Journal of Intelligent Information Systems
https://doi.org/10.1007/s10844-025-00940-w

RESEARCH

Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models

Cheyenne Powell[1] · **Annalisa Riccardi**[1]

Received: 21 November 2024 / Revised: 7 April 2025 / Accepted: 8 April 2025

Abstract
Scheduling systems are critical for planning projects, resources, and activities across many industries to achieve goals efficiently. As scheduling requirements grow in complexity, the use of Artificial Intelligence (AI) solutions has received more attention. However, providing comprehensible explanations of these decision-making processes remains a challenge and blocker to adoption. The emergent field of eXplainable Artificial Intelligence (XAI) aims to address this by establishing human-centric interpretation of influencing factors for machine decisions. The leading field of autonomous interpretation in Natural Language Processing (NLP) is Large Language Model (LLM)s, for their generalist knowledge and reasoning capabilities. To explore LLMs' potential to generate explanations for scheduling queries, we selected a benchmark set of Job Shop scheduling problems. A novel framework that integrates the selected language models, GPT-4 and Large Language Model Meta AI (LLaMA), into scheduling systems is introduced, facilitating human-like explanations to queries from different categories through few-shot learning. The explanations were analysed for accuracy, consistency, completeness, conciseness, and language across different scheduling problem sizes and complexities. The approach achieved an overall accuracy of 59% with GPT-4 and 35% with LLaMA, with minimal impact from the varied schedule sizes observed, proving the approach can handle different datasets and is performance scalable. Several responses demonstrated high comprehension of complex queries; however, response quality fluctuated due to the few-shot learning approach. This study establishes a baseline for measuring generalist LLM capabilities in handling explanations for autonomous scheduling systems, with promising results for an LLM providing XAI interactions to explain scheduling decisions.

Keywords Large language models · Natural language processing · Question answering systems · Scheduling · Explainable artificial intelligence

✉ Cheyenne Powell
cheyenne.powell@strath.ac.uk

Annalisa Riccardi
annalisa.riccardi@strath.ac.uk

[1] Mechanical and Aerospace Engineering, University of Strathclyde, 16 Richmond St., Glasgow G1 1XQ, Scotland, UK

Published online: 17 April 2025

1 Introduction

Scheduling systems play a critical role across many industries, including transportation, healthcare, and manufacturing (Atsmony & Mosheiov, 2022; Yao et al., 2020; Fikar & Hirsch, 2017; Zhou et al., 2020). The efficient allocation of resources and optimal sequencing of tasks is essential for achieving productivity, reducing costs, and improving overall operational performance (Zhou et al., 2020; Amer et al., 2022; Moons et al., 2017). As the use of automated systems has become increasingly common, this has led to greater use of AI Decision Making Systems (DMS)s to manage and improve scheduling capabilities for various purposes, such as construction planning, cloud computing maintenance, and medical treatments (Amer et al., 2021; Rjoub et al., 2021; Squires et al., 2022). DMS, in scheduling, refers to computational systems that assist in making intelligent decisions and generating optimized schedules for a given problem. These systems incorporate various algorithms, techniques, and models to analyze scheduling constraints, objectives, and resources to generate optimal schedules.

A 30 year study on automated scheduling within the construction industry found that by 2021 most, if not all, schedules were still created through fully manual processes in real world applications. Examining over 1500 articles the studies highlighted issues on the accessibility of relevant data which impacts a model ability to learn and the limited capacity human operators have to validate existing approaches with real-world projects. The study recommended the formalisation of methods and data, further extensive testing, and to integrate automated systems to maximise potential benefit (Amer et al., 2021).

In the space industry, a study into AI methodologies for the future of Low Earth Orbit (LEO) satellite arrays revealed the emergent capabilities for signal detection, network optimisation, and the potential for completely automated and robust systems (Al Homssi et al., 2024).

Additional research, targeting the optimisation (Goh et al., 2022) and functional advancement (Herrmann & Schaub, 2023) of satellite scheduling, identified the benefits in performance and scalability of utilising automated systems but also highlighted the dedicated training required to achieve performance reduced effectiveness in flexible or generalised applications and the integration of new data and concepts.

There has also been research reviewing the challenges in human operators managing more advanced AI systems in time-sensitive missions as a result of the size of data being generated and assessed by the systems (Thangavel et al., 2023). Further concerns are present with trust and transparency in autonomous system decisions and the factors that influence these decisions, as the potential effect on mission outcomes can depend on the operator's ability to validate results as accurate or whether they require modification (Picard et al., 2021).

Moreover, a study into the fairness, or perception thereof, of automated decision-making system for nurse shift and duty scheduling was conducted. The study found that staff were much more accepting of decisions when involved in the planning and advised that this be factored into the operation of the model. There were issues highlighted, however, for potential bias applied by the model around need-based factors, such as family commitments, and these would need to be captured and trained out, advising that great care must be taken when working with automated systems that impact a person's wellbeing (Uhde et al., 2020).

These challenges in managing and understanding AI systems and their decisions have led to the emergence of the Explainability measure of a system, which intends to provide explanations on how and why decisions were made (Yang et al., 2024a). The means of establishing ways for an autonomous system to provide explanations on the reasoning behind its decisions and outputs is a new field of AI, known as XAI (Saeed & Omlin, 2023). Though

a dedicated process, XAI is designed to capture and present information behind decisions and reasoning in formats such as natural language, example-based and graphical diagrams (Lai et al., 2021, 2023), for example to identify the most important factors contributing to healthcare professional burnout by highlighting feature importance (Pillai et al., 2024). This process can be completed through a number of methods, categorised into data, model or post-hoc explainability approaches, including explanatory data analysis, joint prediction and explanation, attribution methods, and knowledge extraction methods, to name only a few (Ali et al., 2023; Arrieta et al., 2020). Language use and quality are also inherently critical to how XAI performs and, therefore, is deeply connected with the field of NLP.

Language modelling is a fundamental task in NLP that aims at capturing the statistical patterns and structures within a given language. This approach, in its most common form, involves training a model to predict the next word or sequence of words in a sentence, using the surrounding context as a guide. By learning these patterns, Language Models (LMs) can generate coherent and contextually relevant text, complete sentences, and even perform various language-related tasks (Brown et al., 2020).

LLMs, such as the Generative Pre-trained Transformer (GPT) series, have shown exceptional proficiency in understanding and generating human language. LLMs, were trained on vast amounts of text data and employed deep learning techniques, including self-attention mechanisms and the transformer architecture, to learn rich linguistic patterns and contextual dependencies (Vaswani et al., 2017; Brown et al., 2020). In recent years, the potential of these models has been demonstrated for various NLP tasks and varied technical domains, including text generation, summarization, and question-answering (Shuster et al., 2022; Scao et al., 2022; Glaese et al., 2022; Thoppilan et al., 2022; Powell et al., 2023).

The application of LLMs offers a tremendous opportunity to enhance scheduling systems by leveraging their advanced reasoning capabilities and generating textual explanations to justify the systems' outputs to smooth the decision-making processes, addressing the challenges in bias and data utilisation, and significantly improving operational efficiency (Zheng et al., 2023; Bastola et al., 2023; Schroder, 2023).

LLMs can be broadly categorised as either generalist, such as OpenAI's ChatGPT (Wu et al., 2023), or domain-specific, which are purpose-built or trained for a specific area of function (Jeong, 2024). Generalist LLMs are trained on large amounts of data on almost every publicly available topic and have the ability to comprehend a wide variety of queries. Whereas domain-specific LLMs, which can be developed on top of a generalist platform (Jeong, 2024), are designed to answer targetted questions on a single or small number of topics, with the intent of deepening the comprehension of the system; this is at the expense of flexibility, however, as mentioned earlier. Creating domain-specific capabilities from generalist LLMs requires the implementation of pre-training and fine-tuning data practices, which provides additional context to build domain knowledge in a specific area, such as business processes, finance management, and recommender systems aiming to improve performance and accuracy (Bernardi et al., 2024; Wang et al., 2024; Yang et al., 2024b).

This paper aims to explore the integration of a generalist LLM into benchmark schedules, focusing on their reasoning abilities and the generation of informative textual explanations to questions based on differently sized schedules in tabular form. To the author's knowledge, there is minimal available data in this area; however, two suitable studies have been identified. The first study assessed the performance of GPT models in answering questions on materials science. Introducing the concept of Material Science Question Answering (MaScQA), the study compared the performance of the models when using a zero-shot or a chain-of-thought prompting method (Zaki et al., 2024). The second study focused on assessing the use of

language achieved by GPT models in answering domain-specific questions on Microsoft products and technical IT problems, using reference answers curated by cloud-computing specialists. In this study, a small, domain-specific Language Model (LM) was pre-trained on the question dataset, which the LLM then was given access to when answering the question (Yang et al., 2023).

Likewise, LLaMA has also been extensively studied, often in combination with ChatGPT or GPT-4. For instance a recent study evaluated the clinical decision-making capabilities of LLaMA and found that the accuracy did not meet the human expert standard and could present a considerable risk to patients in real-world applications, even when trained on real clinical case data (Hager et al., 2024). Another study followed a similar process for assessing the performance of LLaMA in responding to case law enquries. Conversely to the performance on clinical decision-making, the post-tuned model demonstrated marked improvement in accuracy and F1-score (Satterfield et al., 2024). This highlights the challenge for generalist LLMs in responding to domain-specific topics, even when pre-training is performed, as results can be inconsistent.

While the combination of both DMS and NLP is yet to be widely applied to scheduling capabilities, it has shown potential in research on staff assignment, where an NLP prediction model was created to autonomously assign staff tasks using unstructured data and construction scheduling (Mo et al., 2020). Elsewhere, NLP was utilised to analyze and validate the logic of manually created schedules based on trained data (Amer et al., 2022) and the use of a GPT model to support the creation of construction schedules based on prompts (Prieto et al., 2023). These systems leverage NLP techniques by extracting and analyzing relevant information from textual data, enabling an automated understanding of scheduling requirements, constraints, and objectives expressed in natural language.

Several challenges need to be addressed to effectively leverage LLMs capabilities for scheduling systems. These include ensuring the robustness and reliability of the models' reasoning capabilities (Kasneci et al., 2023). There are also limitations in certain case studies using a GPT model, as previously mentioned, for instance, lacking domain-specific knowledge to develop construction project schedules due to generalised training and no specialised application existing (Prieto et al., 2023). Additionally, the ethical considerations necessary, such as fairness and data/training bias, were highlighted when using algorithmic DMS as the impact can often be beyond the organisation itself (Marabelli et al., 2021). Furthermore, the interpretability of LLMs, which stems from the lack of transparency in systems operations, is an area of active research that requires attention to ensure that the generated explanations are meaningful and trustworthy (Singh et al., 2023b).

The contributions of this paper are summarised as follows:

- We introduce a novel approach that utilizes LLMs to generate textual explanations for optimally derived solutions to scheduling problems. This approach leverages the LLMs' ability to comprehend tabular data and perform reasoning tasks.
- We propose a comprehensive framework for evaluating the methodology by creating a benchmark dataset that categorizes queries into three types: *Swap*, *Increase*, and *Decrease*. The generated explanations are assessed based on metrics such as correctness, similarity, completeness, conciseness, and language quality, demonstrating a high level of comprehension in the results.
- A benchmark dataset of job-shop scheduling problems was selected to test the proposed methodology. Performance was evaluated across different configurations, varying in the number and combination of jobs and machines, to validate the effectiveness of the approach.

Following this introduction, this paper covers a background and literature review (Section 2) exploring existing research in the relevant areas discussing relatable concepts. The methodology (Section 3) outlines the approach taken for generating queries (also referred to as questions) in each class, as well as the methodology used to engineer the prompt and generate a textual response. The results (Section 4) analyse the findings and discuss observed patterns and performance. Finally, the conclusion (Section 5) summarises the paper's findings and suggests future opportunities to deepen research in this area. Additionally, Appendix A contains tables used in completing assessments and summarising the scores from the LLMs.

2 Background and literature review

2.1 LLMs and XAI

LLMs are evolving in tandem with efforts to incorporate XAI techniques, aiming to improve the interpretability and transparency of these models. By integrating XAI methods, researchers strive to provide insights into the decision-making processes of autonomous systems and enable users to understand and trust their outputs (Datta & Dickerson, 2023; Narteni et al., 2022).

There are a multitude of different approaches in development for XAI, as shown in recent studies (Arrieta et al., 2020; Ali et al., 2023), each designed to address particular details within an AI system. Most XAI techniques identified do not utilise an LLM and instead directly interface or integrate into the AI model to extract the explanatory information. For instance, Exploratory Data Analysis (EDA) tools aim to extract significant features of a domain-specific model, where feature engineering is in place. Alternatively, a Teaching Explanations for Decisions (TED) framework could be applied, to supplement training data with user-based reasoning on a particular decision, which can then be combined with the output from the model; or, for post-hoc approaches, a a Local Rule-Based Explanation (LORE) can be implemented that extracts a decision tree from the model to infer the explanation by establishing the rules for what causes the decision to be made along with the conditions for a reversal of the decision. These techniques either return quantitive-based explanations or are built to be model-specific and do not offer easily understandable, human-language responses. Additionally, the explanations generated are single execution without the means of feedback or interaction, which gives inherent benefits of using an LLM for XAI for a model-agnostic and language-based approach. The most prominent disadvantage of implementing an LLM, especially a publicly accessible model, is the lack of insight into the training of the model and where bias or fairness concerns may not be known or corrected prior to implementation.

One of the critical aspects to maximise the potential of LLMs is to optimize prompting, with a leading technique being chain-of-thought prompting. The technique involves a series of structured textual reasoning steps that result in the final output (Wei et al., 2022). The intent of this process is to refine the performance from an LLM and generate better-reasoned results, with the capacity for the LLM to synthesize its own chain-of-thought prompting following an initial guiding prompt (Shao et al., 2023). However, research into the social bias of LLMs has also shown that chain-of-thought generated explanations can appear well-reasoned but actually contain misleading information, which inhibits the establishment of transparency and trust (Turpin et al., 2023).

To try and overcome the issues of fact hallucination and error propagation in chain-of-thought prompting, research was conducted to apply a more action focused approached

to answering queries. The approach, termed ReACT, creates a combination of reasoning traces and required actions to establish and adapt an executable plan to achieve the outcome, including the utilisation of external sources, aligning to the principles of *reason to act* and *act to reason* (Yao et al., 2023). Once this methodology is implemented, a generalist LLM, such as GPT-4, is able to complete highly technical and domain-specific activities, like root cause analysis of Information Technology (IT) incidents, demonstrating the potential for real-world applications (Roy et al., 2024). The capability of the ReACT framework has facilitated new investigations on how LLM reasoning can be improved further, such as the introduction of PreAct, which leverages environmental and situational predictions into the process to yield greater results in diverse environments (Fu et al., 2024).

Techniques like Layer-wise Relevance Propagation (LRP) provide explanations from the predictions of neural networks by assigning relevance scores to the input features. It aims to understand the importance of each input feature in contributing to the final prediction made by the model (Montavon et al., 2019). LRP works by attributing the model's predictions back to individual input features, providing insights into which parts of the input are most and least relevant for making a particular decision (Bach et al., 2015).

Researchers are exploring methods to build interpretable models by leveraging the knowledge learned from LLMs. Augmented Interpretable Models (Aug-imodel), a recently introduced technique that uses a LM to build an interpretable model but doesn't rely on the LLM during inference, ensuring transparency and efficiency gains in terms of speed and memory (Singh et al., 2023a). It addresses limitations in existing transparent models by incorporating world knowledge from modern LLMs, such as feature correlations. The method includes two approaches: Aug-GAM, which enhances a generalised additive model with LLM-based decoupled embeddings, and Aug-Tree, which improves a decision tree by generating enhanced features using an LLM.

Prototype networks for transformer language models, referred to as Prototypical-Transformer Explanation (Proto-Trex), have the aim of providing explanations for the network's decisions (Friedrich et al., 2021). The study demonstrated that these prototype networks performed on par with non-interpretable baselines for classification tasks across various architectures and datasets. To enhance prototypical explanations, they presented a novel interactive prototype learning setting named iProto-Trex, which took into account user feedback certainty.

The paper "Attention is not Explanation" discussed how attention mechanisms, commonly used in transformer-based models, do not serve as adequate explanations for model predictions. Their results suggested that relying solely on attention weights to interpret model behaviour may not provide meaningful insights into how the model arrives at its decisions (Jain & Wallace, 2019). Meanwhile, a paper by response, "Attention is not not Explanation", presents a counterargument to this claim, where they proposed four alternative tests to assess when and whether attention can be used as an explanation. These tests included a simple uniform-weights baseline, variance calibration based on multiple random seed runs, a diagnostic framework using frozen weights from pre-trained models, and an end-to-end adversarial attention training protocol. The authors aim to gain meaningful interpretations of attention mechanisms in Recurrent Neural Network (RNN) models (Wiegreffe & Pinter, 2019).

XAI in recommender systems, research was conducted with the aim to provide users insights into product recommendations (Kim et al., 2023). Their work emphasized the need for a unified explanation method centred around the human perspective. They later investigated user-centred explainability components, such as scope (global/local) and format (text/visualization), using a conjoint survey. Results showed a preference for local explanations and visualizations over global ones, while lengthy textual interfaces were disliked.

These examples represent the ongoing efforts to make LLMs more explainable and interpretable. By incorporating XAI techniques, researchers are working towards ensuring transparency and building trust in LLMs.

