

Explainable AI in Satellite Scheduling: Integrating Large
Language Models, Knowledge Graphs, and Computational
Argumentation for Onboard and Ground Segment Systems

PhD Thesis

Cheyenne Powell

Submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy

Intelligent Computational Engineering Laboratory
Mechanical and Aerospace Engineering Department
University of Strathclyde, Glasgow

February 24, 2026

This thesis is the result of the author's original research. It has been composed by the author and has not been previously submitted for examination, which has led to the award of a degree.

The copyright of this thesis belongs to the author under the terms of the United Kingdom Copyright Acts as qualified by the University of Strathclyde Regulation 3.50. Due acknowledgement must always be made of the use of any material contained in or derived from this thesis.

Acknowledgements

I would like to thank my supervisor, Dr. Annalisa Riccardi, for allowing me to partake in conferences and gain experience and exposure in the research community. Dr. Audrey Berquand, thank you for the support and assistance in working together on papers and providing support during my work experience at ESA. I would also like to acknowledge the support of Dimitris Kardaris from ESA as well as the ESA TEC SW team through the Visiting Researcher program, which provided me with the exposure and experience required to complete this research. Additionally, I would also like to thank the TypeDB team for meeting with me to provide guidance in applying Knowledge Graphs to Satellite Scheduling. My partner, Edward Borton, for supporting and encouraging me throughout my studies. Last, my mum, this work would not have been possible without your support and our discussions about my dreams when I was little. May you continue to sleep in peace.

Abstract

Scheduling satellite operations is crucial for meeting mission requirements and objectives, but as the demand for larger constellations or more advanced onboard or ground-based capabilities grows, traditional management practices are becoming inadequate. To meet emerging needs, there is increasing motivation to automate systems and processes to leverage the latest Artificial Intelligence (AI) technology; however, the nature of space operations necessitates absolute trust in calculations and decisions. The hidden reasoning of AI systems makes it difficult to validate such solutions as satellite scheduling, reducing the opportunity for highly automated and capable systems.

A new field of AI study, eXplainable Artificial Intelligence (XAI), aims at solving this problem by providing automated systems with the ability to communicate explanations to users so insight into the decision-making process is possible. Through integrating with systems and models, XAI techniques assist with validating or correcting system behaviour, which can substantiate trust and grant assurances that training and model configuration are functioning as designed. Various approaches for XAI can generate textual and visual-based explanations, utilising argumentation and Knowledge Graph (KG) concepts to quantify explanations and ensure good interpretability for users.

This research introduces several novel approaches to meeting XAI requirements, establishing model-agnostic benchmarks in explainability for scheduling systems through the integration of Large Language Model (LLM)s, KGs, and Argumentation Framework (AF)s. An example satellite schedule was derived using a Constraint Programming (CP) Solver to facilitate experimentation of targeting and replanning capabilities, automated tabular data extraction from a database, and query generation and answering

Chapter 0. Abstract

to produce explanations for scheduling decisions with all data made available for reproduction. The results from experiments were evaluated for performance in response accuracy, comprehension, and language quality to demonstrate the current capabilities for generating explanations without dedicated model pre-training or extensive optimisation techniques. This thesis concludes with summarising the findings and proposing future research opportunities to leverage the benchmark approaches established here.

Contents

Acknowledgements	ii
Abstract	iii
List of Figures	ix
List of Tables	xiv
List of Acronyms	xix
Nomenclature	xxvi
1 Introduction	1
1.1 Aims and Objectives	5
1.2 Research Contributions	7
1.3 Research Funding	9
1.4 Thesis Structure	9
2 Background	12
2.1 Chapter Overview	12
2.2 Foundations of Planning and Scheduling Systems	12
2.2.1 Concept of Planning	12
2.2.2 Concept of Scheduling	14
2.2.3 Overview of Scheduling Systems	16
2.2.4 Satellite Planning and Scheduling Systems	20
2.3 Advanced Scheduling Techniques	26

Contents

2.3.1	AI Techniques for Satellite Planning and Scheduling	28
2.4	eXplainable Artificial Intelligence	29
2.4.1	A brief overview of Machine Learning	32
2.4.2	Trustworthiness	32
2.4.3	Interpretability	33
2.4.4	Techniques and Applications	35
2.5	Overview of Large Language Models	39
2.5.1	Large Language Models with scheduling	44
2.6	Background Conclusion	45
3	Satellite Scheduling Problem	48
3.1	Chapter Overview	48
3.2	Literature Review	49
3.3	Satellite Schedule problem definition	50
3.3.1	Satellite Schedule Expanded Problem	53
3.3.2	Satellite Schedule Action Recovery	55
3.4	Methodology	57
3.4.1	Heuristic Schedule Creation	58
3.4.2	Solver Schedule Creation	59
3.4.3	Targeted Areas of Interest	59
3.4.4	Rescheduling from a stochastic failure	61
3.5	Results and Discussion	64
3.5.1	First Guess vs Solver	64
3.5.2	Targeted Areas of Interest	69
3.5.3	Rescheduling	70
3.6	Conclusion	75
4	Explainability - Knowledge Graphs	78
4.1	Chapter Overview	78
4.2	Literature Review	79
4.3	Knowledge Graph Schema	80

Contents

4.3.1	Population of the Schema	82
4.4	Results	83
4.4.1	Validating the schema	83
4.4.2	Retrieval of data from the schema	84
4.5	Conclusion	94
5	Explainability - Comparative Analysis and Argumentation	95
5.1	Chapter Overview	95
5.2	Singular Comparative Analysis	96
5.2.1	Literature Review	96
5.2.2	Methodology	97
5.2.3	Results	100
5.3	Argumentation	101
5.3.1	Literature Review	101
5.3.2	Exchange of Properties	104
5.3.3	Tripolar Argumentation for the Expanded satellite schedule . . .	117
5.4	Conclusion	125
6	Large Language Model application to Argumentation	127
6.1	Chapter Overview	127
6.2	Literature Review	128
6.3	Methodology	132
6.3.1	SEP application with the schedule	133
6.3.2	LM selection and hyper-parameters	134
6.3.3	Prompting approach	135
6.3.4	Dataset and Evaluation Metrics	136
6.4	Results	138
6.5	Conclusion	142
7	Large Language Model application to Knowledge Graphs	144
7.1	Chapter Overview	144
7.2	Literature Review	145

