. Econ. 191 154–169
[3] dos Santos C F, Loures E D F R and Santos E A P 2025 Int. J. Adv. Manuf. Technol. 136 3971–3987
[4] Saaty T L 1988 What is the analytic hierarchy process? (Berlin: Springer) pp 109–121
[5] Hwang C L and Yoon K P 1981 Multiple attribute decision making: methods and applications (New York: Springer)
[6] Maletić D, Maletić M, Lovrencic V, Al-Najjar B and Gomišček B 2014 Organizacija 47 177–186
[7] Al-Najjar B and Alsyouf I 2003 Int. J. Prod. Econ. 84 85–100
[8] Shyjith K, Ilangkumaran M and Kumanan S 2008 Reliab. Eng. Syst. Saf. 93 791–802
[9] Bouchaala A M and Noureddine R 2020 Int. J. Decis. Support Syst. Technol. 12 26–42
[10] Liu Y et al 2024 Buildings 14 3048
[11] Özcan E, Toklu B and Sönmez E 2017 Energy 134 1021–1033
[12] Özcan E C, Ünlüsoy S and Eren T 2017 Renew. Sustain. Energy Rev. 78 1410–1423
[13] Cho J et al 2024 Appl. Sci. 14 11396
[14] Sciau J B et al 2024 J. Air Transp. Manag. 115 102537

[15] Geurtsen M, Adan J and Akçay A 2023 Flex. Serv. Manuf. J. 1–34
[16] Zhang W et al 2024 Reliab. Eng. Syst. Saf. 247 110129
[17] Tzeng G H and Huang J J 2011 Multiple attribute decision making: methods and applications (Boca Raton: CRC Press)
[18] Wang C N et al 2025 Processes 13 1389
[19] Frost D and Dechter R 1998 Proc. CP p 469
[20] Rossi F, Van Beek P and Walsh T (eds) 2006 Handbook of constraint programming (Elsevier)
[21] Schlenkrich M and Parragh S N 2023 Procedia Comput. Sci. 217 1028–1037
[22] Manzini R et al 2015 Comput. Ind. Eng. 87 561–568
[23] Niu G, Yang B S and Pecht M 2010 Reliab. Eng. Syst. Saf. 95 786–796
[24] Jardine A K, Lin D and Banjevic D 2006 Mech. Syst. Signal Process. 20 1483–1510
[25] Muchiri P, Pintelon L, Gelders L and Martin H 2011 Int. J. Prod. Econ. 131 295–302
[26] Vishwas A and Akshay B 2025 MSc Thesis Chalmers Univ. Technol.
[27] Zheng Y, Zhang X and Ganesh V 2013 Proc. ACM SIGSOFT FSE pp 114–124
[28] Khorasani M, Abdou M and Hernández Fernández J 2025 Streamlit for web development (Berkeley, CA: Apress) pp 295–348
[29] Rajashekarappa M et al 2025 IFAC-PapersOnLine 59 2557–2562