In selecting the best approach for this study, the ten strategies outlined in a recent study were considered (some of which are Explainability for Trustworthy LLMs and Human Alignment, LLM Enhancement via Explainable Prompting, and Generating User-Friendly Explanation for XAI) (Wu et al., 2024). As the goal of this study is to establish a benchmark approach for generating textual explanations, the user-friendly explanation approach was selected as the most appropriate. The authors encourage alternative approaches in future experiments.

2.2 Scheduling and XAI

There are valuable applications of XAI in the domain of scheduling offering transparent and comprehensible insights into the decision-making process behind scheduling tasks, however, research and development in this area are still emerging. Through the application of XAI techniques, users will gain a detailed understanding of the underlying logic of the scheduling model and the key factors that influence the generation of schedules (Ben Abdallah et al., 2023; Čyras et al., 2021; Gashi et al., 2023).

In the context of Machine Learning (ML), taking a specific classifier and point in the feature space, applying a rule-based explanation algorithm creates a rule that holds to the features of the classifier, covering the given point and enabling classification. These explanations are robust in the context of the surrounding area in the feature space (Mullins, 2023). Considering this concept for scheduling, human-readable rules are extracted from the scheduling model to provide understandable decision guidelines. Users can gain insights into how certain scheduling decisions are made based on these rules.

Integrating XAI with the scheduling model to incorporate user inputs and feedback may provide a clear understanding of how they influence the final schedule. By considering users' requirements, the scheduling algorithm prioritizes tasks or resources in alignment with individual choices, allowing users to comprehend the rationale behind the prioritization of specific elements in the schedule (Chakraborti et al., 2020).

In a recent study, a robust analysis of consumer preferences for AI interfaces was undertaken using a discrete-choice model grounded in random utility theory. Specifically, the researchers opted for a mixed logit model to effectively account for variations in consumer preferences and accommodate the inherent heterogeneity among users. This approach enabled a comprehensive evaluation of users' choices, facilitating a deeper understanding of the factors influencing their preferences for AI interfaces (Kim et al., 2023). The potential of this research could facilitate the development of interactive interfaces that allow users to explore different scheduling scenarios and understand the impact of their inputs with the help of graphical aids.

A comprehensive survey discusses practical applications of Reinforcement Learning (RL) methods to achieve fair solutions with high accuracy. The survey reviews the theory of fair reinforcement learning, including single-agent RL, multi-agent RL, long-term fairness via RL, and offline learning. Additionally, the authors highlight key issues to explore for advancing fair-RL, such as correcting societal biases, evaluating the feasibility of group fairness or individual fairness, and enhancing explainability in RL known as Explainable Reinforcement Learning (XRL) (Gajane et al., 2022). XRL is aimed at providing clear and

transparent insights into the decision-making process of learning agents, in particular for systems performing sequential decision-making (Puiutta & Veith, 2020).

The study observes the potential of fair XRL for scheduling, by incorporating fairness considerations into scheduling algorithms, users can gain transparent insights into how resources, including time, are allocated, leading to equitable distribution and mitigating biases (Puiutta & Veith, 2020). Further research on fair-RL and XRL techniques for scheduling is considered necessary for building trustworthy and inclusive scheduling systems that cater to diverse user needs.

Another approach is that of CF explanations for XAI. These are considered to be five deficits related to psychological and computational evaluations in CF XAI. These deficits include neglecting users, grounding of plausibility with psychology, considering sparsity based on feature differences, evaluating coverage for plausible explanations, and performing comparative testing (Keane et al., 2021). To apply CF explanations to scheduling, further research can explore XAI techniques that prioritize user-centric explanations, generate plausible and interpretable scheduling decisions, address resource allocation, ensure a comprehensive evaluation of explanations for coverage and trustworthiness, and conduct comparative testing to identify the most effective CF XAI methods for scheduling applications.

By integrating XAI into scheduling users can leverage various XAI techniques and tools, such as classification methods for job scheduling problems, customizable rules, textual descriptions, pseudo-code, decision trees, and flowcharts. Additionally, job sequencing and scheduling problems, frequently formulated as mathematical programming models, can be optimized using AI technologies, with a particular focus on the application of Genetic Algorithm (GA) for finding optimal solutions in the scheduling process. This integration enhances the transparency and interpretability of scheduling decisions, allowing users to better understand the reasoning behind decisions, leading to more informed and beneficial scheduling outcomes (Chen, 2023).

2.3 LLMs and XAI for scheduling

Limited research exists regarding the use of LLMs and XAI in the context of scheduling, resulting in minimal scope for meaningful comparison of the proposed techniques in this paper. This section outlines the potential of using both LLMs and XAI to enhance transparency and interpretability in the decision-making process of scheduling. A recent study, building on the understanding that scheduling data is often in a tabular structure (Francis, 2015), was conducted to determine the potential of LLMs in their ability to understand tabulated data. The research explored using GPT-3 providing several challenges to the model (Sui et al., 2024). This produced varying outputs based on the choice of inputs, including table formats, prompts, partition masks, and role prompting. The paper proposes self-augmentation for effective structural prompting, leveraging LLMs' internal knowledge for tasks like critical value/range identification. As illustrated in Fig. 1, the LLM can extract significant values from the table using self-augmented prompting, which aids in generating improved answers for downstream tasks.

Therefore, in the context of scheduling, this demonstrates that LLMs can process volumes of tabular and textual data, including scheduling rules, constraints, and requirements to assist in automating the scheduling process. These models can also interpret and extract relevant information from unstructured text, facilitating better decision-making and proficient scheduling. Another study analyzed tabulated data from a scheduler based on fixed and structured queries. These queries were targeted at specific scheduled tasks and assessed the

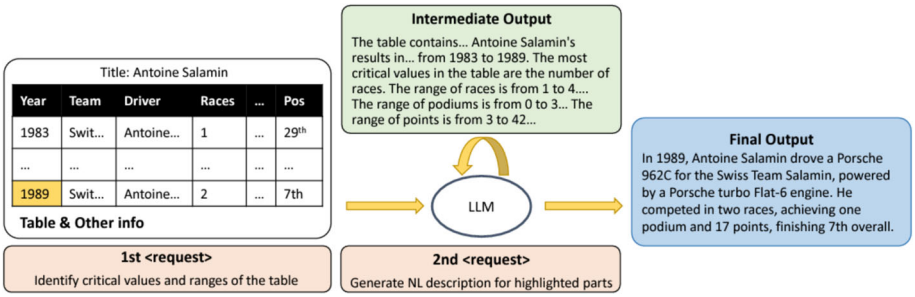


Fig. 1 Illustration of self-augmented prompting (Sui et al., 2024)

feasibility of replacing them with alternative tasks without impacting the schedule, known as a Single Exchange Property (SEP) concept. The generated prompts were fed to the LM along with the initial query to aid with generating an appropriate response (Powell et al., 2023). A summary of the process is shown in Fig. 2.

This outlines the current known capability for the use of LLMs with XAI for solving and explaining queries on scheduling data and problems, signalling the emergent nature of this combination of techniques.

2.4 Scheduling benchmark set

When considering the type of scheduling problem to adapt to the approach in this study, the task-based nature of schedules aligned best with Job Shop Scheduling, which is a specific class of scheduling problems which has been extensively research over many years (Xiong et al., 2022). The approach in this paper is independent of the scheduling problem and can be extended to any other problem type in future research, such as bin packing or employee scheduling.

As such, the history of job shop scheduling was examined, and publicly available benchmark problems were identified from the substantial research by E. Taillard on job shop Scheduling, in which 260 benchmark scheduling problems of varying sizes and optimality were defined. These benchmark schedules have been utilised in research for over 30 years with over 3,000 citations, building a well-established baseline that can be applied to any task-based, job-shop-aligned scheduling problem. Because of this, schedules of different sizes were selected from the original research, where the schedule data was available, to determine how the proposed methodology scales with increased schedule size and complexity, which will form the basis of an XAI experiment to demonstrate the capabilities in a neutral, non-domain-specific setting. Please refer to Taillard (1993) for the algorithm used to create the different problem instances.

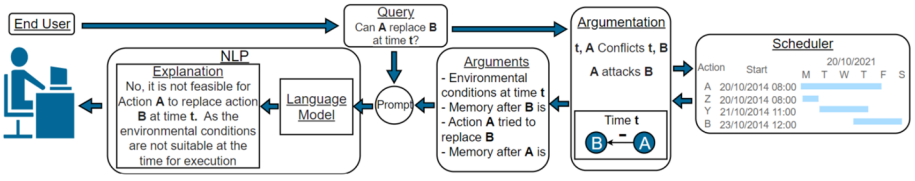


Fig. 2 NLP combined with argumentation to analyse a schedule (Powell et al., 2023)

The framework problems outlined by E. Taillard provided the means of generating solver results for a set of benchmark schedules, where the number of jobs (n) and the number of machines (m) can be altered to control the size of the schedule. Within the schedule, each job has an uninterruptable duration (or processing time), randomly determined between 1 and 99 (Taillard, 1993; Jain & Meeran, 1999), that must be completed for the job to finish and machines can only process one job at a time.

The collective performance of research for solving E. Taillard's benchmark has been summarised from dozens of different research studies to document the lower and upper bounds of solutions (Shylo & Shams, 2018); where the lower bound represents the optimal solution, and the upper bound represents the current best feasible solution, with the goal of optimisation matching the bounds together through exhaustive solving (Brucker & Knust, 2006). From the presented information, at the time of writing, of the 80 Job-Shop Scheduling benchmark problems 21 remain with non-optimal upper bounds (Shylo & Shams, 2018).

The authors of this paper considered the solutions located within (Taillard, 1997) were derived by Brinkkötter and Brucker (2001) for seven different schedules entailing 15 jobs by 15 machines with makespans of 1218, labelled as TA03; 1175 as TA04; 1224 as TA05; 1238 as TA06; 1227 as TA07; 1217 as TA08; and 1274 as TA09. Additionally, two schedules by Henning Dr. rer. nat. (2002) for 20 jobs and 15 machines with a makespan of 1342, labelled as TA13; 20 jobs on 20 machines with a makespan of 1647 as TA26; and lastly, 30 jobs and 20 machines with a makespan of 1956 as TA48 (Shylo, 2002) as shown in Table 1. This paper used these schedules as benchmarks with the proposed methodology to generate questions/answers and explanations.

Figure 3 represents a Gantt chart of the schedule derived by Henning Dr. rer. nat. (2002) of a Job Shop scheduling problem TA13 across a time horizon. Each machine has 20 jobs assigned to it, with no overlapping of jobs across any machines.

3 Methodology

An LLM based method was established to answer queries for three different categories of queries to explore its potential by analysing different types of tabular job-shop scheduled data, adhering to the predefined constraints of the schedule. The types of scheduled data analysed

Table 1 Schedules used with their respective makespans (Brinkkötter & Brucker, 2001; Henning Dr. rer. nat., 2002; Shylo, 2002; Taillard, 1997)

Schedule Label	Schedule Type jobs j by machines m	Makespan
TA03	15j x 15m	1218
TA04	15j x 15m	1175
TA05	15j x 15m	1224
TA06	15j x 15m	1238
TA07	15j x 15m	1227
TA08	15j x 15m	1217
TA09	15j x 15m	1274
TA13	20j x 15m	1342
TA26	20j x 20m	1647
TA48	30j x 20m	1956

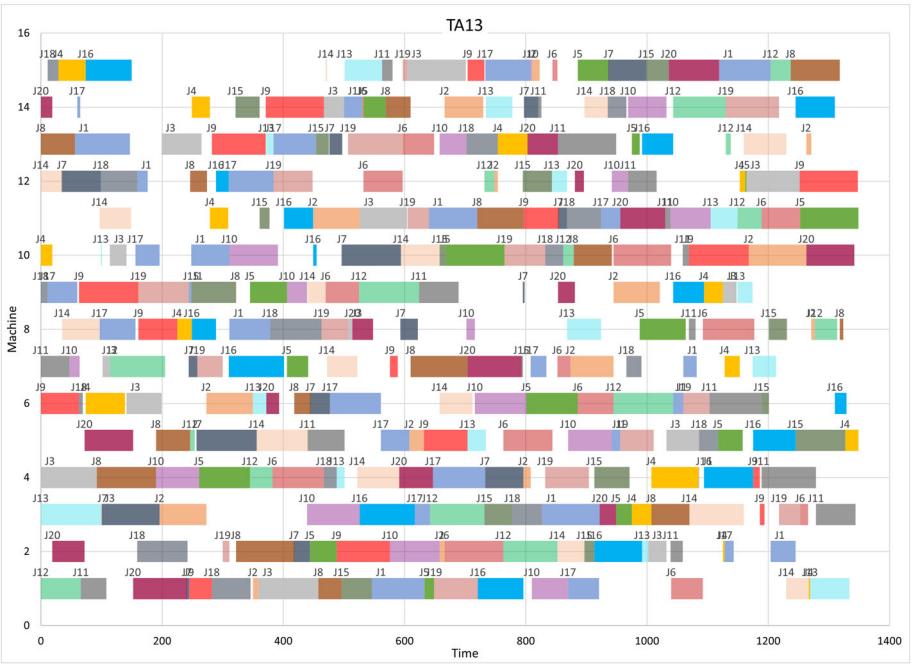


Fig. 3 A representation of schedule TA13 derived by Henning Dr. rer. nat. (2002) proposed by Taillard (1997)

were seven different $15j\text{x}15m$ schedules and one schedule of the following combinations $20j\text{x}15m$, $20j\text{x}20m$, and $30j\text{x}20m$ where j stand for jobs and m represent machines. The approach outlined in this paper could also be applied to any scheduling problem and is not dependent on a job shop scheduling format. The job shop scheduling problem was selected due to the extensive studies conducted on the format, with challenges in solving some of the most complex instances, which fits the differing levels of complexity in problems utilised in this study.

This study introduces the use of OpenAI's GPT-4 and Meta's LLaMA-3.1 to determine the feasibility of task alterations of ten benchmark schedules, where the unique queries are shown in Table 2 and prompts provided are shown in Table 3. GPT-4 was selected due to observed high performance in domain-specific areas, such as medicine (as discussed in Sections 1 and 2), whereas LLaMA-3.1 was selected because of its capability to be used on local machines, broadening the opportunity for this benchmark study to be replicated and built upon (Ersoy & Erşahin, 2024). Figure 4 provides an overview of the approach taken in providing data to the LLMs and how the responses on the scheduled data were assessed. This approach utilises the schedules and queries that were created from each category described in Section 3.1 for the respective schedule and combined to prompt the LLMs as explained in Section 3.2. The generalist LLMs, as opposed to a specialised or pre-trained model, were chosen to explore the capabilities of these emergent tools in domain-specific and technical problems. The LLM's responses were assessed in five different ways as explained in Section 3.3 for each of the query categories where comparisons of the relations were discussed.

Table 2 Table displaying the unique queries for each of the query categories

Query Category	Queries
Swap	1. Could the start time of job *a* be exchanged with the start time of job *b* on machine *c*?
	2. Is it possible for the end time of job *a* to be exchanged with the end time of job *b* on machine *c*?
	3. Is the exchange of job *a* and job *b* on machine *c* feasible?
	4. Can job *a* be exchanged between machines *c* and *d*?
	5. Can the processing times of job *a* on machine *c*, be exchanged with the processing times of job *a* on machine *d*?
	6. I'm considering swapping the start time of job *a* on machine *c* with the start time of job *a* on machine *d*. Is this possible?
	7. Suppose I swapped the end time of job *a* on machine *c* with the start time of job *a* on machine *d*, is this possible?
Increase	1. Can the duration of job *a* on machine *c* be increased by *z* minutes?
	2. Can machine *c* overall running time be increased by *z* minutes without impacting the overall scheduled run time?
	3. Is it possible for the start time of job *a* on machine *c* to be increased by *z* minutes?
	4. If I increased the end time of job *a* by *z* minutes on machine *c*, would that be feasible?
Decrease	1. Is it possible for the duration of job *a* on machine *c* to be reduced by *z* minutes?
	2. Can machine *c* overall running time be reduced by *z* minutes without impacting the overall scheduled run time?
	3. I need to know if the start time of job *a* on machine *c* can be reduced by *z* minutes.
	4. Would reducing the end time of job *a* on machine *c* by *z* minutes, be possible?

Table 3 Table displaying the prompts used to answer each of the query categories

Prompt used across all categories for answering queries
This is a schedule for a job shop problem.
Each row labelled J# represents the job across each machine except the first row, and each column except the first represents a machine number.
The scheduling of jobs and machines is not sequential and can be in any order; however, a machine can only run one job at a time, and the same job cannot run at the same time on different machines.
Jobs are never to be repeated on the same machine, and there are no sequencing or dependency rules for jobs on each machine; for example, job 5 can occur before job 4.
The schedule data provided below is not in order of the schedule and must be restructured to be sequential.
When answering questions on the schedule, please consider all the data available and the potential knock-on impact or conflict with other machines, reviewing all possible or necessary adjustments to fully answer the query. There are also no deadlines for jobs or the schedule.
Every Answer MUST start with a yes or no followed by the explanation.
Three examples of answering questions are below:
1. Could the start time of job 9 be exchanged with the start time of job 15 on machine 2?
answer: Yes, the start time of job 9 can be exchanged with job 15 on machine 2, as there are no overlaps of the same jobs on the other machines.
2. Can machine 11 overall running time be increased by 15 minutes?