Contents

7.3	Methodology	147
7.3.1	Query Creation	149
7.3.2	Code Generation	152
7.3.3	Explanation Generation	152
7.3.4	Assessment Metric	154
7.4	Results	157
7.4.1	Query Creation	157
7.4.2	Query Translation and Code generation	159
7.4.3	Question and Answering over Tabular Data	161
7.4.4	Question and Answering Assessment	165
7.5	Conclusion	174
8	Generalisation of Large Language Model Applications to Scheduling	176
8.1	Chapter Overview	176
8.2	Literature Review	177
8.3	Schedule Benchmark Set	180
8.4	Methodology	182
8.4.1	Query Creation	184
8.4.2	Answering benchmark schedule queries	187
8.4.3	Performance measure	188
8.5	Results	191
8.5.1	Individual Query Categories	194
8.5.2	Cross Category Comparison and Performance	202
8.6	Conclusion	216
9	Conclusion	218
9.1	Chapter Overview	218
9.2	Review of Objectives and Contributions	218
9.3	Limitations	225
9.4	Future Research	226

Contents

Bibliography

229

List of Figures

1.1	AI categories and definitions	2
1.2	A representation of the flow of this thesis.	9
2.1	Optimisation Search Methods	16
2.2	Gantt chart overview	16
2.3	Scheduling Problem classification overview for static and dynamic schedules	18
2.4	Classification of machine-based schedules, including the variations of job shop schedules	19
2.5	scheduling problem classification overview with respect to machine environments	21
2.6	Satellite scheduling process	25
2.7	Onboard activity planning with respect to scheduling	27
2.8	Overview of an onboard decision maker and re-planner between the mother and daughter satellite	30
2.9	Fifty nodes representing the basic and the improved Temporal Conflict Network (TCN)	31
2.10	Taxonomy overview for Interpretability	35
2.11	Growth of LLMs	41
3.1	A reflection on the implications constraints have on a satellite schedule .	54
3.2	Short-term schedule created to recover from the cancellation of action c (yellow) during the execution of the mid-term schedule.	56

List of Figures

3.3	Illustration of image processing cancellation at position a on day n , showing the corresponding field of view. A pink arrow indicates the backtracked position of the satellite at b on day $n - \delta t$. The future predicted opportunity for image retaking is represented by the satellite at position c on the predicted day $n + \delta t$	57
3.4	Schedule generation and solving process overview, where initial actions are optimised through the solver and further calculated for processing data.	60
3.5	Steps required to recover when the next opportunity for an image can be retaken. Red represents changes that have negatively impacted the schedule; blue represents already executed tasks, and green represents the steps for recovery by the system that positively impacted the schedule, creating the short-term schedule.	62
3.6	Action instance comparison between both schedules for Days 1-2	65
3.7	Day 1 Memory allocation comparison between the two methods. The number of images taken, processed and downlinked is on the left with respect to the memory utilised, shown on the right.	68
3.8	Day 2 Memory allocation comparison between the two methods. The number of images taken, processed and downlinked is on the left with respect to the memory utilised, shown on the right.	68
3.9	Schedule snippet representing targets of interest with all actions shaded blue, and environmental factors shaded green, yellow and orange.	70
3.10	Corresponding Memory and Power plot for Figure 3.9	71
3.11	Day 3 before and after the process task removal, and image retaken on the same day. Purple - overall memory, Blue - Images in memory, yellow - Processed images in memory, green - Downlinked images in memory (frequently resets to zero)	76
4.1	KG schema created, displaying an overview of the satellite scheduling problem at time instance i with entities as pink rectangles, relations as orange diamonds, and the attributes as blue ovals.	84

List of Figures

4.2	KG displaying the overlapping of actions with respect to the memory profile taken from typeDB studio for question 2.	87
4.3	KG displaying the detailed environmental factors with the satellite's coordinates for question 3. (Figures <i>b</i> and <i>c</i> are on the next page)	88
4.3	KG displaying the detailed environmental factors with the satellite's coordinates for question 3. (Figure <i>a</i> is on the previous page)	89
4.4	KG showing result options for question 4. (Figure <i>b</i> on the next page) .	89
4.4	KG showing result options for question 4. (Figure <i>a</i> on the previous page)	91
4.5	KG representation for the downlinking action and onboard memory for question 5.	92
5.1	End User Interactions	100
5.2	An overview of the argumentation process across one-time instance representing the occurrence of a single exchange of an action with memory variables V_{ap} , V_{ar} , and V_{ad} within a schedule S to alter the memory m_{i1} at that instance.	108
5.3	An overview of the argumentation process across each time interval representing the values (V_{ap} , V_{ar} , V_{ad} , and V_{ae}) of actions a_p , a_r , a_d , and a_e are exchanged with each other via bi-directional arrows in colours teal and purple, based on the action scheduled at i	109
5.4	Conditions involved with attacks a_p (Image taking), a_r (Image Processing), a_d (Down-linking), and a_e (Idle time) during time i and the conditions affecting '−' and supporting '+' these actions from occurring.	109
5.5	Summary of feasible and infeasible solutions when an action attack occurs on the original schedule for a_p , a_r , a_d , a_e	112
5.6	Results of a_p attack on a_r resulting in an alternate memory m'_i at time instance i at 21 : 04 hrs, followed by the updated memory m'_j exceeding the available memory on-board M_{max} shown as the horizontal green line on the right with all memory profiles referring to the axis on the right and action son the left.	113

List of Figures

5.7	A magnified view of Fig. 5.6 showing where memory at time instance j' breached the maximum on-board memory M_{max} following an attack at instance i	114
5.8	An example showing the implications of pairwise exchange occurring between any two actions within a 70 seconds window of a satellite schedule.	115
5.9	An example showing the implications of pairwise exchange occurring across a 16.58-minute window, highlighting where the 70-second window fits within.	116
5.10	An overview of the argumentation structure for supports and attacks for all actions within the schedule as well as the environmental conditions and impact on all actions	119
5.11	A Tripolar argumentation representation of arguments between 4 main actions (a_{r10} , a_{d4} , a_e , a_{p1}) and environmental conditions at time $i = 208s$.	124
6.1	The use of NLP in the system.	133
6.2	The functionality of SEP applied in 3 scenarios resulting in 3 different memory branches with the initially scheduled actions (a , b and c) on the conditions that the attacks are successful.	134
6.3	Decision flow based on the constraints and environmental conditions for generating prompts from the AA layer to establish an explanation based on the query from the user	135
7.1	Overview of the experiment approach showing LLM use in KG query creation (A), code generation for data extraction (B), and explanation assessment (C), with colour-coded shading.	148
7.2	An example process for prompting an LLM to generate executable code for retrieving data to facilitate an explanatory response to a decision query.	150
7.3	Average Assessment scores for Validation, Bert, Correctness for each query category, including final score without explanation examples provided. S_a is the Swap Actions query category, S_s is Single Substitution, V is Validation, and F_B is Forward and Backward.	169