Table 3 continued

Prompt used across all categories for answering queries
answer: No, machine 11 overall run time cannot be increased by 15 minutes as there would be an overlap in other jobs.
3. I need to know if the start time of machine 15 can be reduced by 13 minutes.
answer: Yes, the start time of machine 15 can be reduced by 13 minutes, as there is availability within the time requested.

3.1 Query creation

Three query categories were created, each containing variations of a set number of unique questions, each of which is shown in Table 2. These are:

- Swap - Inquiries around exchanging the processing times, start times, and same jobs across machines. This category has seven unique benchmark questions.
- Increase - Entails any queries involving an extension of job start, finish, or processing duration time on any job on a machine or machine run time. This category has four unique benchmark questions.
- Decrease - Similar to increase, entails the reduction of job start, finish, or processing duration processing time of a job or overall machine run time. Also has four unique benchmark questions.

The swap category offered a greater range of possible queries over the other categories, which included exchanging the same jobs across different machines, while the schedule maintained that all jobs were scheduled to run on each machine, as well as the option to swap any two jobs on the same machine. Additionally, the exchange of start times or processing times was asked.

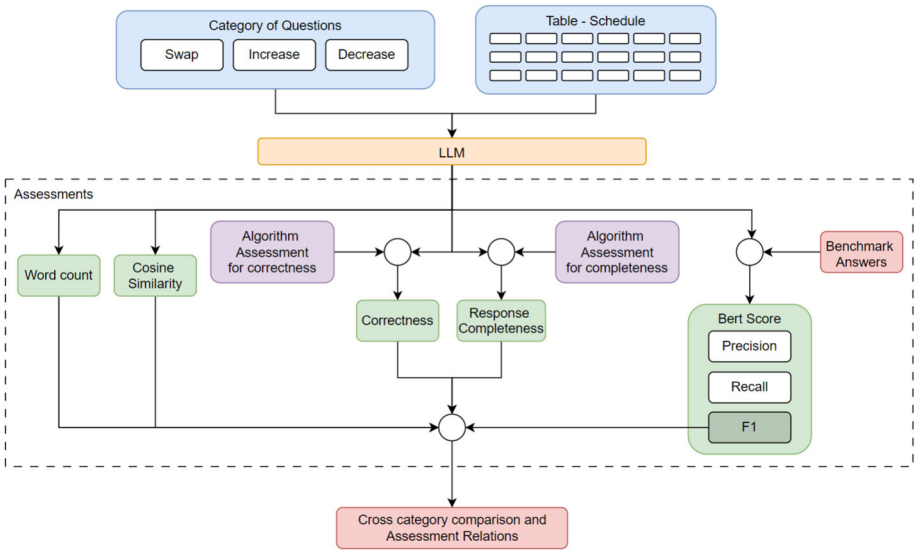


Fig. 4 Overview of LLM analysis on scheduled data

Meanwhile, the Increase and Decrease categories were provided with four unique questions, with each containing a variation of machine and job numbers, supplied to the LLMs for answering.

All unique queries were repeated n times with varying job and machine numbers used during the assessments to measure the consistency of responses by the LLMs. The job and machine numbers, including the times and durations, were randomly generated with constraints ensuring the machine and job numbers were within range of the type of schedule the query was asked against.

3.2 Answering benchmark schedule queries

Two tables for each schedule were created in a text file and used as part of the prompt to the model. The first table contained the processing times for each job on each machine with j rows by m columns. The second table, however, contained the start times for each job on each machine with j rows by m columns.

Prompts, shown in Table 3, were provided with the scheduled data and combined with the queries created from Section 3.1 to assist in the generation of the answers to the queries created. The overall process employs an example-based few-shot approach with the supplied schedule, using the api connector to both GPT-4 and LLaMA-3.1. While the prompt only includes one example question and answer pair for each query category, this approach is considered few-shot learning (instead of one-shot learning) as the LLMs are unaware of the separate query categories and will consider all examples when formulating the answers. The decision was made not to include any additional prompt optimisation techniques, such as chain-of-thought (Shao et al., 2023) or ReACT (Yao et al., 2023), as the experiment aims to establish baseline performance with generalist LLMs for this novel investigation.

The results, including all query variants, were analysed for their performance in correctness, cosine similarity, completeness, response length, and use of language to assess the quality of the LLM response to answering the queries. The metrics used are outlined in the following Sub-Sections.

3.3 Performance measure

The evaluation of the performance of the proposed methodology is based on the analysis of the accuracy, consistency and readability of the responses to the varying query categories and unique queries answered by the LLMs. Additionally, the potential similarities and patterns between each assessment metric and query category, in how the LLMs were able to generate a response to queries from the tabular schedules of different sizes, was analysed.

3.3.1 Correctness

In determining the correctness of the answers generated by the LLMs, the *yes*/*no* responses were assessed, producing a binary value to signify whether the answer was deemed correct or not. This was done algorithmically to independently check the feasibility of the schedule alterations based on the questions, then compared with the response from the LLM. If the alterations queried in the original query were feasible and correlated to the *yes* response, a score of *1* was given; likewise, if the alterations queried were not feasible and the response was *no*, a score of *1* was also given to show the response as correct; otherwise, if any other

result, a score of *0* was recorded. However, where answers contain two components for correctness, for example, when exchanging processing times between two jobs (within the Swap category), either by altering the end time or start time, provided the algorithm produces a result for the two outcomes, where at least one coincides with the output from the LLMs, a score of *1* was given to represent the response as correct. As all queries asked should return a *yes*/*no* response no other conditions were required to assess correctness in the LLMs responses.

3.3.2 Cosine similarity

In the context of LLMs, cosine similarity is the measure of similarity between two textual statements. This was calculated by computing, following the method presented by Face (2024), the similarities between the answers created by the LLMs across each repeated unique question, excluding itself. This means n answers were generated for each query within each category, and each of these answers (from Table 2) was computed for the cosine similarity against the others generated for that question. The returned values were averaged across all responses to the query and recorded.

The cosine similarity score ranges from *-1* to *1*, where *-1* represents no similarity whatsoever, and *1* would be an identical response. Assessing the cosine similarity allows for a measure of consistency in language, tone, and response structure, which enables familiarity with users in real-world applications.

3.3.3 Response completeness

The response completeness was calculated to assess the LLMs capabilities in identifying and referencing the key components of the question, which include the job number(s) and machine(s) specified. Additionally, the similarity of the response to the query asked was also part of the calculation to evaluate how much common language and terminology was used in the response.

In calculating the completeness of the response from the LLMs, two steps were followed:

1. Check if the job and machine numbers from the query were mentioned within the response. These values range from *0* to *1*, with *0* being no mention of the jobs or machines within the explanatory responses and *1* representing 100% of the noted jobs and machines mentioned.
2. Calculate the cosine similarity between the query and answers generated to assess the use of common words and terminology.

The resultant values are averaged to generate the total response completeness, which will attain a value between *0* and *1*; where a score of *0* means the response excludes all relevant information provided in the question, and a score of *1* perfectly evidences the relevant information from the query and the greater comprehension the LLM exhibits.

3.3.4 Word count

The word count of each response was also measured to analyse the difference in length of the responses to assess if there is a correlation with other assessments and query categories. The response tokens were set to a constant limit as detailed in Section 4 to minimise the fluctuation in response length and better represent real-world implementations.

3.3.5 BERTScore

A sample benchmark of 30 queries and answers (including both *yes* and *no* responses), shown in Table 9 in Appendix A.1, was created for each category, by the authors of this paper. Each sample was mapped to the list of answers generated by the LLMs to return the BERTScore (F1), which assessed the quality of language used in each response in relation to the samples provided. The BERTScore (F1) is the average of two-component values:

1. Precision - measures the accuracy of words within the response; and
2. Recall - measures the quality of phrases used within the response.

All three scores were calculated and presented; however, the results focused on analysing the BERTScore (F1) values. The calculation method follows the instructions provided in Face (2024) and is scored between *0* and *1*, where *0* has no resemblance to reference material and *1* is identical to a statement in the reference material.

3.3.6 Comparative performance analysis

Once all the assessments were calculated for all queries across all schedules and each query category, the results were analysed to compare the performance observed between each category. Graphs were plotted for each assessment metric to visualise the results and aid in assessing performance differences. The comparative performance was discussed, detailing relevant insights and reasoning gained from the experiment.

4 Results and discussion

The results section is presented in two parts. The first part summarises the results for each query category individually, and the second part discusses the results across all categories and scheduling problem sizes. There were a total of fifteen unique queries across all the categories: seven queries for Swap and four queries for both Increase and Decrease. Each query contained variations of job numbers, machine numbers, and different time intervals suitable for the respective schedules, following which the subsequent responses were assessed, and the results were averaged for each query and discussed in each category.

Section 4.1 contains the assessed results for the Swap, Increase, Decrease query categories, where all ten schedule formats were analysed, namely seven schedules of $15j$x$15m$ (represented as schedules 1 - 7), one schedule $20j$x$15m$, one schedule $20j$x$20m$, and finally, schedule $30j$x$20m$. The results are presented in Tables 10 through 21 in the Appendices (A.2, A.3 and A.4), the data from which was also used to plot all figures shown in Section 4.2.

Section 4.2 contains a comparison overview between the categories with their respective Figures, where each schedule is represented as 15_15_1 to 15_15_7 for all schedules of $15j$x$15m$, 20_15_1 for schedule $20j$x$15m$, 20_20_1 for schedule $20j$x$20m$, and 30_20_1 for schedule $30j$x$20m$. Additionally, the overall performance is compared with results observed in other studies to determine the success of the experiments.

The analysis conducted within each section provided valuable insight into the application and performance of the LLMs approach for distinct scheduling problems. It is important to note the values in bold text shown in Tables 5 through 21 excluding Table 9, represent the highest average scores for each performance measure for the respective schedules.

Using GPT-4 and LLaMA-3.1 required hyperparameters to allow the exploration of the variations of answers. Upon finding the most suitable settings shown in Table 4, it remained consistent throughout testing to enable a fair assessment across each response.

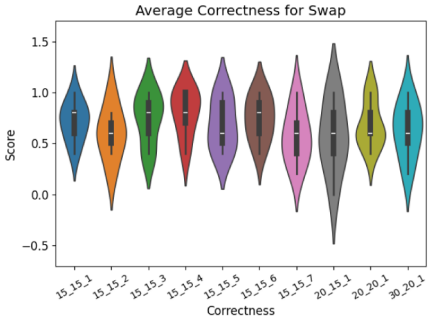
Table 4 Model configuration to generate answers

Model configuration	
temperature	1
max tokens	100
top_p	1
frequency penalty	0
presence penalty	0

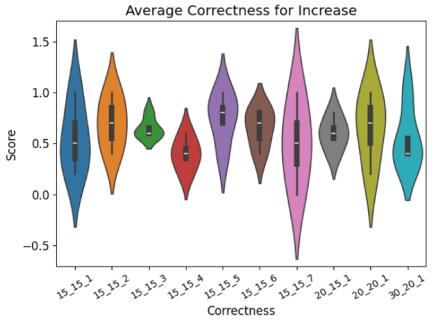
4.1 Individual query categories

4.1.1 Swap query category

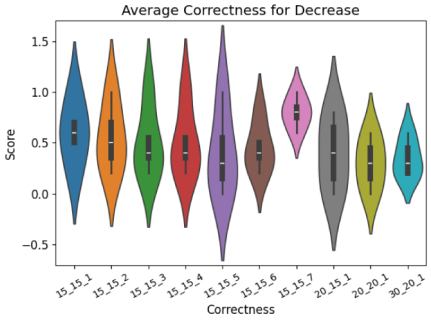
For the Swap query category, Tables 10 and 12 represent the average correctness scores ranging between 0.57 and 0.80 across all schedule sizes for GPT-4, with over 74% of queries achieving an average correctness score of 0.60 or higher. However, Tables 11, and 13 show the LLaMA responses achieved averages between 0.00 and 0.29, as 40% of queries returned an average of 0.00. The GPT-4 performance is presented in Fig. 5a, where the scores were



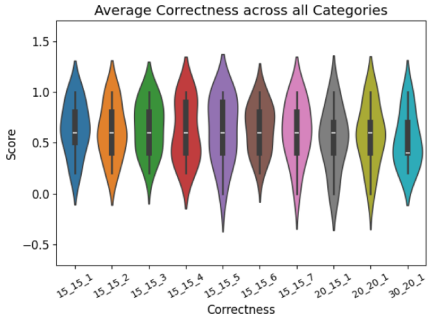
(a) Average Correctness across the Swap query category for GPT-4.



(b) Average Correctness across the Increase query category for GPT-4.



(c) Average Correctness across the Decrease query category for GPT-4.



(d) Average Correctness across all query categories for each schedule for GPT-4.

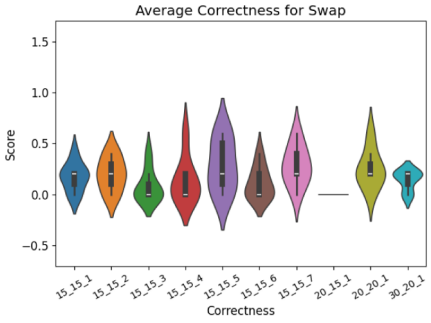
Fig. 5 Average Correctness across the three query categories for all schedules for GPT-4

relatively high across all schedule variants. While LLaMA's results shown in Fig. 6a visualise the low scores for all schedules.

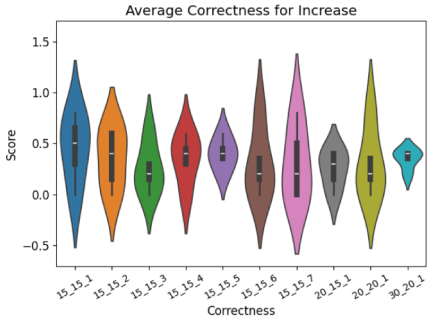
The average cosine similarity scores for GPT-4 ranged between 0.78 and 0.82, demonstrating consistent similarities between answers without being identical, which was closely matched by LLaMA with scores between 0.77 and 0.83. A density plot was created showing where GPT-4 and LLaMA had only slight variations for cosine similarity, shown in Figs. 7a and 8a, where all $15j$x$15m$ schedules represented with solid lines were compared with $20j$x$15m$, $20j$x$20m$, and $30j$x$20m$ as broken lines.

With the completeness assessment, the average scores measured between 0.93 and 0.955 for GPT-4, and between 0.94 and 0.96 for LLaMA, representing a high degree of recall from the elements provided within the query by both LLMs. Looking at Fig. 9a for GPT-4, each schedule was plotted against their average scores taken from each question, with schedule 5 of $15j$x$15m$, showing the lowest reading 0.8153 taken from query 7 shown in Table 10. Schedule $20j$x$15m$, however, shows the second lowest reading of 0.8531, also taken from query 7 in Table 12. The plot for LLaMA, in Fig. 10a, reveals less variance, with all schedules closely aligned in average score distribution.

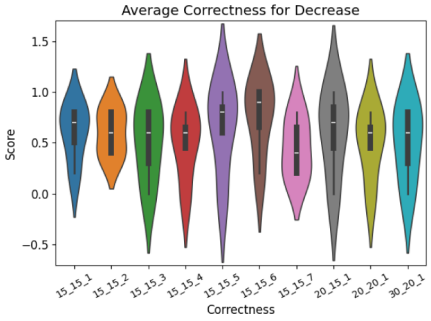
The average word count was calculated as between 59 and 73 words per response from GPT-4, and between 54 and 66 from LLaMA, which suggests a high degree of consistency in responses from both LLMs. However, when looking closely at the individual query responses



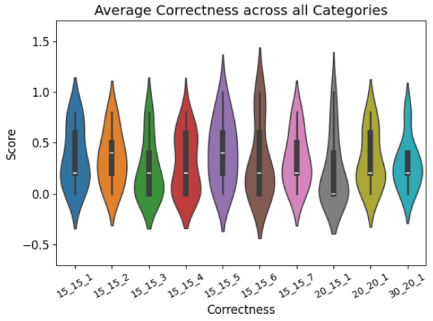
(a) Average Correctness across the Swap query category for LLaMA.



(b) Average Correctness across the Increase query category for LLaMA.

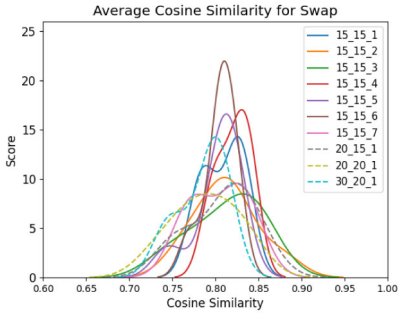


(c) Average Correctness across the Decrease query category for LLaMA.

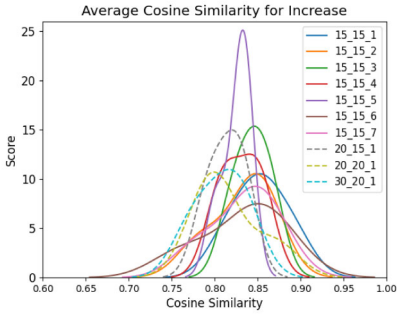


(d) Average Correctness across all query categories for each schedule for LLaMA.