List of Figures

7.4	Average Assessment scores for Validation, Bert, Correctness for each query category, including final score using the provided explanation example with explanation examples provided. S_a is the Swap Actions query category, S_s is Single Substitution, V is Validation, and F_B is Forward and Backward.	170
7.5	Word count of Human vs GPT-4, with and without explanation prompt examples of the four categories	172
8.1	Illustration of self-augmented prompting	179
8.2	A representation of a benchmark schedule TA13 derived by Henning et al.	183
8.3	Overview of LLM analysis on scheduled data and user queries	186
8.4	Average Correctness across all query categories for all schedules for GPT-4	205
8.5	Average Correctness across all query categories for all schedules for LLaMA-3.1	206
8.6	Average Cosine Similarity across all query categories for GPT-4	208
8.7	Average Cosine Similarity across all query categories for LLaMA-3.1	209
8.8	Average Response Completeness across all query categories for GPT-4	210
8.9	Average Response Completeness across all query categories for LLaMA-3.1	211
8.10	Average Word Count across the all query categories for GPT-4	212
8.11	Average Word Count across all query categories for LLaMA-3.1	213
8.12	Average BertScore F1 across all query categories for GPT-4	214
8.13	Average BertScore F1 across all query categories for LLaMA-3.1	215

List of Tables

2.1	Examples of Model-Agnostic XAI Techniques	36
2.2	Examples of Model-Specific XAI Techniques	37
2.3	Examples of Rule-Based and Symbolic XAI Techniques	37
2.4	Examples of Contrastive Explanation XAI Techniques	38
2.5	Examples of Visualisation XAI Techniques	38
2.6	Examples of Argumentation-Based XAI Techniques	38
2.7	Most popular LLM evaluation metrics by summary category	42
3.1	Overview of simple satellite schedule constraints	51
3.2	Memory profile for actions	52
3.3	Results for Days 1-7	66
3.4	A representation of the day n memory profile and the effect in the database when a processed instance was cancelled from the mid-term schedule during execution at 71 seconds, creating an updated list, deleting the previous instances related to the image.	72
3.5	A representation of day $n - \delta t$ depicting what happens in the database when the located scheduled image has been removed from the mid-term data, represented in the short-term data.	73
3.6	Proposed times for image retake opportunities from the recovery algorithm.	74
3.7	Section of regenerated schedule from the solver due to the predicted coordinates from the recovery algorithm.	75

List of Tables

4.1	Table Representing the scheduled data points mapped to each KG attribute at every time instance.	83
4.2	Example 1 for KG code for types of queries	86
4.3	Example 2 for KG code for types of queries	90
4.4	Tabular results for the execution of codes for questions 1, 2 (Figure 4.2), 3 (Figure 4.3), 4 (Figure 4.4), and 5 (Figure 4.5)	93
5.1	Percentages of action exchange infeasibilities over a period of 1 day . . .	113
5.2	An overview of each action and condition attack required for every action type.	122
5.3	Example calculations of argumentation weights for each action type and the closest probable alternative, except Idle. *Please note, the total of all attacking actions is presented here.	123
6.1	Hyperparameters used for GPT-3	134
6.2	Argumentation Prompt Examples	136
6.3	Scoring scale (end result: normalise average score)	137
6.4	Average scores and standard deviation (subscript) obtained for each evaluation metric and prompting approach.	139
6.5	Examples of explanations generated by the GPT-3 model with the few-shot learning 10 examples approach	139
6.6	Examples of text generated by the GPT-3 model with different learning approaches	140
6.7	Continuation of examples of text generated by the GPT-3 model with different learning approaches.	141
6.8	Average score per user query with the Zero-Shot approach	142
6.9	Average score per user query with the Few-Shot (3 examples) approach	142
6.10	Average score per user query with the Few-Shot (10 examples) approach	142
7.1	Table reporting example queries used in the prompt as a few-shot learning strategy.	151
7.2	Few-shot data sample used for GPT-4 for code generation	153

List of Tables

7.3	Prompts containing constraints for answering questions with tabular results retrieved from the Knowledge Graph with human responses for additional responses	155
7.4	Hyper-parameters used for GPT-4	157
7.5	Table reporting for each query’s category two examples of automatically generated questions	158
7.6	Cosine Similarity between each question for each category and across each category (Standard deviation in subscript)	158
7.7	The average number of iterations for each run for each query category .	159
7.8	Queries with their respective code generated	160
7.9	Tabular results for the query categories	162
7.10	Explanation created by the LLM for Question 1 (Swap Actions) with and without an explanation example	163
7.11	Explanation created by the LLM for Question 2 (Single Substitution) with and without an explanation example	164
7.12	Explanation created by the LLM for Question 3 (Verification) with and without an explanation example	165
7.13	Explanation created by the LLM for Question 4 (Forward and Backward) with and without an explanation example	166
7.14	The mean values and standard deviation scores for all question families with and without example explanation	166
7.15	Assessment of the four exemplary questions for responses in Tables 7.10, 7.11, 7.12, and 7.13	173
8.1	Schedules used with their respective makespans.	182
8.2	Table displaying the unique questions for each of the query categories .	185
8.3	Benchmark answers for each query category	192
8.4	Table displaying the prompts used to answer each of the query categories	193
8.5	Model configuration to generate answers	194
8.6	Average results for the Swap category 15jx15m all schedules for GPT-4 .	196
8.7	Average results for the Swap category 15jx15m all schedules for LLaMA-3.1	196

List of Tables

8.8	Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for GPT-4	197
8.9	Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1	197
8.10	Average results for the Increase category 15jx15m all schedules for GPT-4	199
8.11	Average results for the Increase category 15jx15m all schedules for LLaMA-3.1	199
8.12	Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for GPT-4	200
8.13	Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1	200
8.14	Average results for the Decrease category 15jx15m all schedules for GPT-4	202
8.15	Average results for the Decrease category 15jx15m all schedules for LLaMA-3.1	202
8.16	Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for GPT 4	203
8.17	Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1	203
8.18	Average results across all categories for each schedule (GPT-4)	203
8.19	Average results across all categories for each schedule (LLaMA-3.1)	203
8.20	Average results for Optimal and Non-Optimal Schedules (GPT-4)	204
8.21	Average results for Optimal and Non-Optimal Schedules (LLaMA-3.1)	204
8.22	Average results for all categories across all schedules (GPT-4)	204
8.23	Average results for all categories across all schedules (LLaMA-3.1)	205

List of Acronyms

AA Abstract Argumentation.