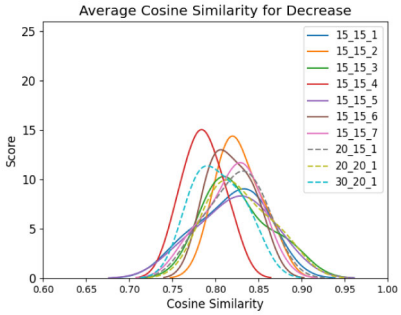
Fig. 6 Average Correctness across the three query categories for all schedules for LLaMA-3.1



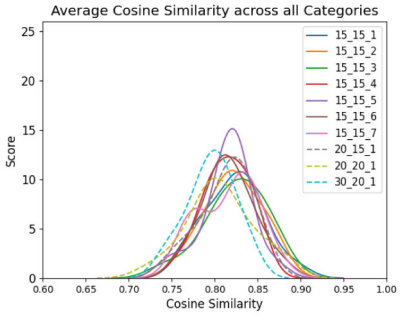
(a) Average Cosine Similarity across the Swap query category for GPT-4.



(b) Average Cosine Similarity across the Increase query category for GPT-4.



(c) Average Cosine Similarity across the Decrease query category for GPT-4.



(d) Average Cosine Similarity across all query categories for GPT-4.

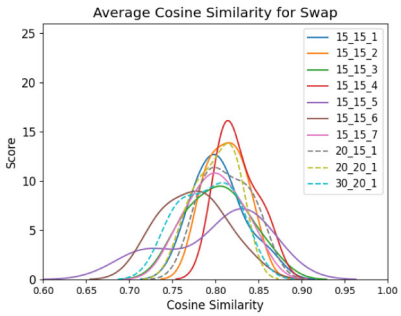
Fig. 7 Average Cosine Similarity across the three query categories for GPT-4

from both LLMs, there were noticeable variances in the length as shown in Figs. 11a and 12a, which infers that particular wording of a query can greatly influence the length of the response.

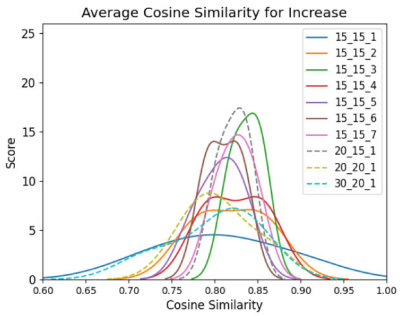
For BERTscore (F1), the average scores for GPT-4 ranged between 0.61 and 0.66, with LLaMA achieving between 0.59 and 0.64, which shows the quality of the responses provided by both LLMs had a high degree of consistency with the human sample responses, with minimal fluctuation shown in the scores, while also not too closely aligning with the reference material. The consistency of these scores was very similar across both LLMs and is shown in Figs. 13a and 14a, representing a density violin plot of these values.

4.1.2 Increase query category

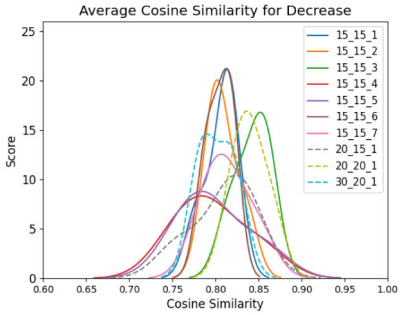
The results for GPT-4 from the Increase query category are shown in Tables 14, and 16, where the average correctness scores were between 0.4 and 0.75, and over 62% of queries scoring an average of 0.6 or higher. The responses from LLaMA shown in Tables 15, and 17, achieved an average correctness between 0.25 and 0.45, with 12% of answers scoring 0 and 13% scoring 0.6 or higher across all variations. The fluctuations in these scores were reflected in Figs. 5b and 6b for GPT-4 and LLaMA respectively. Schedule 4 of $15j$x$15m$ stands out for having the lowest correctness score from GPT-4, along with schedule 7 from $15j$x$15m$,



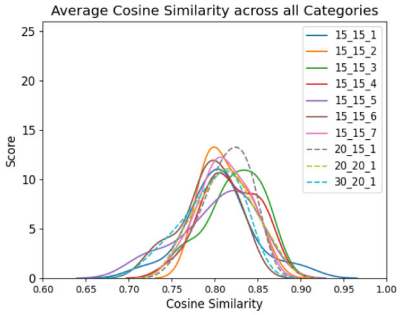
(a) Average Cosine Similarity across the Swap query category for LLaMA.



(b) Average Cosine Similarity across the Increase query category for LLaMA.



(c) Average Cosine Similarity across the Decrease query category for LLaMA.



(d) Average Cosine Similarity across all query categories for LLaMA.

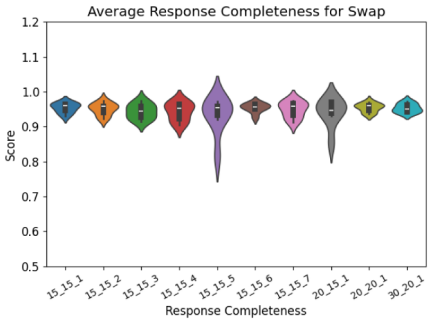
Fig. 8 Average Cosine Similarity across the three query categories for LLaMA-3.1

which has a very broad distribution due to both fully incorrect and fully correct answered questions, whereas the scores from LLaMA were consistently distributed.

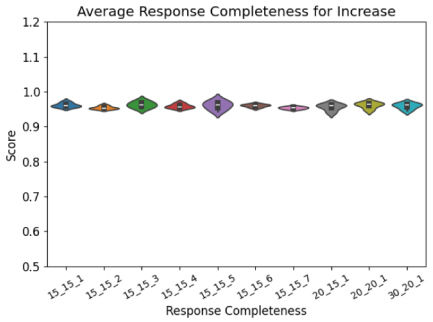
The average cosine similarity scores for GPT-4 ranged between 0.80 and 0.855, presenting a high degree of consistency in the LLM responses, with LLaMA achieving very similar results, scoring between 0.79 and 0.84. The density plots are shown in Figs. 7b and 8b (GPT-4 and LLaMA), identified schedule 5 of $15jx15m$ from GPT-4 as the highest density with a value around 0.83 when compared with the other schedules, while schedules $20jx15m$, $20jx20m$, and $30jx20m$ represented a lower overall density range. The distribution of cosine scores from LLaMA was similar for all schedules, with the exception of schedules 1 of $15jx15m$ and $30jx20m$ with scores ranging from 0.72 to 0.90 and 0.72 to 0.84 respectively, exceeding the average range.

The average completeness was scored between 0.95 and 0.965 for both GPT-4 and LLaMA, demonstrating the LLMs were both able to identify relevant information from the queries in almost every case. The very narrow range in scores can be seen in Figs. 9b and 10b, emphasizing how consistently the LLMs referenced the correct job and machine numbers.

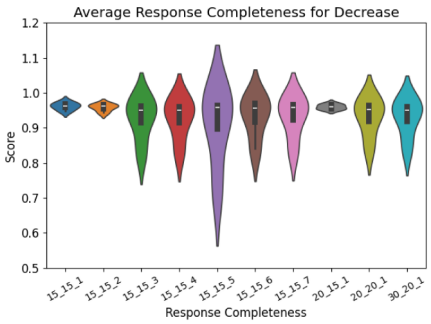
Considering the average word count, which ranged between 51 and 66 for GPT-4 and between 53 and 71 for LLaMA, it can be observed the length of responses was fairly concise, with neither of the LLMs used the full token limit on average. It can be noted, from GPT-4, that queries 1 and 3 from $15jx15m$ schedules 5, 6, and 3, respectively, had significantly lower



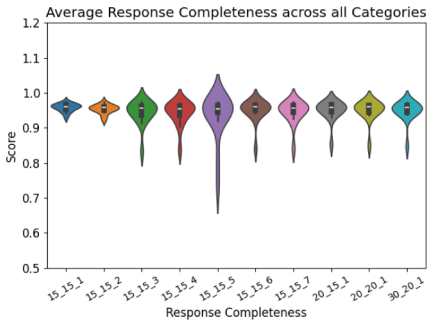
(a) Average Response Completeness across the Swap query category for GPT-4.



(b) Average Response Completeness across the Increase query category for GPT-4.



(c) Average Response Completeness across the Decrease query category for GPT-4.



(d) Average Response Completeness across all query categories for GPT-4.

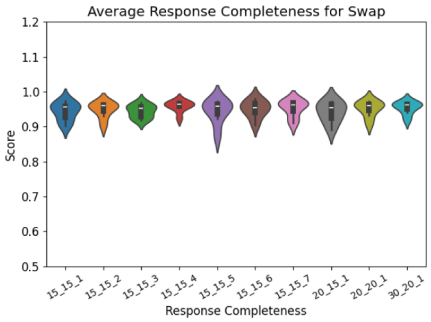
Fig. 9 Average Response Completeness across three query categories for GPT-4

than average word counts, visible in Fig. 11b (also shown in Table 14). However, the nature of the queries within the category means, at times, the LLM can answer sufficiently well with very few words. This pattern was not shared by the responses from LLaMA, which had a more even distribution of length in response as shown in Fig. 12b.

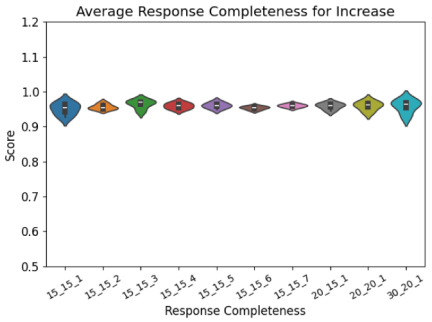
For the average BERTscore (F1) this category, for GPT-4, achieved scores between 0.64 and 0.70, which represents that the LLM consistently used language aligned to human reference material, as shown by the minimal fluctuation in Fig. 13b. LLaMA scored between 0.63 and 0.665, demonstrating very similar performance, as shown in Fig. 14b.

4.1.3 Decrease query category

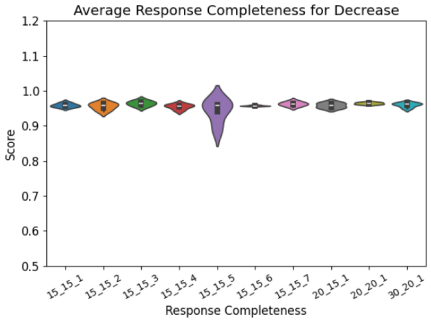
In the Decrease query category results, Tables 18, and 20 for GPT-4 showed the correctness score ranged between 0.3 and 0.8, outlining the variance in correctness scores and where 42% of answers were above a score of 0.6. While Tables 19, and 21 represent the responses from LLaMA which scored between 0.45 and 0.75, with 40% of answers scoring above 0.6. For GPT-4 it was observed the larger schedules performed below the average score ranges of the $15j$x$15m$ schedules, as shown in Fig. 5c, with overall average correctness scores ranging from 0.30 to 0.40. This may be the result of the comprehension necessary to successfully answer these query types in addition to assessing larger and more complex



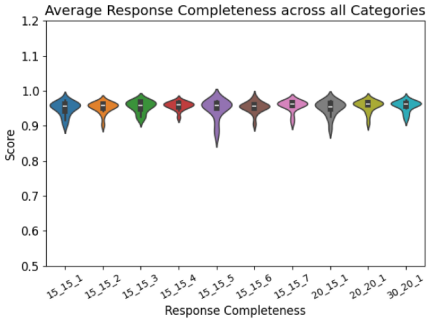
(a) Average Response Completeness across the Swap query category for LLaMA.



(b) Average Response Completeness across the Increase query category for LLaMA.



(c) Average Response Completeness across the Decrease query category for LLaMA.



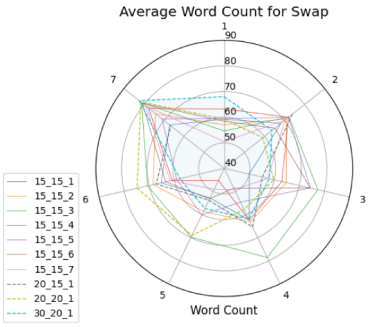
(d) Average Response Completeness across all query categories for LLaMA.

Fig. 10 Average response completeness across the three category queries for LLaMA-3.1

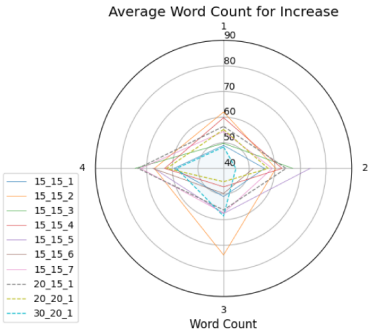
datasets; the results fluctuated where further experiments could be conducted to validate the pattern. This was not the case with LLaMA, where the larger schedules achieved the same levels of performance as the smaller variants, as shown in Fig. 6c, which demonstrated the difference in comprehension that different LLMs can have.

For the cosine similarity assessment, the scores ranged between 0.78 and 0.83 for GPT-4 and between 0.79 and 0.84 for LLaMA, maintained a consistent measure of similarity across all queries and schedule sizes from both LLMs. Given the average scores for each query of schedule 4 of $15j$x$15m$ from GPT-4 contain the lowest values in this category, the scores from all other schedules, however, were very closely aligned as presented in the density graph in Fig. 7c. The distribution of similarity scores from LLaMA was more varied, with schedules either aligning to the bottom or the top of the score range, seen in Fig. 8c, albeit with relatively small differences.

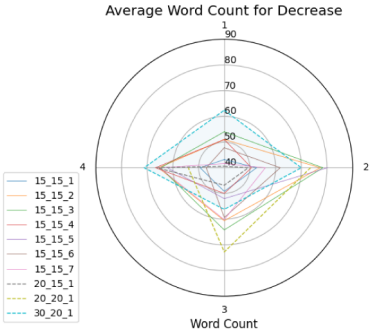
With the average completeness scores that ranged between 0.90 and 0.965 for GPT-4 and between 0.94 and 0.965 for LLaMA, a greater variance was observed from GPT-4 in this category. However, it can be seen in Fig. 9c how closely aligned completeness scores are for schedules 1 and 2 of $15j$x$15m$, and schedule $20j$x$15m$, while the others had much larger score ranges. Conversely, outside of schedule 5 of $15j$x$15m$, the completeness scores for LLaMA were evenly and closely distributed, as shown in Fig. 10c, highlighting the ability for the LLM to return relevant information, even if the assessment may be incorrect.



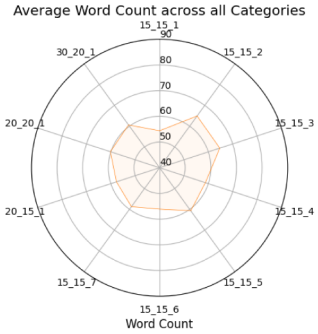
(a) Average Word Count across the Swap query category for GPT-4.



(b) Average Word Count across the Increase query category for GPT-4.



(c) Average Word Count across the Decrease query category for GPT-4.

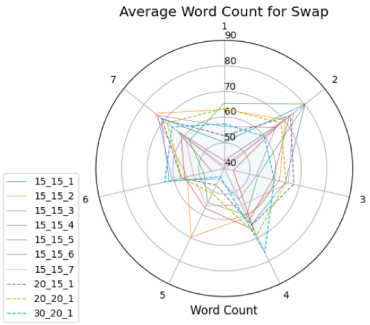


(d) Average Word Count across all query categories for GPT-4.

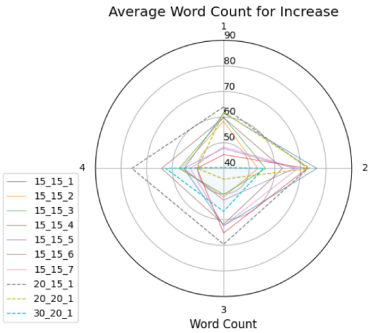
Fig. 11 Average Word Count across the three query categories for GPT-4

For the word count, GPT-4 returned responses that ranged between 48 and 66 words on average, which was the largest range of all the query categories. LLaMA returned responses between 62 and 70, towards matching the ranges from other query categories, emphasizing the consistency in responses from this LLM. Considering Fig. 11c for GPT-4, there was a clear pattern of query 1 having significantly fewer words (11 words less per response) on average when compared with the overall schedule averages. This appears to be due to the straightforward nature of the question: *Is it possible for the duration of job X on machine Y to be reduced by Z minutes?*, which the LLM is able to answer very concisely. Whereas with LLaMA in Fig. 12c showed no distinguishing pattern or irregular response.

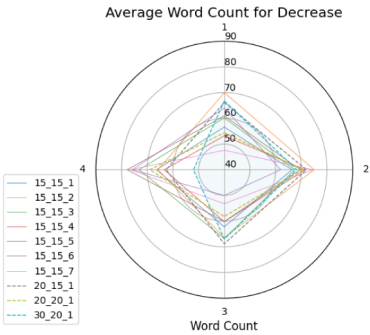
Finally, for the BERTscore (F1) assessment, the scores from GPT-4 in this category ranged between 0.67 and 0.72, which was observed to be the highest range of scores for all the query categories. For LLaMA, the F1 scores ranged from 0.61 to 0.65, which aligned with the previous scores from the other query categories. The consistency from both LLMs can be seen in Figs. 13c and 14c where there was a close similarity of distribution across all schedules, with the exception of schedules 4 and 6 of $15j$x$15m$ for LLaMA which were densely distributed in the scores. The uniformity seen in the results also showed that increasing the data size and complexity does not have an adverse impact on the LLMs interpretation and style of responses.



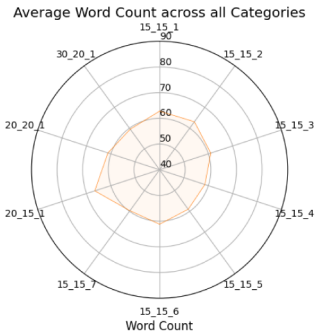
(a) Average Word Count across the Swap query category for LLaMA.



(b) Average Word Count across the Increase query category for LLaMA.



(c) Average Word Count across the Decrease query category for LLaMA.



(d) Average Word Count across all query categories for LLaMA.

Fig. 12 Average Word Count across the three query categories for LLaMA-3.1

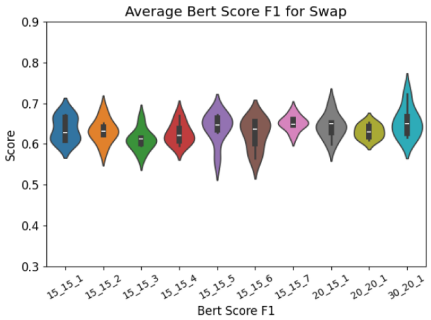
4.2 Cross category comparison and performance discussion

To gain further insight from the results, the assessment metrics from each query category across both LLM responses were also compared against each other to identify any correlations or significant differences in performance. The results were also collated for each schedule variant, in Tables 5 and 6, to assess whether scheduling size and complexity have any influence on the assessment scores.

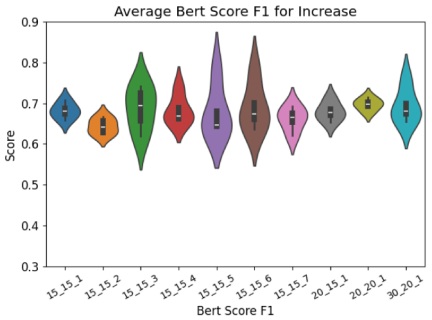
4.2.1 Average correctness

The overall average correctness for the Swap category from GPT-4 was 0.67, while the Increase and Decrease categories scored 0.60 and 0.49, respectively, presented in Table 7. For LLaMA the overall average of correctness was 0.17, 0.33, and 0.57 for the Swap, Increase, and Decrease categories respectively, shown in Table 8. It is also worth noting that in Fig. 5a, b, and c, from the GPT-4 results, the deviation range in correctness averages increased from the Swap category to the Increase category and then again to the Decrease category, with the same pattern observed in the LLaMA results, seen in Figs. 6a, b, and c.

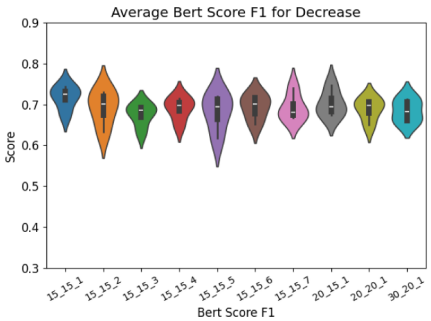
GPT-4 performed well and consistently with correctness scores for Swap, underpinning this LLMs ability to interpret the queries in this category. The Increase category was less



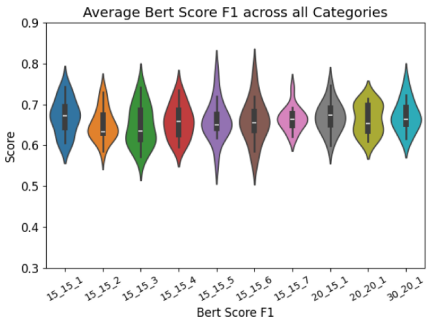
(a) Average BertScore F1 across the Swap query category for GPT-4.



(b) Average BertScore F1 across the Increase query category for GPT-4.



(c) Average BertScore F1 across the Decrease query category for GPT-4.



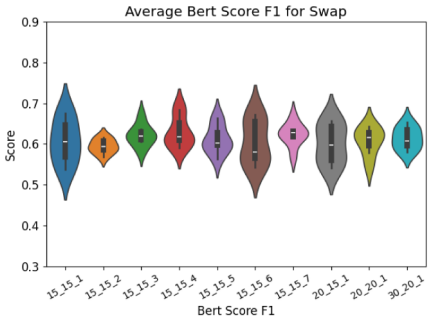
(d) Average BertScore F1 across all query categories for GPT-4.

Fig. 13 Average BertScore F1 across the three query categories for GPT-4

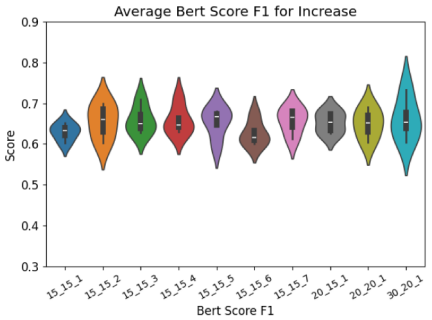
consistent and, as a result, returned a drop in the overall average correctness, although some of the schedules matched the performance seen within the Swap category. With the Decrease category, there was consistently lower performance across all schedules, with the clear exception of schedule 7 of $15j$x$15m$, which alone matched the level of performance of Swap.

The reduction in average correctness for Increase and Decrease query categories, from GPT-4, was most likely due to two things: initially, by requiring calculations to modify the time by z minutes, and secondly, the openness of query 2 leaving room for different interpretations for a generalist LLM. This brings the requirement on the LLM to understand the queries and utilise deeper comprehension in analysing the schedule data to determine the feasibility of the change. Additionally, with the Swap category, the queries were more direct and closed and may be resolved easily without calculations required of the schedule data, and therefore, a deep comprehension may not be required.

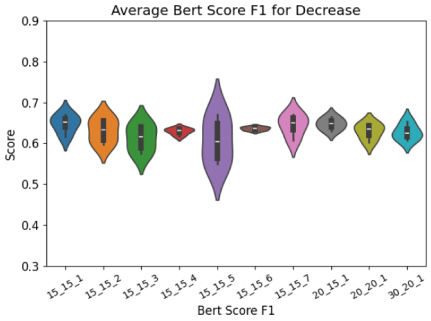
Interestingly, the average correctness scores from LLaMA presented the reverse pattern, with the Swap category returning the lowest scores, with improvements seen in the Increase and improved further in the Decrease category; the responses from LLaMA in the Decrease category outperformed GPT-4, the only area where this model performed better. This underpins the importance of assessing different LLMs, even without pre-training certain models can perform better under certain conditions. LLaMAs capability with the Decrease query



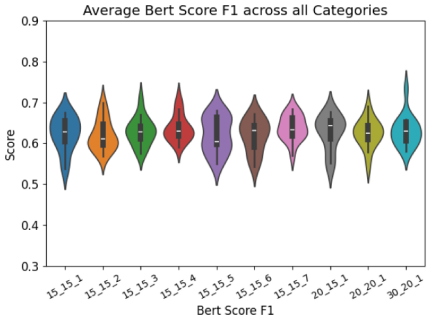
(a) Average BertScore F1 across the Swap query category for LLaMA.



(b) Average BertScore F1 across the Increase query category for LLaMA.



(c) Average BertScore F1 across the Decrease query category for LLaMA.



(d) Average BertScore F1 across all query categories for LLaMA.

Fig. 14 Average BertScore F1 across the three query categories for LLaMA-3.1

Table 5 Average results across all categories for each schedule (GPT-4)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m_1	0.62	0.8266	**0.9593**	54.4	**0.6788**
15jx15m_2	0.62	0.8241	0.9545	**64.9**	0.6561
15jx15m_3	**0.63**	**0.8274**	0.9436	64.7	0.6581
15jx15m_4	0.57	0.8124	0.9439	58.8	0.6667
15jx15m_5	0.61	0.8168	0.9321	60.7	0.6663
15jx15m_6	0.61	0.8182	0.9485	56.1	0.6692
15jx15m_7	0.62	0.8187	0.9443	58.7	0.6686
20jx15m_1	0.52	0.8148	0.9522	57.7	0.6753
20jx20m_1	0.55	0.8113	0.9487	60.2	0.6723
30jx20m_1	0.51	0.7981	0.9462	60.6	0.6777

Table 6 Average results across all categories for each schedule (LLaMA-3.1)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m_1	0.41	0.8058	0.9511	62.8	0.6283
15jx15m_2	0.38	0.8136	0.9536	63.1	0.6269
15jx15m_3	0.28	**0.8281**	0.9575	61.0	0.6313
15jx15m_4	0.33	0.8155	0.9576	58.6	0.6395
15jx15m_5	**0.45**	0.8021	0.9491	59.0	0.6239
15jx15m_6	0.39	0.7969	0.9540	61.2	0.6223
15jx15m_7	0.35	0.8131	**0.9584**	59.7	**0.6417**
20jx15m_1	0.28	0.8149	0.9531	**66.6**	0.6337
20jx20m_1	0.35	0.8160	**0.9584**	61.2	0.6289
30jx20m_1	0.33	0.7971	0.9576	59.5	0.6345

Table 7 Average results for all categories across all schedules (GPT-4)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
Swap	**0.67**	0.8049	0.9479	**63.9**	0.6350
Increase	0.60	**0.8287**	**0.9589**	57.6	0.6798
Decrease	0.49	0.8170	0.9353	57.6	**0.6919**
Overall Total	0.59	0.8169	0.9473	59.7	0.6689

Table 8 Average results for all categories across all schedules (LLaMA-3.1)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
Swap	0.17	0.8027	0.9489	59.8	0.6116
Increase	0.33	**0.8159**	**0.9583**	59.2	**0.6504**
Decrease	**0.57**	0.8124	0.9580	**64.8**	0.6313
Overall Total	0.35	0.8103	0.9551	61.3	0.6311

category should be explored further in future studies to better understand why these queries are better comprehended than others.

Furthermore, in line with the reduction in overall correctness averages from GPT-4, and increases seen with LLaMA, the increased complexity of the queries also introduced more variability in the correctness scores, which signifies that both LLMs are more inconsistent in comprehending the necessary data and details. This raises an important question: does the LLM understand the rules required for a job shop schedule? Pre-training or chain of thought prompting techniques may be able to boost the performance of generalist LLMs in these domain-specific experiments and possibly narrow the gap in overall correctness.

It is worth noting, in Figs. 5d and 6d when considering the collected average scores by each schedule, the density distribution is near-identical across all schedules, underlining that the schedule size and complexity has little to no influence on a generalist LLM's capabilities in accurately assessing scheduling data and queries.

4.2.2 Average cosine similarity

Reviewing the overall average scores of cosine similarity Figs. 7a, b, c, 8a, b, and c show that the density of scores for each query category was closely aligned for both LLMs, with only one individual stand-out schedule. Schedule 5 of $15j$x$15m$ for GPT-4, within the Increase query category, had a particularly narrow density, resulting in the clear separation from the other schedules, although this outcome was coincidental, was not matched by LLaMA, and does not offer any insight into the operations of the LLM or the performance of other schedules or query categories.

The observed close alignment across the schedules and query categories, visualised in Figs. 7d and 8d, demonstrates that both LLMs used very similar language in responses, regardless of the type of query or the size of the dataset.

4.2.3 Average response completeness

Considering the average scores for completeness, as shown in Fig. 9a, b, and c for GPT-4, the scores achieved in each of the query categories were closely aligned on average, with total averages of 0.9479, 0.9589, and 0.9353 for Swap, Increase, and Decrease respectively, shown in Table 7. For LLaMA the total averages were 0.9489, 0.9583, and 0.9580 for Swap, Increase, and Decrease respectively, shown in Table 8, with the distribution plotted in Fig. 10a, b, and c.

The responses in the Swap and Increase query categories were the most consistent for GPT-4, with minimal variance and exceptional results, while the distribution of the LLaMA scores was very even across all schedules and query categories, collectively shown in Fig. 10d. The responses returned in the Decrease query category, for GPT-4, had the most variance and outlying responses, which impacted the overall average, being the lowest of all the average scores, however, the completeness scores overall were consistent throughout the experiment for all categories and schedule sizes, as shown in Fig. 9d for GPT-4, with a near-uniform density (excluding the observed outlier results). This proves that both LLMs were able to interpret and return valid, relevant information, regardless of query type or data complexity, even when referencing domain-specific concepts.

4.2.4 Average word count

Looking into the average word count scores for GPT-4, depicted in Fig. 11a, b and c, the average word count for responses in the Swap query category was 64, with Increase and Decrease both returning 58 words on average; rounded up from results in Table 7. The responses from LLaMA averaged 60, 59, and 65 for the Swap, Increase, and Decrease query categories, respectively (after rounding), as shown in Table 8, with the distributions plotted in Fig. 12a, b, and c.

In addition to writing responses of similar size from both LLMs, there is also a shared pattern of occasional outlying short responses, as seen through the schedules and query categories. While assessing the word count by the schedule size presents a marginally larger variance in length, as shown in Tables 5 and 6 and Figs. 11d and 12d, these results provide predictability to the operation of both LLMs as users can expect to receive responses of similar length, regardless of the style of query asked or the size of the data within the schedule.

4.2.5 Average BertScore F1

Assessing the average BertScore density plots in Figs. 13a, b, c, 14a, b, and c, there is a high level of consistency with the average BertScores across all schedules from both LLMs. In the Swap category, the GPT-4 responses achieved an average BertScore of 0.6350, with LLaMA achieving 0.6116, and the categories of Increase and Decrease achieved average scores of 0.6798 and 0.6919 for GPT-4, and 0.6504 and 0.6313 from LLaMA respectively, detailed in Tables 7 and 8.

With the best overall average BertScore being achieved by GPT-4 in the Decrease query category, it is evident that these responses best aligned with the human sample references, although there is no significant difference in the performance across the query categories, with LLaMA only scoring 5% lower overall. This is also the observation in Figs. 13d and 14d, where there is a highly similar spread of scores across all schedule variants.

Given the generalist LLMs used in this experiment, these results are encouraging, as the queries and data were specialist and domain-specific. A larger sample size of reference answers would likely improve the observed BertScores, and this should be considered for any future experiments of this type.

4.2.6 Performance discussion

As mentioned in Section 1, there is limited published research in this area of study at the time of writing, which means there is no available data for direct comparison of performance results. The two identified isolated examples with sufficient similarities offer some insight into how the experimental results of this paper compare to existing research.

For correctness measures (referred to as Accuracy in the referenced study), in the study assessing MaScQA, the GPT-4 model achieved accuracy scores of 60.15 for the zero-shot approach and 62.0 for the chain-of-thought method, which very closely aligns with the GPT-4 results generated through the experiment in this paper (overall average of 59) (Zaki et al., 2024). The narrow margin of difference is encouraging as the results generated for this paper did not follow any extensive prompt optimisation techniques and, therefore, opens the opportunity for further investigation. It should be expected that the LLaMA results would improve with prompt optimisation or pre-training as well, as the overall results were considerably lower than GPT-4.

Another study focused on assessing the BertScore achieved by GPT models in answering domain-specific queries on Microsoft products and technical IT problems. The results of the study returned an overall BertScore of 56.91 from the GPT-4 model, which is significantly lower than both sets of results achieved in this paper (overall average of 0.6689 for GPT-4, and 0.6311 for LLaMA) (Yang et al., 2023). The performance demonstrated in this paper shows what can be achieved without dedicated pre-training and additive information to an LLM, underpinning the inherent capabilities of generalist LLMs and the approach introduced in this paper. Furthermore, the authors acknowledge that unintended bias may be introduced to a small, self-written set of reference material and that future studies should consider using publicly maintained reference material or one generated from a broader range of authors.

No suitable comparative research could be found for the completeness and cosine similarity scores, and the limited value that could be derived from comparing the word count of responses was recognised.

The results are encouraging for setting a solid performance basis from which more in-depth or targetted research can build on. The model devised in this study can help form the framework for enabling human-machine interactions or feedback to automated systems through an LLM (or integrated LM), which can add the introduction of AI solutions in task-based scheduling industries, such as manufacturing, logistics, construction, and shift workers. It is important to note that the involvement of human operators or workers is critical to the adoption and success of automated systems (as highlighted in Uhde et al. (2020)), which will facilitate the correction or mitigation of ethical concerns in task assignment and scheduling, avoiding such issues as worker overload or gender-bias.

5 Conclusion

This paper focused on exploring the capabilities of generalist LLMs in answering queries, with explanations, on a benchmark schedule to determine the potential for enabling trust in automated systems for the future. Existing research exposed the limited number of studies investigating the use of generalist LLMs to advance the understanding of automated scheduling systems and establish a means of XAI.

Benchmark schedules were selected to create a baseline dataset of varying sizes and complexity derived from the Job Shop concept of scheduling to set out the novel experiment. Query categories were defined to challenge the LLMs with different temporal and logical considerations for swapping or modifying elements of the provided schedule datasets. A single, common prompt was designed to trigger the question-answer with a single example query provided for each query category as a few-shot learning approach for the LLMs. The method of analysing the answer responses to the varied sizes of benchmark schedules was introduced along with several assessment criteria calculating the number of correct responses, as well as the use of language within each response.