AAF Abstract Argumentation Framework.

ACO Ant Colony Optimization.

ADF Abstract Dialectical Framework.

AF Argumentation Framework.

AFs Argumentation Frameworks.

AI Artificial Intelligence.

ALE Accumulated Local Effects.

ALOS Advanced Land Observing Satellite.

AM Additive Manufacturing.

ANN Artificial Neural Network.

APS Automated Planning and Scheduling.

ASPEN Automated Scheduling/Planning ENvironment.

BA Bipolar Argumentation.

BAF Bipolar Argumentation Framework.

BERT Bidirectional Encoder Representations from Transformers.

BLEU Bilingual Evaluation Understudy.

List of Acronyms

CA Computational Argumentation.

CBR Case-Based Reasoning.

CF Counterfactual.

CHIP Constraint Handling In Prolog.

CLP Constraint Logic Programming.

CMs Computational Models.

CNN Convolution Neural Networks.

CoT Chain-of-Thought.

CP Constraint Programming.

CSA Canadian Space Agency.

DeepLIFT Deep Learning Important FeaTures.

DJSSP Distributed Job Shop Scheduling Problem.

DL Deep Learning.

DMS Decision Making Systems.

DPFSP Distributed Permutation Flow Shop Scheduling Problem.

DS Daughter Satellites.

DSCOVR Deep Space Climate Observatory.

DSS Distributed Satellite Systems.

EDA Exploratory Data Analysis.

EO Earth Observation.

ESA European Space Agency.

List of Acronyms

EVA ESA Virtual Assistant.

FBO Feasible Better Objective.

FLAN Finetuned Language Net.

FS Final Score.

FSL Few-Shot Learning.

FWO Feasible Worse Objective.

GA Genetic Algorithm.

GEO Geostationary Orbit.

GLaM Generalist Language Model.

GLORIA Geostationary/Low-Earth Orbiting Radar Image Acquisition System.

GNMT Google Neural Machine Translation.

GNSS Global Navigation Satellite System.

GPT Generative Pre-trained Transformer.

GPT-3 Generative Pre-trained Transformer 3.

Grad-CAM Gradient-weighted Class Activation Mapping.

GRASP Greedy Randomized Adaptive Search Procedure.

GSO Ground Station Operators.

HDM-GA Hybrid Dynamic Mutation Genetic Algorithm.

HEO High Earth Orbit.

HGA Hybrid Genetic Algorithms.

HILML Human-In-The-Loop Machine Learning.

List of Acronyms

IAC International Astronautical Congress.

iAF Incomplete Argumentation Framework.

ICE Individual Conditional Expectation.

ISS International Space Station.

IT Information Technology.

JAXA Japan Aerospace Exploration Agency.

KG Knowledge Graph.

KSPDG Kerbal Space Program Differential Games.

LEO Low Earth Orbit.

LIME Local Interpretable Model-agnostic Explanations.

LLaMA Large Language Model Meta AI.

LLaVA Large Language and Vision Assistant.

LLM Large Language Model.

LLMs Large Language Models.

LM Language Model.

LMs Language Models.

LORE Local Rule-Based Explanation.

LRP Layer-wise Relevance Propagation.

LSTM Long Short-Term Memory.

MCA Multiple Comparative Analysis.

MCTS Monte Carlo Tree Search.

List of Acronyms

MEO Medium Earth Orbit.

METEOR Metric for Evaluation of Translation with Explicit Ordering.

MILP Mixed Integer Linear Programming.

ML Machine Learning.

MNIFSP Mixed No-Idle Flow-shop Scheduling Problem.

MS Mother Satellite.

MSG Meteosat Second Generation.

MSODSP Multi-Satellite Observation and Data-downlink Scheduling Problem.

MSS Multi-Satellites System.

MT Machine Translation.

NASA National Aeronautics and Space Administration.

NLG Natural Language Generation.

NLP Natural Language Processing.

NLU Natural Language Understanding.

NMT Neural Machine Translation.

NN Neural Network.

NSGA-II Non-Dominated Sorting Genetic Algorithm II.

OV Objective Value.

PaLM Pathways Language Model.

PBAF Probabilistic Bipolar Argumentation Framework.

PDPs Partial Dependence Plots.

List of Acronyms

PEP Pairwise Exchange Property.

PPO Proximal Policy Optimization.

PS Predictive Solving.

PSP Parker Solar Probe.

QBAF Quantitative Bipolar Argumentation Framework.

RAG Retrieval-Augmented Generation.

RAX Remote Agent Experiment.

RCA Root Cause Analysis.

REMPS Robust and Explainable Mission Planning and Scheduling.

RHO Rolling-Horizon Optimization.

RL Reinforcement Learning.

RNN Recurrent Neural Network.

RNNs Recurrent Neural Networks.

ROUGE Recall-Oriented Understudy for Gisting Evaluation.

RS Recommender System.

SAF Structured Argumentation Framework.

SBERT Sentence-BERT.

SCA Singular Comparative Analysis.

SCMs Structural Causal Models.

SCSP Satellite Constellation Scheduling Problem.

SEP Single Exchange Property.

SGTS Satellite-Ground Time Synchronization.

SHAP SHapley Additive exPlanations.

SNE Stochastic Neighbour Embedding.

SP Scheduling Problem.

STK Satellite Tool Kit.

SVCCA Singular Vector Canonical Correlation Analysis.

SVM Supported Vector Machines.

t-SNE t-Distributed Stochastic Neighbour Embedding.

TA Tripolar Argumentation.

TAF Tripolar Argumentation Framework.

TCN Temporal Conflict Network.

TED Teaching Explanations for Decisions.

TT&C Track, Telemetry & Control.

UMAP Uniform Manifold Approximation and Projection.

VLM Vision-Language Model.

XAI eXplainable Artificial Intelligence.

XRL eXplainable Reinforcement Learning.

XUI eXplanation User Interface.

ZSL Zero-Shot Learning.

Nomenclature

A	All actions
a	Any action
a'	Alternate action
a_d	Image downlinking action
a_e	Idle instances
a_p	Image taking action
a_r	Image processing action
An_{mem}	Analysed images in memory
$Args_F$	Argument F
D^+	Positive Fixed Decision
D^-	Negative Fixed Decision
d_i	Total number of images downlinked
D_m	Memory released to downlink processed images
G	Mission's Goal
GS	Ground Station
i	Any time instance or interval

Nomenclature

I_{mem}	Images in memory
I_m	Memory required to store an image
j	Secondary time instance or interval
LR	Light range exposure
M	Onboard Memory
m_i	Total memory used at any instance
m'_i	Alternate memory at any instance i
m'_j	Alternate memory at any secondary instance j
M_{max}	Maximum available memory onboard
n	Number
P	Onboard Power
p_i	Total number of remaining images taken in memory
PR	Probability
r_i	Total number of remaining processed images in memory
R_m	Memory utilised at every time instance during the processing of an image
R_{num}	The number of processes instances
S	Schedule
T	Time horizon of a schedule
t	Time
t_{a-dur}	Action duration
Tot_a	The total number of actions

Nomenclature

v^*	The smallest value of memory value
v_a	Value of memory required by any action a
$v_{a'}$	Value of memory required by any alternate action a'
v_{ad}	Value of memory required by action a_d
v_{ae}	Value of memory required by action a_e
v_{ap}	Value of memory required by action a_p
v_{ar}	Value of memory required by action a_r
VT	Visibility of Target
VT_n	Specific Visibility of Target n
W_M	Weight associated with the onboard memory M
W_{An_mem}	Weight relation for Analysed Images in Memory
W_{G_ae}	Weight relation between idle a_e and the mission's Goal G
W_{G_a}	Weight relation between the goal G and any action based on the specified requirements
W_{GS}	Weight of Ground Station accessibility
W_{I_mem}	Weight relation for Images in Memory
W_{LR}	Weight relation for Light Range exposure
W_{P_ae}	Weight relation between Power P and idle a_e
W_{P_a}	Weight relation between the power level P and the scheduled action
W_{VT_n}	Weight governing the visibility of a specific target n
X	Binary decision matrix
$X_{i,a}$	Binary decision matrix for action a scheduled any time i

Nomenclature

Y Intervals where a_p may occur

Chapter 1

Introduction

Do you trust Artificial Intelligence (AI)? A question that is frequently asked as the era of AI becomes increasingly pervasive. With the expansion of technology in our daily lives, AI is seamlessly being integrated, from autonomous vehicles [1] and medical services [2], to personalised recommendations [3], and smart home assistance [4]. The notion of trust in AI transcends mere technological expertise, and in the domain of satellite scheduling, this question couldn't be more applicable where the reliability and efficiency of AI-driven decisions can directly impact the mission critical devices we might no longer physically come in contact with after launch; missions that may influence decisions impacting human life, such as natural disaster response.

In order to establish trust, you must first understand what AI is and how it makes decisions. However, the definition of AI has been subject to much debate and contention, with one current modern definition stated as:

Artificial Intelligence is the study of how to make computers do things at which, at the moment, people are better [5].

This definition lacks technical specificity and excludes the beyond-human capabilities that have been demonstrated presently. By contrast, and in proof of the multifaceted nature of the field of AI, an alternative modern definition divides AI into four related but distinct categories, namely *Thinking Humanly*, *Thinking Rationally*, *Acting Humanly*, and *Acting Rationally*, as shown in Figure 1.1 [6].

While these definitions may still be lacking in specificity, combined, they cover the

<p>Thinking Humanly</p> <p>“The exciting new effort to make computers think . . . <i>machines with minds</i>, in the full and literal sense.” (Haugeland, 1985)</p> <p>“[The automation of] activities that we associate with human thinking, activities such as decision-making, problem solving, learning . . .” (Bellman, 1978)</p>	<p>Thinking Rationally</p> <p>“The study of mental faculties through the use of computational models.” (Charniak and McDermott, 1985)</p> <p>“The study of the computations that make it possible to perceive, reason, and act.” (Winston, 1992)</p>
<p>Acting Humanly</p> <p>“The art of creating machines that perform functions that require intelligence when performed by people.” (Kurzweil, 1990)</p> <p>“The study of how to make computers do things at which, at the moment, people are better.” (Rich and Knight, 1991)</p>	<p>Acting Rationally</p> <p>“Computational Intelligence is the study of the design of intelligent agents.” (Poole <i>et al.</i>, 1998)</p> <p>“AI . . . is concerned with intelligent behavior in artifacts.” (Nilsson, 1998)</p>

Figure 1.1: Different categories and definitions of AI [6]

near totality of the field of AI. The mere fact that no single definition for AI as a means of research or system development can be agreed upon speaks volumes to the depth and breadth of capabilities of AI systems and the variety of solutions it can offer.

The recent rapid growth of AI technologies could suggest that AI as a whole is still in the early stages of research, yet the formal concept of AI emerged in the mid-20th century. The term AI was coined by John McCarthy [7], rooted in the groundwork established by E. Berkley [8], A. Turing [9] and A.L. Samuel [10] in the years previous.

E. Berkley, in 1949, endeavoured to expose the revolutionary work on creating ‘mechanical brains’ to the general public, stating their importance in facilitating such technologies as an automatic address book, an automatic library, an automatic stenographer, and an automatic recogniser [8]. These visions of future applications of automated systems may seem quaint and archaic by modern standards, but these aspirations were what drove the creation of modern-day computers and smartphones.

For A. Turing, arguably the best-known name associated with the field of AI, his publication, *Computer Machinery and Intelligence*, Turing introduced the ‘Imitation

Game’, which was later simply referred to as the Turing Test [9]. The outline of the test is to have a series of questions presented by an independent judge to two entities, the identity of which is unknown to the judge. One entity will be a human, the other will be an AI system, with the goal of the AI system to mislead the judge into believing their responses are from a human. The test served as a benchmark for evaluating the performance of AI systems for decades, but in recent years, it has seen reduced relevance in research and systems evaluation [11]. Even so, no AI system is yet to score perfectly in these tests, and the methodology presented by Turing fueled the invention of many new ways to assess the performance of AI systems, such as accuracy, F1 score, and error rate to name but a few [12].