The results showed the GPT-4 was correct more often than not, with more inaccurate responses from LLaMA, and the language used throughout the experiment was largely concise, complete, consistent, and aligned to human interpretation. While there were clear fluctuations in the assessment of some of the LLMs responses, the performance of the LLMs was not influenced by the size or complexity of the schedule datasets, highlighting the potential for this approach to be introduced to real-world applications, such as construction planning or manufacturing scheduling, and much larger schedules. The performance is also encouraging as the generalist LLMs from this experiment were not pre-trained or supplemented

with specialist knowledge, presenting the opportunity for further enhancement of the success achieved in this paper.

Future and further studies should consider introducing prompt optimisation techniques to explore the potential of increasing overall correctness scores. Additionally, as in several cases, there were identifiable query and response pairs that performed exceptionally, either negatively or positively, from the others in the same query category, which could be captured and introduced as a feedback loop to improve overall performance. Altering the model hyper-parameter settings could also impact the performance, as well as comparing the performance of additional alternative generalist LLM models. Furthermore, in the event of a domain-specific, scheduling management focused LLM for development, the approach established in this paper should be investigated for performance differences and improvements. Finally, testing this approach on a real-world application and data, integrated with an automated scheduling system, could directly prove the capability of AI for scheduling while providing in-built explainability and feedback loop to enable greater trust in wider adoption.

Appendix A: Results for all the categories and query types

A.1 Benchmark "yes" and "no" answers for each query category

Table 9 Benchmark answers for each query category

Query Category	YES	NO
Swap	1. Yes, an exchange of the start times of job 7 with job 12 on machine 5 can take place. There will not be any overlap or conflict with other jobs and rescheduling would not be required.	1. No, the exchange of start times for job 12 and job 3 on machine 9 cannot be done as this would cause a conflict with other jobs within the schedule. To make this exchange possible it would be required to reschedule all activities.
	2. Yes, it is possible to exchange the end times of job 7 with job 12 on machine 5. There will not be any overlap or conflict with other jobs and rescheduling would not be required.	2. No, it is not possible to exchange the end times of job 6 with job 13 on machine 1 as there would be a conflict with other jobs in the schedule.
	3. Yes, it can be considered feasible to exchange jobs 9 and 14 on machine 12, as there are no conflicts or overlaps that would prevent this from occurring.	3. No, there is no feasible option in the current schedule to exchange jobs 11 and 2 on machine 3. If the exchange took place there would be overlaps with other jobs and would require a complete reschedule to find a feasible solution.
	4. Yes, it appears possible to exchange job 7 between machines 8 and 11. This will not cause any overlaps or scheduling conflicts with other jobs or machines.	4. No, it does not appear possible to exchange job 9 between machines 1 and 5, as doing so would cause a conflict with other jobs in the schedule. Jobs cannot overlap when being processed on machines and therefore this exchange cannot be completed.

Table 9 continued

Query Category	YES	NO
	5. Yes, there is no reason the exchange of processing times of Job 6 on machine 12 with Job 9 on machine 9 cannot be completed, as there are no identified conflicts preventing this action.	5. No, this exchange of processing times between job 11 on machine 3 with job 2 on machine 13 is not possible as this would lead to an overlap with other jobs in the schedule and would require rescheduling in order to make this possible.
	6. Yes, it would be allowable and possible to swap the start times of jobs 8 and 4 on machines 2 and 12 respectively. There are no overlaps with other jobs that would prevent this from being possible.	6. No, the start times of job 5 on machine 7 and job 14 on machine 2 cannot be swapped as this will cause overlap and conflict with other jobs within the schedule. The schedule would need to be completely modified to allow this to happen.
	7. Yes, swapping the end times of job 9 on machine 11 with job 4 on machine 7 can be done, as there are no issues with other jobs that could stop this from happening.	7. No, the swapping of the end times of job 6 on machine 13 with job 15 on machine 15 is not achievable due to the conflicts and overlaps this would trigger with other jobs within the schedule. A full reschedule would be required to make this possible.
Increase	1. Yes, it would be possible to increase the duration of job 8 on machine 4 by 10 minutes as there is sufficient slack in the schedule to allow this without issue.	1. No, it wouldn't be possible to increase the duration of job 3 on machine 14 by 11 minutes as this would cause an overlap with the jobs starting later on this machine and would therefore require a complete reschedule.
	2. Yes, the overall running time of machine 7 can be increased by 12 minutes without impacting the overall scheduled completion time, as its increased finishing time does not exceed the scheduled completion time.	2. No, the overall schedule run time will be impacted by increasing the running time of machine 13, as this will exceed the current schedule completion time and therefore the increase is not possible.
	3. Yes, it is possible for the start time of job 9 on machine 14 to be increased by 20 minutes as this increase does not affect the start or completion of other jobs within the schedule.	3. No, its impossible to increase the start of job 11 on machine 1 as the consequence of this would trigger conflicts and overlaps with other jobs within the schedule and therefore a complete reschedule would be required.
	4. Yes, it is feasible to increase the end time of job 5 on machine 6 by 12 minutes as there is adequate capacity for the schedule to tolerate this without requiring a complete reschedule.	4. No, increasing the end time of job 2 on machine 3 is not feasible as this would conflict with the start time of other jobs within the schedule and would therefore require a complete reschedule to satisfy this requirement.
Decrease	1. Yes, there is the possibility to decrease the duration of job 8 on machine 5 by 13 minutes as this will not cause any conflict with other jobs or breach scheduling rules.	1. No, the duration of job 10 on machine 15 cannot be decreased by 16 minutes as this will cause the job to breach scheduling rules or conflict with other jobs within the schedule.
	2. Yes, the overall run time of machine 7 can be reduced by 17 minutes without impacting the overall schedule, as this change keeps the maximum schedule run time the same.	2. No, this is not possible as the overall schedule run time is impacted by reducing the overall running time of machine 9 by 16 minutes and therefore cannot be achieved without a complete reschedule.

Table 9 continued

Query Category	YES	NO
	3. Yes, the start time of job 2 on machine 14 can be reduced by 18 minutes, as this does not cause any overlap with existing jobs nor break any of the scheduling rules.	3. No, the start time of job 18 on machine 3 cannot be reduced by 16 minutes as this would cause an overlap with an existing job or breach the scheduling rules.
	4. Yes, it would be possible to reduce the end time of job 17 on machine 12 by 14 minutes as this will not have an impact on any other jobs or the operation of the schedule overall.	4. No, the end time of job 15 on machine 6 cannot be reduced by 14 minutes due to this breaching the scheduling rules defined for the problem.

A.2 Results for the Swap category and all questions

Table 10 Average results for the Swap category 15jx15m all schedules for GPT-4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	0.6	0.7827	0.9600	59.2	0.6085
2	0.8	0.8169	0.9482	**68.2**	**0.6726**
3	0.4	0.7818	0.9305	50.0	0.6600
4	0.6	0.7926	0.9425	62.4	0.6111
5	0.8	0.8300	0.9631	52.8	0.6711
6	0.8	**0.8308**	**0.9682**	65.2	0.6288
7	**1.0**	0.8295	0.9670	67.2	0.6066
Avg	0.71	0.8092	0.9542	60.7	0.6370
15jx15m schedule 2					
1	0.6	0.8074	0.9593	57.0	**0.6801**
2	0.6	0.8289	0.9501	71.0	0.6188
3	0.8	0.7610	0.9228	64.8	0.5861
4	0.4	0.8086	0.9320	63.2	0.6324
5	0.2	0.7847	0.9585	58.6	0.6514
6	0.6	**0.8751**	**0.9733**	68.4	0.6310
7	**1.0**	0.8231	0.9584	**74.2**	0.6281
Avg	0.60	0.8127	0.9506	65.3	0.6326
15jx15m schedule 3					
1	0.4	0.8380	0.9614	54.6	**0.6604**
2	**1.0**	0.8000	0.9441	72.2	0.5953
3	0.8	0.7774	0.9149	77.4	0.5721
4	0.8	0.7428	0.9237	78.6	0.6062
5	0.4	0.8425	0.9563	69.8	0.6149
6	**1.0**	0.8235	**0.9740**	70.6	0.6139

Table 10 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
7	0.8	**0.8595**	0.9299	**81.2**	0.6133
Avg	0.74	0.8120	0.9435	72.1	0.6109
15jx15m schedule 4					
1	0.8	0.8279	0.9610	63.2	0.5962
2	**1.0**	0.8278	0.9527	72.0	0.6478
3	0.8	0.8010	0.9154	63.0	0.6214
4	**1.0**	0.7924	0.9376	62.4	0.6104
5	0.4	**0.8420**	**0.9698**	45.2	**0.6712**
6	0.6	0.8130	0.9313	60.8	0.6301
7	**1.0**	0.8404	0.9659	**77.6**	0.6062
Avg	0.80	0.8206	0.9477	63.5	0.6262
15jx15m schedule 5					
1	0.6	0.8040	0.9537	59.4	0.6417
2	**1.0**	**0.8228**	0.9550	72.0	**0.6721**
3	0.8	0.7990	0.9207	**74.4**	0.5621
4	0.6	0.7460	0.9430	54.6	0.6607
5	0.4	0.8080	0.9620	60.2	0.6464
6	0.4	0.8194	**0.9722**	59.0	0.6653
7	**1.0**	0.8218	0.8153	71.0	0.6285
Avg	0.69	0.8030	0.9317	64.4	0.6395
15jx15m schedule 6					
1	0.6	0.8096	0.9640	58.4	0.6369
2	0.8	0.8097	0.9567	65.2	0.6548
3	0.8	**0.8258**	0.9264	74.4	0.5651
4	0.4	0.7981	0.9490	48.6	**0.6583**
5	0.6	0.7703	**0.9670**	53.2	0.6544
6	**1.0**	0.8201	0.9627	63.4	0.6163
7	**1.0**	0.8056	0.9527	**80.8**	0.5848
Avg	0.74	0.8056	0.9541	63.4	0.6244
15jx15m schedule 7					
1	0.6	**0.8443**	0.9579	50.4	0.6473
2	0.8	0.7771	0.9389	70.2	0.6620
3	0.4	0.7765	0.9259	57.4	0.6572
4	0.2	0.8326	0.9128	49.2	**0.6802**
5	0.6	0.7626	0.9681	49.0	0.6490
6	0.4	0.8123	**0.9734**	61.6	0.6452
7	**1.0**	0.8249	0.9644	**79.2**	0.6205
Avg	0.57	0.8043	0.9488	59.6	0.6516

Table 11 Average results for the Swap category 15jx15m all schedules for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	0.2	0.7999	0.9604	50.2	0.6727
2	0.0	0.7719	0.9025	**80.2**	0.6057
3	0.2	0.7981	0.9381	64.4	0.5872
4	0.2	0.8087	0.9161	64.4	0.5493
5	**0.4**	**0.8572**	**0.9736**	39.2	**0.6751**
6	0.2	0.7776	0.9566	62.0	0.6212
7	0.0	0.8216	0.9579	71.4	0.5371
Avg	0.17	0.8050	0.9436	61.7	0.6069
15jx15m schedule 2					
1	0.0	0.8190	0.9600	63.0	**0.6165**
2	0.2	**0.8443**	0.9625	70.4	0.5843
3	0.2	0.7926	0.9310	62.0	0.6005
4	0.0	0.8175	0.9029	61.0	0.5850
5	**0.4**	0.7821	0.9580	69.8	0.5681
6	**0.4**	0.8310	0.9596	54.8	0.6156
7	0.2	0.7990	**0.9649**	**74.4**	0.5932
Avg	0.20	0.8122	0.9484	65.1	0.5947
15jx15m schedule 3					
1	0.0	0.8092	0.9586	65.4	0.6057
2	0.0	0.8291	0.9544	**80.4**	0.5822
3	0.0	0.7926	0.9310	53.4	0.6358
4	0.0	0.7647	0.9179	60.2	0.6203
5	**0.4**	**0.8590**	**0.9676**	54.8	0.6163
6	0.2	0.7628	0.9266	48.2	**0.6700**
7	0.0	0.8122	0.9518	57.4	0.6232
Avg	0.09	0.8042	0.9440	60.0	0.6219
15jx15m schedule 4					
1	0.2	0.8171	0.9662	34.8	0.6843
2	0.0	0.8478	0.9633	**67.0**	0.6127
3	0.0	0.8126	0.9540	43.6	**0.6845**
4	0.0	0.8108	0.9242	66.6	0.6048
5	**0.6**	0.8297	0.9609	57.2	0.5912
6	0.2	**0.8623**	**0.9724**	55.8	0.6183
7	0.0	0.7999	0.9680	61.6	0.6175
Avg	0.14	0.8257	0.9584	55.2	0.6305

Table 11 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 5					
1	0.2	0.8139	0.9591	56.4	0.6510
2	0.4	0.8423	0.9638	65.6	0.5982
3	0.0	0.8257	0.9516	52.8	0.6046
4	0.2	0.7264	0.9221	53.4	0.5971
5	**0.6**	0.8275	**0.9715**	44.6	**0.6643**
6	**0.6**	0.7147	0.8734	60.8	0.6019
7	0.0	**0.8500**	0.9639	**73.2**	0.5640
Avg	0.29	0.8001	0.9436	58.1	0.6116
15jx15m schedule 6					
1	0.0	0.7909	0.9608	43.0	0.6479
2	0.0	0.7829	0.9480	**72.4**	0.5712
3	0.0	0.7548	0.9339	59.4	0.5795
4	0.0	0.7276	0.9054	62.4	0.5422
5	**0.4**	0.7924	**0.9799**	34.4	**0.6718**
6	0.2	**0.8326**	0.9727	45.0	0.6624
7	0.2	0.7408	0.9533	62.8	0.5574
Avg	0.11	0.7746	0.9506	54.2	0.6046
15jx15m schedule 7					
1	0.2	**0.8445**	0.9694	41.0	**0.6662**
2	**0.6**	0.8294	0.9620	**74.4**	0.5703
3	0.2	0.7561	0.9252	65.6	0.6183
4	0.4	0.7994	0.9097	55.8	0.6185
5	0.4	0.7722	**0.9702**	56.2	0.6282
6	0.2	0.8050	0.9673	52.2	0.6334
7	0.0	0.7955	0.9631	59.4	0.6279
Avg	0.29	0.8003	0.9524	57.8	0.6233

Table 12 Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for GPT-4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	0.4	**0.8527**	0.9698	37.0	**0.6957**
2	**1.0**	0.8176	0.9456	**74.0**	0.6510
3	0.6	0.7662	0.9302	55.2	0.6266
4	0.4	0.7486	0.9450	65.4	0.5990
5	0.0	0.8104	**0.9704**	53.8	0.6514
6	0.6	0.8167	0.9686	67.2	0.6509
7	**1.0**	0.8386	0.8531	66.6	0.6294
Avg	0.57	0.8073	0.9404	59.9	0.6434
20jx20m					
1	0.4	0.8017	0.9612	59.2	0.6497
2	0.6	0.7709	0.9551	59.2	0.6377
3	0.6	0.7613	0.9371	59.8	0.6100
4	0.6	0.7302	0.9379	57.8	**0.6534**
5	0.6	0.8242	0.9609	69.0	0.6234
6	**1.0**	**0.8394**	**0.9710**	74.8	0.6114
7	**1.0**	0.7932	0.9594	**81.6**	0.6303
Avg	0.69	0.7887	0.9547	65.9	0.6308
30jx20m					
1	0.8	0.7872	0.9512	68.0	0.6153
2	0.8	**0.8102**	0.9454	63.8	**0.7231**
3	0.6	0.7535	0.9399	58.0	0.6353
4	0.6	0.7417	0.9393	62.0	0.6713
5	0.4	0.7949	**0.9714**	57.4	0.6499
6	0.2	0.8040	0.9683	57.8	0.6633
7	**1.0**	0.8054	0.9580	**82.4**	0.6176
Avg	0.63	0.7853	0.9533	64.2	0.6537

Table 13 Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	**0.0**	0.8284	0.9547	52.8	0.6430
2	**0.0**	**0.8516**	0.9537	**73.0**	0.5414
3	**0.0**	0.7884	0.9257	67.8	0.5702
4	**0.0**	0.7794	0.8911	64.6	0.5971
5	**0.0**	0.8088	**0.9725**	47.2	**0.6576**
6	**0.0**	0.8390	0.9723	57.2	0.6443
7	**0.0**	0.7979	0.9237	71.0	0.5509
Avg	0.00	0.8133	0.9419	61.9	0.6006
20jx20m					
1	0.2	0.7636	0.9592	63.4	**0.6438**
2	**0.6**	0.8165	0.9608	68.4	0.5780
3	0.2	0.8061	0.9327	64.6	0.5445
4	0.2	0.7802	0.9082	**69.0**	0.6168
5	0.4	0.8233	0.9666	50.8	0.6376
6	0.0	**0.8256**	**0.9720**	57.0	0.6181
7	0.2	0.7885	0.9618	68.4	0.6138
Avg	0.26	0.8006	0.9516	63.1	0.6075
30jx20m					
1	**0.2**	0.7734	0.9602	57.4	0.6357
2	**0.2**	**0.8279**	0.9627	60.0	0.6327
3	0.0	0.7565	0.9396	60.0	0.5856
4	0.0	0.7503	0.9191	**76.4**	0.5795
5	**0.2**	0.8232	**0.9702**	43.6	**0.6538**
6	**0.2**	0.7981	0.9630	63.8	0.6080
7	**0.2**	0.8057	0.9647	66.0	0.6048
Avg	0.14	0.7907	0.9542	61.0	0.6143