Shortly after A. Turing’s publication, A.L. Samuel [13] published his work on designing Machine Learning (ML), the first to introduce this term, procedures to enable a system to learn the game of checkers. Through iterative steps, the study demonstrates that the system is able to consistently outperform the average person in the game, winning more than it would lose, though A.L. Samuel does state that the scale of the experiment is limited. Regardless, the paper theorised the future capabilities of the principles established within the study, which was later developed into what is now known as alpha-beta pruning. A.L. Samuel’s studies were foundational for the more recent developments in ML, allowing for the formulation of Reinforcement Learning (RL) and paving the way for modern AI techniques and capabilities [14]

The history of AI and humans’ investigation into our solar system and space both began in the 1950s, with our first exit from Earth’s atmosphere achieved in 1961 [15]. Naturally, it was some years later before the two fields combined, as both underwent a significant journey of maturity and technological advancement. The first AI system introduced for a space mission was Remote Agent Experiment (RAX) for the Deep Space 1 mission, in 1999 [16]. This mission focused on testing the system’s capabilities in flight management through tests over multiple days, where the system would be required to complete such tasks as navigating the craft, taking images of a specific location and responding to an unexpected component failure. The results of the experiment proved RAX successful in every objective and outlined the potential for future

autonomous systems being utilised in spacecraft and missions [17].

However, despite these initial encouraging results, wide adoption of AI systems and solutions for spacecraft and missions has not been achieved. Further research into applications within the space industry have explored the capabilities across satellite and ground operations, including GPS systems [18], Low Earth Orbit (LEO) networks [19], onboard decision making systems [20], and distributed satellite systems [21], yet these studies highlight the limited scope for real-world implementation that is currently present.

One of the critical factors that is influencing hesitation in broadly adopting AI systems into space missions and operations is the lack of trust that operators and organisational leaders have in such systems. As AI systems are not designed to explain the reasoning used in determining a decision, being able to identify why a system is behaving or performing in a particular way has been a challenge. Therefore, the adoption of such systems into an environment with high cost (both financial and human-life) is slow and cautious, with further research into how systems derive their decisions being required to enable trust in AI solutions [22], [23], [24].

The need to better understand the inner workings of AI has driven forward a new area of research, eXplainable Artificial Intelligence (XAI), which focuses on enabling autonomous systems with the ability to provide clear explanations to why a decision was made and what influenced the decision [25]. The experience of the human operators and their ability to comprehend responses from an autonomous system, typically provided through the utilisation of Natural Language Processing (NLP), is crucial to fulfilling the goal that XAI principles aim to deliver [26]; and as AI systems and models become more complex and capable in making and executing decisions, the need for transparent and interpretable interactions becomes ever greater, with XAI aiming to enable access into the reasoning of the system [27].

Hence, XAI studies and experiments are seen as critical in facilitating the broad adoption of autonomous systems in a wide variety of industries, such as education, transport, manufacturing, and business operations [28], [29]. Yet, the area of greatest need for XAI are also the most sensitive and challenging, where life and livelihood is at

stake, namely healthcare [30], finance [31], autonomous vehicles [32], and satellite operations [23], [33], [34]. The investigations into the real-world potential for autonomous systems demonstrate keen interest in making the most of the capabilities of AI models, but developing trust in how the systems determine their decisions requires XAI to make it possible and, therefore, must become a fundamental part of system design and configuration.

So, do you trust AI? Lee et al. defined trust in automation as *the attitude that an agent will help achieve an individual's goals in a situation characterized by uncertainty and vulnerability* [35], however, while Kohn et al. agree with the definition they also affirm that trust is often earned over time and not something that is granted immediately and freely, so it is relatable that the same would apply to non-human entities as well [36]. Given the relatively sudden emergence of new and highly capable AI systems, it seems only natural not to trust the system, as not enough time and evidence exist to provide assurance that the system is trustworthy. XAI opens the door and enables dialogue with a system, which grows the evidence that a system can (or cannot) be trusted; because, how can you trust what you do not understand? As AI and XAI models mature over time, coupled with the growth of interactions and understanding, trust, even if fragile, feels inevitable, which sets the foundations for the implementation of autonomous systems into an ever-increasing number of applications.

1.1 Aims and Objectives

The aim of this thesis is to focus on the development and applications of XAI concepts for Robust and Explainable Mission Planning and Scheduling (REMPS) with a particular focus on the intricacies of satellite planning and scheduling to enhance transparency and build trust through human interpretability and interactivity in using AI implemented schedulers. This research is applicable to any kind of schedule and does not focus on the robustness of the scheduling, meaning the schedules capacity to react and handle unplanned changes, but on the XAI aspects of an automated system, with the goal of enabling decision insight and developing trust in system-made decisions.

Chapter 1. Introduction

To achieve this, the lead research question is: *How can XAI be utilised to enhance human trust, interpretability, and interactivity in autonomous satellite planning and scheduling systems?*

This question is broken into three component parts to best address the issues raised:

- *Q1: What is required to establish trust in autonomous decision-making scheduling systems?*
- *Q2: What techniques exist to facilitate XAI interactions with human operators?*
- *Q3: Can XAI deliver insight into decisions made by an autonomous scheduling system?*

This thesis aims to answer the above questions by delivering on these five objectives:

- Establish a formulation of a simplified Earth Observation (EO) satellite scheduling problem by analysing the constraints of benchmark schedules and satellite scheduling concepts, covering both ground-based and onboard scheduling scenarios.
- Develop innovative theoretical and computational frameworks, combining mathematical modeling with simulation algorithms, that leverage state-of-the-art XAI techniques for evaluating and interpreting benchmark and satellite schedules.
- Formulate a novel computational approach to compare and explain the rationale behind scheduling decisions through in-depth conflict analysis.
- Create an original dataset of question-and-answer templates tailored to benchmark and the derived satellite schedules, enabling a system to respond to operator natural language queries with contextually relevant and explainable information.
- Propose a novel interactive methodology focused on explainability for both benchmark and derived satellite schedules to provide interpretable insight for scheduling decisions.

1.2 Research Contributions

This thesis is comprised of recent publications through journals and conference proceedings:

1. 2021, ‘*Towards explainability of on-board satellite scheduling for end user interactions*’, Paper presented at 72nd International Astronautical Congress (IAC) [37]. This paper introduces the application of Constraint Programming (CP) techniques applied to satellite scheduling. It is the foundation for this research where XAI techniques will be applied. It also introduces Singular Comparative Analysis (SCA) and argumentation, where computational approaches are taken to detect the differences between schedules. The details of this paper are discussed in Chapters 3 and 5.
2. 2022, ‘*On-board re-planning of an earth observation satellite for maximisation of observation campaign goals*’, Paper presented at 73rd IAC [38]. This paper discusses the recovery procedures when an action stochastically fails, leading to recalculating the next opportunities for the failed actions while considering constraints and rescheduling opportunities. The details of this paper are introduced and discussed in Chapter 3.3.2 and applied in Chapter 6.
3. 2023, ‘*Abstract argumentation for explainable satellite scheduling*’, Paper presented at 2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA) [Peer-reviewed] [39]. Talks about the application of Abstract Argumentation (AA) concepts with satellite scheduling, using two techniques Single Exchange Property (SEP) and Pairwise Exchange Property (PEP), where visual representations are produced in developing XAI concepts. This paper is used in two chapters, Chapters 5 and 6.
4. 2023, ‘*Natural language processing for explainable satellite scheduling*’, Paper presented at SPACEOPS [40]. Introduces the application of Large Language Model (LLM)s with SEP of AA for satellite scheduling and is discussed in Chapter 6.