A.3 Results for the increase category and all questions

Table 14 Average results for the Increase category 15jx15m all schedules for GPT-4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	**1.0**	0.8489	**0.9707**	49.0	**0.7076**
2	0.4	0.8146	0.9567	56.0	0.6582
3	0.2	**0.8915**	0.9615	51.0	0.6796
4	0.6	0.8567	0.9575	**59.4**	0.6817
Avg	0.55	0.8529	0.9616	53.9	0.6818
15jx15m schedule 2					
1	0.8	0.8313	0.9512	62.0	**0.6666**
2	0.4	0.7833	0.9534	60.4	0.6540
3	**1.0**	**0.8580**	**0.9610**	**73.8**	0.6241
4	0.6	0.8547	0.9514	67.0	0.6303
Avg	0.70	0.8319	0.9543	65.8	0.6437
15jx15m schedule 3					
1	0.6	0.8183	0.9601	49.8	**0.7426**
2	0.6	**0.8682**	0.9651	67.0	0.6700
3	**0.8**	0.8507	**0.9741**	35.4	0.7202
4	0.6	0.8385	0.9521	**74.6**	0.6200
Avg	0.65	0.8439	0.9628	56.7	0.6882
15jx15m schedule 4					
1	0.4	0.8181	0.9537	59.8	0.6636
2	0.4	0.8040	0.9548	62.8	0.6742
3	0.2	**0.8556**	**0.9674**	47.2	**0.7361**
4	**0.6**	0.8451	0.9588	**63.0**	0.6586
Avg	0.40	0.8307	0.9587	58.2	0.6831
15jx15m schedule 5					
1	**1.0**	0.8338	**0.9748**	38.4	**0.7758**
2	0.4	0.8021	0.9655	**73.8**	0.6407
3	0.8	**0.8353**	0.9585	57.6	0.6498
4	0.8	0.8302	0.9452	67.2	0.6446
Avg	0.75	0.8254	0.9610	59.3	0.6777
15jx15m schedule 6					
1	**0.8**	0.8504	0.9614	35.0	**0.7752**
2	0.4	0.7640	0.9543	51.6	0.6689
3	0.6	0.8340	**0.9647**	49.8	0.6777
4	**0.8**	**0.8765**	0.9606	**67.6**	0.6371
Avg	0.65	0.8312	0.9602	51.0	0.6897

Table 14 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 7					
1	0.6	0.8456	0.9524	54.4	**0.6935**
2	0.0	0.7803	**0.9575**	59.0	0.6637
3	0.4	**0.8714**	0.9553	57.4	0.6687
4	**1.0**	0.8356	0.9494	**72.4**	0.6206
Avg	0.50	0.8332	0.9536	60.8	0.6616

Table 15 Average results for the Increase category 15jx15m all schedules for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	0.0	0.8179	0.9620	61.6	0.6366
2	0.6	0.7216	0.9290	**76.4**	0.6021
3	0.4	**0.8970**	**0.9678**	62.4	0.6286
4	**0.8**	0.7895	0.9454	55.4	**0.6526**
Avg	0.45	0.8065	0.9511	64.0	0.6300
15jx15m schedule 2					
1	**0.6**	0.7938	0.9503	**58.4**	0.6382
2	**0.6**	0.7759	0.9567	53.4	0.6813
3	0.2	0.8460	**0.9674**	51.2	**0.6988**
4	0.0	**0.8620**	0.9514	56.6	0.6029
Avg	0.35	0.8194	0.9564	54.9	0.6553
15jx15m schedule 3					
1	0.2	0.8297	**0.9725**	**60.0**	0.6595
2	0.2	**0.8563**	0.9697	56.4	0.6415
3	0.0	0.8477	0.9703	50.2	**0.7090**
4	**0.6**	0.8154	0.9453	57.4	0.6288
Avg	0.25	0.8373	0.9645	56.0	0.6597
15jx15m schedule 4					
1	0.4	0.8526	**0.9693**	45.4	**0.7102**
2	**0.6**	0.7926	0.9495	**72.6**	0.6489
3	0.0	**0.8557**	0.9633	65.2	0.6459
4	0.4	0.7970	0.9557	50.4	0.6290
Avg	0.35	0.8245	0.9595	58.4	0.6585

Table 15 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 5					
1	0.2	0.8181	**0.9710**	47.8	0.6713
2	0.4	0.7739	0.9602	**67.8**	0.5981
3	**0.6**	**0.8357**	0.9608	52.4	**0.6810**
4	0.4	0.8011	0.9507	55.6	0.6627
Avg	0.40	0.8072	0.9607	55.9	0.6533
15jx15m schedule 6					
1	0.2	0.8252	**0.9592**	59.8	0.6123
2	**0.8**	0.7900	0.9538	63.0	0.6204
3	0.2	0.7968	0.9555	61.8	**0.6708**
4	0.0	**0.8317**	0.9471	**64.2**	0.6010
Avg	0.3	0.8109	0.9539	62.2	0.6261
15jx15m schedule 7					
1	0.4	0.8324	**0.9661**	48.4	0.6787
2	**0.8**	0.7985	0.9612	**69.0**	0.6134
3	0.0	**0.8502**	0.9594	62.4	**0.6849**
4	0.0	0.8189	0.9542	53.0	0.6519
Avg	0.30	0.8250	0.9602	58.2	0.6572

Table 16 Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for GPT-4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	0.6	0.7880	**0.9626**	56.4	**0.7113**
2	0.6	0.8071	0.9586	64.4	0.6539
3	**0.8**	**0.8367**	0.9616	56.2	0.6778
4	0.4	0.8243	0.9414	**73.4**	0.6769
Avg	0.6	0.8140	0.9560	62.6	0.6800
20jx20m					
1	0.8	0.8073	0.9635	55.2	0.6986
2	0.2	0.7911	**0.9674**	57.4	0.6779
3	0.6	**0.8698**	0.9657	45.2	**0.7148**
4	**1.0**	0.7970	0.9482	**62.2**	0.6983
Avg	0.65	0.8163	0.9612	55.0	0.6974
30jx20m					
1	**1.0**	0.8214	**0.9657**	48.4	**0.7555**

Table 16 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
2	0.4	0.7704	0.9646	44.8	0.6813
3	0.4	**0.8382**	0.9586	**58.6**	0.6550
4	0.4	0.7989	0.9470	58.4	0.6863
Avg	0.55	0.8072	0.9589	52.6	0.6945

Table 17 Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	**0.4**	0.8000	**0.9671**	64.0	0.6345
2	**0.4**	0.8156	0.9638	73.0	**0.6772**
3	0.2	0.8323	0.9585	69.6	0.6739
4	0.0	**0.8407**	0.9461	**75.8**	0.6258
Avg	0.25	0.8221	0.9589	70.6	0.6529
20jx20m					
1	0.2	0.8048	0.9629	62.0	0.6392
2	**0.8**	0.7653	0.9607	**73.8**	0.6043
3	0.2	**0.8610**	**0.9725**	44.2	**0.6903**
4	0.0	0.7972	0.9416	50.2	0.6649
Avg	0.30	0.8071	0.9594	57.6	0.6497
30jx20m					
1	0.2	0.8333	**0.9742**	40.4	**0.7343**
2	**0.4**	0.8006	0.9674	55.6	0.6579
3	**0.4**	**0.8349**	0.9621	56.8	0.6492
4	**0.4**	0.7257	0.9309	**62.6**	0.6047
Avg	0.35	0.7986	0.9586	53.9	0.6615

A.4 Results for the decrease category and all questions

Table 18 Average results for the Decrease category 15jx15m all schedules for GPT-4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	0.6	0.8120	0.9608	43.2	**0.7437**
2	0.2	0.7668	0.9651	**52.6**	0.6770
3	**1.0**	0.8386	0.9478	49.2	0.7222
4	0.6	**0.8529**	**0.9740**	49.4	0.7282
Avg	0.60	0.8176	0.9619	48.6	0.7178
15jx15m schedule 2					
1	0.4	0.8028	0.9589	51.0	**0.7308**
2	0.2	0.8205	**0.9670**	**77.4**	0.6338
3	**1.0**	**0.8606**	0.9436	60.6	0.6870
4	0.6	0.8278	0.9646	65.6	0.7163
Avg	0.55	0.8279	0.9585	63.7	0.6920
15jx15m schedule 3					
1	0.2	0.7991	0.8327	54.0	0.6915
2	0.4	0.8044	0.9605	**78.4**	0.6347
3	**1.0**	0.8238	0.9414	64.2	0.6809
4	0.4	**0.8783**	**0.9639**	64.8	**0.6934**
Avg	0.50	0.8264	0.9246	65.4	0.6751
15jx15m schedule 4					
1	0.4	0.7861	0.8371	51.0	0.7013
2	0.2	0.7592	**0.9646**	50.6	0.6960
3	**1.0**	**0.8137**	0.9405	50.0	**0.7150**
4	0.4	0.7842	0.9589	**66.8**	0.6506
Avg	0.50	0.7858	0.9252	54.6	0.6907
15jx15m schedule 5					
1	0.2	0.7691	0.7342	39.6	0.7102
2	0.0	0.8259	**0.9653**	**80.2**	0.6178
3	**1.0**	**0.8686**	0.9507	52.0	**0.7196**
4	0.4	0.8246	0.9649	62.2	0.6786
Avg	0.40	0.8220	0.9038	58.5	0.6815
15jx15m schedule 6					
1	0.4	0.8011	0.8409	47.8	0.7155
2	0.2	0.7955	0.9687	**61.8**	0.6525
3	**0.8**	0.8275	0.9422	59.4	0.6866
4	0.4	**0.8477**	**0.9730**	47.0	**0.7193**
Avg	0.45	0.8179	0.9312	54.0	0.6935

Table 18 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 7					
1	0.6	0.7770	0.8405	41.6	**0.7400**
2	0.8	0.8250	0.9652	56.0	0.6669
3	**1.0**	**0.8497**	0.9493	60.2	0.6874
4	0.8	0.8230	**0.9664**	**65.2**	0.6759
Avg	0.80	0.8187	0.9303	55.8	0.6926

Table 19 Average results for the Decrease category 15jx15m all schedules for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 1					
1	**0.8**	**0.8207**	**0.9662**	56.6	0.6471
2	0.6	0.7796	0.9593	**66.4**	**0.6713**
3	0.2	0.8118	0.9530	62.2	0.6564
4	**0.8**	0.8113	0.9564	**66.4**	0.6169
Avg	0.60	0.8058	0.9587	62.9	0.6479
15jx15m schedule 2					
1	**0.8**	0.7917	0.9536	70.0	0.6548
2	0.4	0.8054	**0.9645**	**75.0**	0.5981
3	0.4	**0.8346**	0.9423	66.4	0.6118
4	**0.8**	0.8051	0.9633	65.8	**0.6581**
Avg	0.60	0.8092	0.9559	69.3	0.6307
15jx15m schedule 3					
1	0.4	0.8161	**0.9727**	60.2	0.6394
2	**0.8**	0.8386	0.9661	67.4	0.5927
3	0.0	**0.8591**	0.9546	66.8	**0.6422**
4	**0.8**	0.8578	0.9626	**73.2**	0.5753
Avg	0.50	0.8429	0.9640	66.9	0.6124
15jx15m schedule 4					
1	**0.8**	0.7986	**0.9617**	53.2	0.6292
2	0.6	0.7832	0.9564	**71.4**	**0.6361**
3	0.0	**0.8508**	0.9574	60.2	0.6345
4	0.6	0.7527	0.9444	63.4	0.6184
Avg	0.50	0.7963	0.9550	62.1	0.6296

Table 19 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
15jx15m schedule 5					
1	**1.0**	0.7593	0.8927	64.0	0.6410
2	0.8	0.7845	**0.9646**	62.0	0.5492
3	0.0	**0.8543**	0.9576	50.0	**0.6703**
4	0.8	0.7984	0.9574	**75.8**	0.5672
Avg	0.65	0.7991	0.9431	63.0	0.6069
15jx15m schedule 6					
1	**1.0**	**0.8174**	0.9572	61.0	0.6308
2	**1.0**	0.7867	**0.9596**	69.6	0.6398
3	0.2	0.8172	0.9567	60.2	**0.6400**
4	0.8	0.7993	0.9566	**77.8**	0.6340
Avg	0.75	0.8051	0.9575	67.2	0.6362
15jx15m schedule 7					
1	0.6	0.8062	**0.9702**	47.6	**0.6710**
2	0.2	0.8160	0.9623	74.6	0.6413
3	0.2	**0.8494**	0.9631	53.2	0.6584
4	**0.8**	0.7846	0.9551	**76.6**	0.6078
Avg	0.45	0.8141	0.9627	63.0	0.6446

Table 20 Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for GPT 4

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	0.2	0.8341	0.9545	40.6	0.7033
2	0.0	0.7824	**0.9688**	49.4	0.6852
3	**0.8**	**0.8564**	0.9528	46.8	**0.7478**
4	0.6	0.8196	0.9643	**66.2**	0.6734
Avg	0.40	0.8231	0.9601	50.8	0.7024
20jx20m					
1	0.2	0.8052	0.8492	38.8	**0.7107**
2	0.0	0.8004	0.9621	**73.6**	0.6494
3	**0.6**	0.8349	0.9422	72.8	0.6897
4	0.4	**0.8753**	**0.9681**	54.0	0.7052
Avg	0.30	0.8290	0.9304	59.8	0.6887
30jx20m					
1	0.4	0.7849	0.8473	62.4	0.7044
2	0.2	0.7770	0.9589	70.0	0.6610

Table 20 continued

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
3	**0.6**	0.8135	0.9426	56.2	**0.7138**
4	0.2	**0.8318**	**0.9656**	**71.2**	0.6607
Avg	0.35	0.8018	0.9263	65.0	0.6850

Table 21 Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score f1
20jx15m					
1	0.6	0.8184	**0.9659**	66.2	0.6528
2	**1.0**	0.7590	0.9522	**72.0**	0.6426
3	0.0	**0.8420**	0.9519	68.8	**0.6652**
4	0.8	0.8173	0.9641	62.6	0.6303
Avg	0.60	0.8092	0.9585	67.4	0.6477
20jx20m					
1	**0.8**	0.8354	**0.9676**	54.4	0.6253
2	0.6	0.8180	0.9660	**70.6**	**0.6453**
3	0.0	**0.8657**	0.9614	58.2	0.6437
4	0.6	0.8424	0.9621	69.0	0.6032
Avg	0.50	0.8404	0.9643	63.1	0.6294
30jx20m					
1	**0.8**	**0.8242**	**0.9648**	66.8	0.6291
2	0.4	0.7836	0.9637	**68.2**	0.6073
3	0.0	0.8141	0.9505	67.0	0.6195
4	**0.8**	0.7857	0.9608	52.0	**0.6544**
Avg	0.50	0.8019	0.9600	63.5	0.6276

Acknowledgements This study was half-funded by ESA under the OSIP Co-Sponsored PhD activity: "Robust and Explainable Mission Planning and Scheduling (REMPS)" No. 4000132894/20/NL/MH/hm. The authors would also like to acknowledge the support of ESA through the Visiting Researcher program. Finally, they wish to warmly thank Dimitris Kardaris (ESA) for all the insightful discussion about mission operations constraints and opportunities for explainability.

Author Contributions C.P. made substantial contributions to the conception or design of the work; the acquisition, analysis, and interpretation of data; wrote and reviewed the manuscript and prepared all figures and tables. A.R. assisted in the conceptualization of the work and reviewed the manuscript.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing Interests The authors declare no competing interests.