Chapter 1. Introduction

5. 2023, ‘*Question answering over knowledge graphs for explainable satellite scheduling*’, Paper presented at 74th IAC [41]. This paper expands on the use case of Knowledge Graph (KG)s with satellite scheduling but also investigates applying LLM within an automated system to retrieve queries from end users, create code and provide answers to the end user (operator). This paper received the best interactive presentation award for the Applications and Operators category at IAC. It is discussed in Chapters 4 and 7.
6. 2024, ‘*Explaining AI Decisions in Autonomous Satellite Scheduling via Computational Argumentation*’, Paper presented at SPAICE [Peer-reviewed] [42]. This paper introduces the concept of applying Tripolar Argumentation Framework (TAF) to a satellite schedule via a technique influenced by a decision-making algorithm briefly discussed in Chapter 3, where conflicts between influencing factors are assessed. This is explained in Chapter 5.3.3
7. 2025, ‘*Question answering over knowledge graphs for explainable satellite scheduling*’, an extended [Peer-reviewed] journal paper published in the Journal of Aerospace Information Systems in the American Institute of Aeronautics and Astronautics society [43]. This paper contains a very detailed analysis of exploring KGs with LLMs that is also used in chapters 4 and 7.
8. 2025, ‘*Generating Textual Explanations for Scheduling Systems Leveraging the Reasoning Capabilities of Large Language Models*’, Paper published in the [Peer-reviewed] Journal of Intelligent Information Systems: Integrating Artificial Intelligence and Database Technologies [44], chosen due to the field-agnostic nature of this study. This paper delves into investigating the capabilities of two different LLMs in and assessing benchmark schedules in tabular form, and answering queries accordingly. This is discussed in Chapter 8.

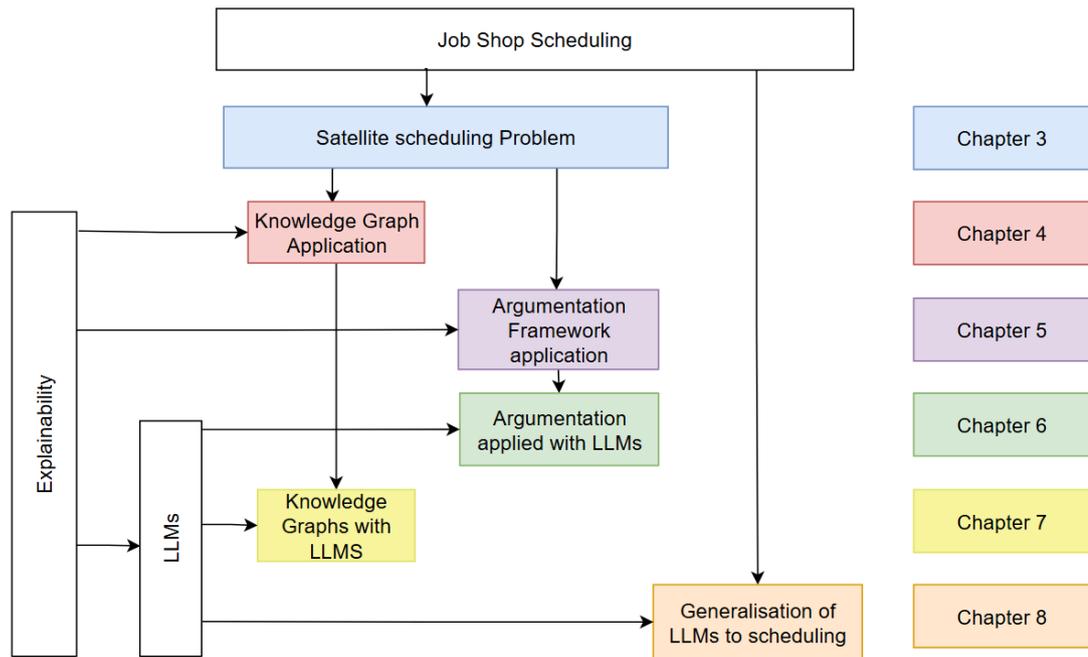


Figure 1.2: A representation of the flow of this thesis.

1.3 Research Funding

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; and half funded by the Strathclyde Center for Doctoral Training in Explainable AI for industrial decision support.

1.4 Thesis Structure

The layout of this Thesis is shown in Figure 1.2, with an overview describing each chapter, which, including this chapter, is as follows:

- Chapter 1 introduces the purpose of this thesis, outlining the research questions, objectives and published contributions by the author.
- Chapter 2 provides a background of all the necessary knowledge required to achieve a baseline understanding of planning and scheduling concepts for scheduling systems, including satellite scheduling systems, AI scheduling techniques,

Chapter 1. Introduction

LLM applications, followed by XAI techniques and their applications.

- Chapter 3 discusses the simple EO satellite scheduling problem that is used throughout this thesis. This chapter introduces the approach to solving the problem via a CP technique based on the defined constraints considered for the problem and a sample solution. This chapter also investigates how the system may recover due to a stochastic failure introduced in the schedule during a mission.
- Chapter 4 introduces the application of KG with satellite scheduling, detailing how the KG schemas for the satellite schedules may be derived and examples of how queries may be written to retrieve data and interconnected properties between the factors used within the scheduling problem.
- Chapter 5 introduces SCA and Argumentation concepts and how they were applied to the satellite scheduling problem via a computational approach. Initially, SCA is assessed in Chapter 5.2, followed by the application of argumentation in Chapter 5.3, broken into two subchapters, Chapter 5.3.2 utilising two argumentation techniques SEP and PEP. While the concept of TAF introduced in Chapter 5.3.3 is applied to a RL derived schedule to assess its capabilities in generating data required for providing explanations, aligned with Contribution 6.
- Chapter 6 combines the concept of AA utilising SEP with LLMs to investigate how they may be integrated into a system with queries received, followed by the analysis of the extraction of data from the AA layer to provide answers to the end user.
- Chapter 7 utilises LLMs with KGs for exploring the abilities of LLMs within an automated developed system for this research to understand user queries to create code for the extraction of data from the KG to provide relevant explanations to the user.
- Chapter 8 talks about an in-depth analysis of the capabilities of applying two different LLMs to scheduling, using 10 benchmark schedules for extracting tabular data, understanding queries and providing explanations to the end user.