References

Al Homssi, B., Dakic, K., Wang, K., et al. (2024). Artificial intelligence techniques for next-generation massive satellite networks. *IEEE Communications Magazine, 62*(4), 66–72. https://doi.org/10.1109/MCOM.004.2300277

Ali, S., Abuhmed, T., El-Sappagh, S., et al. (2023). Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence. *Information Fusion, 99*(101), 805. https://doi.org/10.1016/j.inffus.2023.101805

Amer, F., Hockenmaier, J., & Golparvar-Fard, M. (2022). Learning and critiquing pairwise activity relationships for schedule quality control via deep learning-based natural language processing. *Automation in Construction, 134*(104), 036. https://doi.org/10.1016/j.autcon.2021.104036

Amer, F., Koh, H. Y., & Golparvar-Fard, M. (2021). Automated methods and systems for construction planning and scheduling: Critical review of three decades of research. *Journal of Construction Engineering and Management, 147*(7), 03121002. https://doi.org/10.1061/(asce)co.1943-7862.0002093

Arrieta, A. B., Díaz-Rodríguez, N., Del Ser, J., et al. (2020). Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion, 58*, 82–115. https://doi.org/10.1016/j.inffus.2019.12.012

Atsmony, M., & Mosheiov, G. (2022). Scheduling to maximize the weighted number of on-time jobs on parallel machines with bounded job-rejection. *Journal of Scheduling, 26*(2), 193–207. https://doi.org/10.1007/s10951-022-00745-7

Bach, S., Binder, A., Montavon, G., et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE, 10*(7), e0130140. https://doi.org/10.1371/journal.pone.0130140

Bastola, A., Wang, H., Hembree, J., et al. (2023). *Llm-based smart reply (lsr): Enhancing collaborative performance with chatgpt-mediated smart reply system.*https://doi.org/10.48550/arXiv.2306.11980

Ben Abdallah, E., Grati, R., & Boukadi, K. (2023). Towards an explainable irrigation scheduling approach by predicting soil moisture and evapotranspiration via multi-target regression. *Journal of Ambient Intelligence and Smart Environments, 1–22*,. https://doi.org/10.3233/AIS-220477

Bernardi, M. L., Casciani, A., Cimitile, M., et al. (2024). Conversing with business process-aware large language models: the bpllm framework. *Journal of Intelligent Information Systems, 62*(6), 1607–1629. https://doi.org/10.1007/s10844-024-00898-1

Brinkkötter, W., & Brucker, P. (2001). Solving open benchmark instances for the job-shop problem by parallel head-tail adjustments. *Journal of Scheduling, 4*(1), 53–64. https://doi.org/10.1002/1099-1425(200101/02)4:1<53::AID-JOS59>3.0.CO;2-Y

Brown, T., Mann, B., Ryder, N., et al. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, Hadsell R., et al. (Eds.) *Advances in neural information processing systems* (Vol. 33, pp. 1877–1901). Curran Associates, Inhttps://doi.org/10.5555/3495724.3495883

Brucker, P., & Knust, S. (2006). Complex scheduling. *Springer*. https://doi.org/10.1007/3-540-29546-1

Chakraborti T, Sreedharan S, & Kambhampati S (2020) The emerging landscape of explainable automated planning & decision making. In C. Bessiere (Ed.) *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20, international joint conferences on artificial intelligence organization* (pp. 4803–4811). https://doi.org/10.24963/ijcai.2020/669, survey track

Chen, T. C. T. (2023). *Explainable Artificial Intelligence (XAI) in manufacturing* (pp. 1–11). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-031-27961-4_1

Čyras, K., Rago, A., Albini, E., et al. (2021). Argumentative xai: a survey. In Z.H. Zhou (Ed.) *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, international joint conferences on artificial intelligence organization* (pp. 4392–4399)https://doi.org/10.24963/ijcai.2021/600, survey Track

Datta, T., Dickerson, J.P. (2023). *Who's thinking? a push for human-centered evaluation of llms using the xai playbook*.https://doi.org/10.48550/arXiv.2303.06223

Ersoy, P., & Erşahin, M. (2024). Optimal llm execution strategies for llama 3.1 language models across diverse hardware configurations: a comprehensive guide. *Computational Intelligence and Machine Learning,5*. https://www.cimachinelearning.com/llm-execution-strategies.php

Face, H. (2024). *all-MiniLM-L6-v2 sentence transformer*.https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

Face, H. (2024). Metric: bert_score. https://huggingface.co/spaces/evaluate-metric/bertscore

Fikar, C., & Hirsch, P. (2017). Home health care routing and scheduling: A review. *Computers & Operations Research, 77*, 86–95. https://doi.org/10.1016/j.cor.2016.07.019

Francis, A. (2015). Graphical modelling classification for construction project scheduling. *Procedia Engineering, 123*, 162–168. https://doi.org/10.1016/j.proeng.2015.10.073, selected papers from Creative Construction Conference 2015

Friedrich, F., Schramowski, P., Tauchmann, C., et al. (2021). Interactively providing explanations for transformer language models. In: *HHAI*. https://doi.org/10.3233/faia220218

Fu, D., Huang, J., Lu, S., et al. (2024). *PreAct: Prediction enhances agent's planning ability*.https://doi.org/10.48550/arXiv.2402.11534

Gajane, P., Saxena, A., Tavakol, M., et al (2022). *Survey on fair reinforcement learning: Theory and practice*.https://doi.org/10.48550/arXiv.2205.10032

Gashi, M., Mutlu, B., & Thalmann, S. (2023). Impact of interdependencies: Multi-component system perspective toward predictive maintenance based on machine learning and xai. *Applied Sciences, 13*(5), 3088. https://doi.org/10.3390/app13053088

Glaese, A., McAleese, N., Trębacz, M., et al. (2022). *Improving alignment of dialogue agents via targeted human judgements*.https://doi.org/10.48550/arXiv.2209.14375

Goh, E., Venkataram, H.S., Balaji, B., et al. (2022). SatNet: A benchmark for satellite scheduling optimization. In: *AAAI-22 workshop on Machine Learning for Operations Research (ML4OR)*. Pasadena, CA: Jet Propulsion Laboratory, National Aeronautics and Space Administration, 2022. https://doi.org/2014/56106

Hager, P., Jungmann, F., Holland, R., et al. (2024). Evaluation and mitigation of the limitations of large language models in clinical decision-making. *Nature Medicine, 30*(9), 2613–2622. https://doi.org/10.1038/s41591-024-03097-1

Henning Dr rer nat A. (2002). *Practical job shop scheduling issues*. PhD thesis, Friedrich Schiller University, Jena. https://www.db-thueringen.de/receive/dbt_mods_00000873, dissertation, Friedrich Schiller Universityät Jena, 2003

Herrmann, A., & Schaub, H. (2023). Reinforcement learning for the agile earth-observing satellite scheduling problem. *IEEE Transactions on Aerospace and Electronic Systems, 59*(5), 5235–5247. https://doi.org/10.1109/TAES.2023.3251307

Jain, S., & Wallace, B.C. (2019). Attention is not explanation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* Vol. 1 (Long and Short Papers) (pp. 3543–3556). Minneapolis, Minnesota: Association for Computational Linguistics. https://doi.org/10.18653/v1/N19-1357

Jain, A. S., & Meeran, S. (1999). Deterministic job-shop scheduling: Past, present and future. *European Journal of Operational Research, 113*(2), 390–434. https://doi.org/10.1016/S0377-2217(98)00113-1

Jeong C (2024) Domain-specialized llm: Financial fine-tuning and utilization method using mistral 7b. Journal of Intelligence and Information Systems 30(1):93–120, https://doi.org/10.13088/jiis.2024.30.1.093

Kasneci, E., Sessler, K., Küchemann, S., et al. (2023). Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and Individual Differences, 103*(102), 274. https://doi.org/10.1016/j.lindif.2023.102274

Keane, M.T., Kenny, E.M., Delaney, E., et al. (2021). If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques. In Z.H., Zhou (Eds.), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21, international joint conferences on artificial intelligence organization* (pp. 4466–4474). https://doi.org/10.24963/ijcai.2021/609, survey Track

Kim, D., Song, Y., Kim, S., et al. (2023). How should the results of artificial intelligence be explained to users? - research on consumer preferences in user-centered explainable artificial intelligence. *Technological Forecasting and Social Change, 188*(122), 343. https://doi.org/10.1016/j.techfore.2023.122343

Lai, V., Chen, C., Liao, Q.V., et al. (2021). Towards a science of human-ai decision making: A survey of empirical studies. CoRR. https://doi.org/10.48550/arXiv.2112.11471

Lai, V., Chen, C., Smith-Renner, A., et al. (2023). Towards a science of human-ai decision making: An overview of design space in empirical human-subject studies. In *Proceedings of the 2023 ACM Conference on*

Fairness, Accountability, and Transparency, Association for Computing Machinery, New York, NY, USA, FAccT '23 (pp. 1369–1385). https://doi.org/10.1145/3593013.3594087

Marabelli, M., Newell, S., & Handunge, V. (2021). The lifecycle of algorithmic decision-making systems: Organizational choices and ethical challenges. *The Journal of Strategic Information Systems, 30*(3), 101683. https://doi.org/10.1016/j.jsis.2021.101683

Montavon, G., Binder, A., Lapuschkin, S., et al. (2019). *Layer-Wise Relevance Propagation: An overview* (pp. 193–209). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-28954-6_10

Moons, S., Ramaekers, K., Caris, A., et al. (2017). Integrating production scheduling and vehicle routing decisions at the operational decision level: A review and discussion. *Computers & Industrial Engineering, 104*, 224–245. https://doi.org/10.1016/j.cie.2016.12.010

Mo, Y., Zhao, D., Du, J., et al. (2020). Automated staff assignment for building maintenance using natural language processing. *Automation in Construction, 113*(103), 150. https://doi.org/10.1016/j.autcon.2020.103150

Mullins, B. (2023). *The shape of explanations: A topological account of rule-based explanations in machine learning*. https://doi.org/10.48550/arXiv.2301.09042

Narteni, S., Orani, V., Ferrari, E., et al. (2022). A new xai-based evaluation of generative adversarial networks for imu data augmentation. In *2022 IEEE international conference on e-health networking, application & services (HealthCom)* (pp 167–172). https://doi.org/10.1109/HealthCom54947.2022.9982780

Picard, G., Caron, C., Farges, J.L., et al. (2021). Autonomous agents and multiagent systems challenges in earth observation satellite constellations. In *Proceedings of the 20th international conference on autonomous agents and multiagent systems, International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, AAMAS '21* (pp. 39—44). https://doi.org/10.5555/3463952.3463961

Pillai, M., Liu, C. C., Kwong, E., et al. (2024). Using an explainable machine learning approach to prioritize factors contributing to healthcare professionals' burnout. *Journal of Intelligent Information Systems, 62*(4), 1113–1124. https://doi.org/10.1007/s10844-024-00862-z

Powell, C., Berquand, A., Riccardi, A. (2023). Natural language processing for explainable satellite scheduling. In *SPACEOPS 2023, ARE, p #349*. https://strathprints.strath.ac.uk/85129/

Prieto, S. A., Mengiste, E. T., & García de Soto, B. (2023). Investigating the use of chatgpt for the scheduling of construction projects. *Buildings,13*(4). https://doi.org/10.3390/buildings13040857

Puiutta, E., & Veith, E. M. (2020). Explainable reinforcement learning: A survey. *International cross-domain conference for machine learning and knowledge extraction* (pp. 77–95). Springer. https://doi.org/10.1007/978-3-030-57321-8_5

Rjoub, G., Bentahar, J., Abdel Wahab, O., et al. (2021). Deep and reinforcement learning for automated task scheduling in large-scale cloud computing systems. *Concurrency and Computation: Practice and Experience, 33*(23), e5919. https://doi.org/10.1002/cpe.5919

Roy, D., Zhang, X., Bhave, R., et al. (2024). Exploring llm-based agents for root cause analysis. In *Companion proceedings of the 32nd ACM international conference on the foundations of software engineering, association for computing machinery, New York, NY, USA, FSE 2024* (pp. 208—219). https://doi.org/10.1145/3663529.3663841

Saeed, W., & Omlin, C. (2023). Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities. *Knowledge-Based Systems, 263*(110), 273. https://doi.org/10.1016/j.knosys.2023.110273

Satterfield, N., Holbrooka, P., & Wilcoxa, T. (2024). Fine-tuning llama with case law data to improve legal domain performance. OSF. https://doi.org/10.31219/osf.io/e6mjs

Scao, T.L., Fan, A., Akiki, C., et al. (2022). Bloom: A 176b-parameter open-access multilingual language model. https://doi.org/10.48550/ARXIV.2211.05100

Schroder, M. (2023). Autoscrum: Automating project planning using large language models. https://doi.org/10.48550/arXiv.2306.03197

Shao, Z., Gong, Y., Shen, Y., et al. (2023). Synthetic prompting: Generating chain-of-thought demonstrations for large language models. In A. Krause, E. Brunskill, K. Cho, et al .(Eds.), *Proceedings of the 40th international conference on machine learning, PMLR, Proceedings of Machine Learning Research* (Vol. 202, pp. 30706–30775). https://doi.org/10.5555/3618408.3619681

Shuster, K., Xu, J., Komeili, M., et al. (2022). *Blenderbot 3: A deployed conversational agent that continually learns to responsibly engage.*https://doi.org/10.48550/arXiv.2208.03188

Shylo, O, (2002), *Best known lower and upper bounds: Job shop scheduling problem : Taillard's instances.*https://optimizizer.com/TA.php

Shylo, O.V., & Shams, H. (2018). Boosting binary optimization via binary classification: A case study of job shop scheduling. https://doi.org/10.48550/arXiv.1808.10813

Singh, C., Askari, A., Caruana, R., et al. (2023). Augmenting interpretable models with large language models during training. *Nature Communications*. https://doi.org/10.1038/s41467-023-43713-1

Singh, C., Askari, A., Caruana, R., et al. (2023). Augmenting interpretable models with large language models during training. *Nature Communications*. https://doi.org/10.1038/s41467-023-43713-1

Squires, M., Tao, X., Elangovan, S., et al. (2022). A novel genetic algorithm based system for the scheduling of medical treatments. *Expert Systems with Applications, 195*(116), 464. https://doi.org/10.1016/j.eswa.2021.116464

Sui, Y., Zhou, M., Zhou, M., et al. (2024). Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In Proceedings of the 17th ACM international conference on web search and data mining, association for computing machinery, New York, NY, USA, WSDM '24 (pp. 645—654). https://doi.org/10.1145/3616855.3635752

Taillard, E. (1997). Best lower and upper bounds known, from or-lib. http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/jobshop.dir/best_lb_up.txt

Taillard, E. (1993). Benchmarks for basic scheduling problems. *European Journal of Operational Research, 64*(2), 278–285. https://doi.org/10.1016/0377-2217(93)90182-M, project Management anf Scheduling

Thangavel, K., Spiller, D., Sabatini, R., et al. (2023). Trusted autonomous operations of distributed satellite systems using optical sensors. *Sensors,23*(6). https://doi.org/10.3390/s23063344

Thoppilan, R., De Freitas, D., Hall, J., et al. (2022). Lamda: Language models for dialog applications. https://doi.org/10.48550/arXiv.2201.08239

Turpin, M., Michael, J., Perez, E., et al. (2023). Language models don't always say what they think: unfaithful explanations in chain-of-thought prompting. In *Proceedings of the 37th international conference on neural information processing systems, Curran Associates Inc., Red Hook, NY, USA, NIPS '23*. https://doi.org/10.5555/3666122.3669397

Uhde, A., Schlicker, N., Wallach, D.P., et al. (2020). Fairness and decision-making in collaborative shift scheduling systems. In *Proceedings of the 2020 CHI conference on human factors in computing systems* (pp. 1–13). https://doi.org/10.1145/3313831.3376656

Vaswani, A., Shazeer, N., Parmar, N., et al. (2017). Attention is all you need. In *Proceedings of the 31st international conference on neural information processing systems, Curran Associates Inc., Red Hook, NY, USA, NIPS'17* (pp. 6000—6010). https://doi.org/10.5555/3295222.3295349

Wang, Y., Shi, X., & Zhao, X. (2024). Mllm4rec: multimodal information enhancing llm for sequential recommendation. *Journal of Intelligent Information Systems, 1–17*,. https://doi.org/10.1007/s10844-024-00915-3

Wei, J., Wang, X., Schuurmans, D., et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. In S. Koyejo, S. Mohamed, A. Agarwal, et al. (Eds.), *Advances in neural information processing systems* (Vol. 35, pp. 24824–24837). Curran Associates, Inc. https://doi.org/10.5555/3600270.3602070

Wiegreffe, S., & Pinter, Y. (2019). Attention is not not explanation. In conference on empirical methods in natural language processing. https://doi.org/10.48550/arXiv.1908.04626

Wu, X., Zhao, H., Zhu, Y., et al. (2024). Usable xai: 10 strategies towards exploiting explainability in the llm era. https://doi.org/10.48550/arXiv.2403.08946

Wu, T., He, S., Liu, J., et al. (2023). A brief overview of chatgpt: The history, status quo and potential future development. *IEEE/CAA Journal of Automatica Sinica, 10*(5), 1122–1136. https://doi.org/10.1109/JAS.2023.123618

Xiong, H., Shi, S., Ren, D., et al. (2022). A survey of job shop scheduling problem: The types and models. *Computers & Operations Research, 142*(105), 731. https://doi.org/10.1016/j.cor.2022.105731

Yang, J., Jin, H., Tang, R., et al. (2024). Harnessing the power of llms in practice: A survey on chatgpt and beyond. *ACM Transactions on Knowledge Discovery from Data,18*(6). https://doi.org/10.1145/3649506

Yang, X., Zhu, J., & De Meo, P. (2024). A quantum-like zero-shot approach for sentiment analysis in finance. *Journal of Intelligent Information Systems, 1–17*,. https://doi.org/10.1007/s10844-024-00912-6

Yang, X., Zhu, J., & De Meo, P. (2024). A quantum-like zero-shot approach for sentiment analysis in finance. *Journal of Intelligent Information Systems, 1–17*,. https://doi.org/10.1007/s10844-024-00912-6

Yao, S., Zhao, J., Yu, D., et al. (2023). React: Synergizing reasoning and acting in language models. In The eleventh international conference on learning representations. https://openreview.net/forum?id=WE_vluYUL-X

Yao, E., Liu, T., Lu, T., et al. (2020). Optimization of electric vehicle scheduling with multiple vehicle types in public transport. *Sustainable Cities and Society, 52*(101), 862. https://doi.org/10.1016/j.scs.2019.101862

Zaki, M., Jayadeva, Mausam, et al. (2024). Mascqa: investigating materials science knowledge of large language models. *Digital Discovery, 3*, 313–327. https://doi.org/10.1039/D3DD00188A

Zheng, Z., Ren, X., Xue, F., et al. (2023). Response length perception and sequence scheduling: An llm-empowered llm inference pipeline. https://doi.org/10.48550/arXiv.2305.13144

Zhou, L., Zhang, L., & Fang, Y. (2020). Logistics service scheduling with manufacturing provider selection in cloud manufacturing. *Robotics and Computer-Integrated Manufacturing, 65*(101), 914. https://doi.org/10.1016/j.rcim.2019.101914