Chapter 1. Introduction

- Lastly, Chapter 9 provides the concluding remarks of this thesis, summarising all the findings of the research aims and objectives and proposals for future work where this research may be enhanced and used in more complex real-time systems.

Chapter 2

Background

2.1 Chapter Overview

This chapter contains the foundation of the concepts required for achieving XAI for satellite scheduling. This will entail the basic requirements for scheduling concepts and methodologies, that can be applied to satellite scheduling. The concepts of AI and their applications to scheduling will be addressed, as well as explainability and how it may be applied to scheduling.

2.2 Foundations of Planning and Scheduling Systems

2.2.1 Concept of Planning

The difference in function that planning and scheduling perform can be difficult to discern, with some studies and industries having blurred or mixed lines between each concept [45, 46]. However, in simple terms, within this thesis, planning is considered responsible for determining *What* needs to be done, while scheduling determines *When* the activities should be completed [47, 48].

Exploring the concept of planning first, a singular definition for the practice of planning has proven to be a challenge for researchers and business leaders over the years [49]. In 1981, Henry Mintzberg, who is still considered a leading voice in business management today, highlighted this very issue and asserted that a single definition

Chapter 2. Background

needed much more investigation, and in the meantime, proposed four conceptual definitions for planning instead, each intended to be an improvement over the previous. These were:

- Planning as future thinking - considering the future in allocating resources, building relationships, and assessing and actioning situations
- Planning as integrated decision-making - a conscious attempt to integrate decisions across different areas
- Planning as formalised procedure and articulated result - formalise procedures to make and integrate decisions and articulate the result.
- Planning as programming - performing activities for the future on instinctual reasoning

Mintzberg outlined the vagaries in these definitions, considering each and collectively inadequate as a formal definition. The planning thought leaders joined the open discussion, with Synder responding on the *need for evaluation* in planning [50], but it was not until 1994 that Mintzberg felt satisfied with a complete definition, presenting the following in his seminal book ‘The Rise and Fall of Strategic Planning’:

Planning is a formalised procedure to produce an articulated result in the form of an integrated system of decisions.

By Mintzberg’s intent, this definition captures the core essence of planning as an action and a thought process. However, in the author’s opinion, this definition lacks functional directness and therefore finds Harold Kerzner’s description of planning much more satisfactory and actionable [51]. Kerzner wrote plainly in the 2007 publication ‘Project Management: A Systems Approach to Planning, Scheduling, and Controlling’:

Planning is the function of selecting objectives and establishing the policies, procedures, and programs necessary for achieving them.

This succinctly captures the intent and purpose of planning more directly, aligning better with the process required before scheduling can be completed. Considering the separation of planning and scheduling activities, W.A. Moylan aligned with the

above definition of planning, following that scheduling is the conversion of plans into an executable timetable of activities [52].

2.2.2 Concept of Scheduling

Across many domains, scheduling systems are critical in ensuring tasks are completed within an allotted time, efficiently utilising the limited available resources for specified jobs by optimising one or more objectives [53]. These domains range from manufacturing and production [54,55], which is the main focus of this paper, to transportation [56] and healthcare [57], to space operations [58], each entailing a series of tasks for optimisation of costs and other objectives. As the number of tasks increases, so do the constraints and resources, increasing the complexities of creating optimal and near-optimal solutions.

Within manufacturing, scheduling problems usually consist of a set of machines with a definite set of tasks, also known as jobs, to be executed on these machines. Each job contains a start, processing, and end time, including times required for setting up the job and the weight of the job with respect to others [53]. The conditions required to schedule certain jobs may depend on the industry in which the schedule is based, with these conditions also known as constraints. Constraints vary based on job precedence, resource requirements, routing, and other conditions [53, 59]. In solving constraint-based schedules, all conditions must be satisfied for all tasks on every machine, equating to a feasible solution where an objective function is sometimes defined to minimise or maximise the function [60].

There are several approaches towards achieving an optimal solution, one of which is a heuristic approach, derived from the Greek word *heuriskein*, meaning to discover and explore; in modern language refers to a method that produces a feasible but not necessarily an optimal solution within a timely manner for specific problem. The method is frequently used to solve combinatorial problems such as the travelling salesman problem, which involves route finding to ensure all destinations have been visited and returned to the starting point while keeping the distance travelled to a minimum. This approach can be achieved in several ways, such as local search, branch and bound

Chapter 2. Background

search, to name a few [61, 62].

Expanding on the problem specific approach in finding optimal scheduling solutions, a meta-heuristic approach is flexible and can be applied to any problem, which is effective for solving flexible job shop scheduling problems that are usually large-scale manufacturing systems [63–65]. These can be solved in several different ways, including but not limited to tabu search, neighbourhood search, simulated annealing, Greedy Randomized Adaptive Search Procedure (GRASP), repeated random search, and Ant Colony Optimization (ACO) [61, 62, 66].

For simulated annealing, this approach uses a stochastic algorithm that searches the solution space with a stochastic hill-climbing technique [67]. A tabu search approach uses three distinct strategies for managing the search: a forbidding strategy, which controls what is considered in the search; a freeing strategy, which releases elements from the search; and a short-term strategy, which manages the relationship between the other strategies. In more advanced implementations, a learning strategy can also be added to build iterative improvement in longer searches [68]. In neighbourhood searching, the approach targets the nearest neighbour nodes or entities to search for the solution [61]. With GRASP a construction algorithm is used whereby a new solution is built one element at a time, utilising a selection rule focused on the best additions to the solution [61]. For ACO the approach adapts the naturally observed path-finding abilities of ants by developing a series of agents that function like ants. The agents build solutions using probabilistic greedy algorithms, with probabilities of two components: a greedy factor which evaluates the short-term cost of a decision, and a trail factor that measures the quality of solutions which included that decision before [61].

In seeking optimal solutions heuristic and meta-heuristic approaches can be either exhaustive/enumerative, to find an exact solution, or random to find a near-optimal or approximate solution, as shown in Figure 2.1. Xhafa et al. discussed the classification highlighting the decision required between choosing a random, inexact but efficient approach and a costly yet definitive one [69].

When scheduling problems are solved, they are often represented in Gantt charts, which are generally shown in the form of machines or people with scheduled jobs or

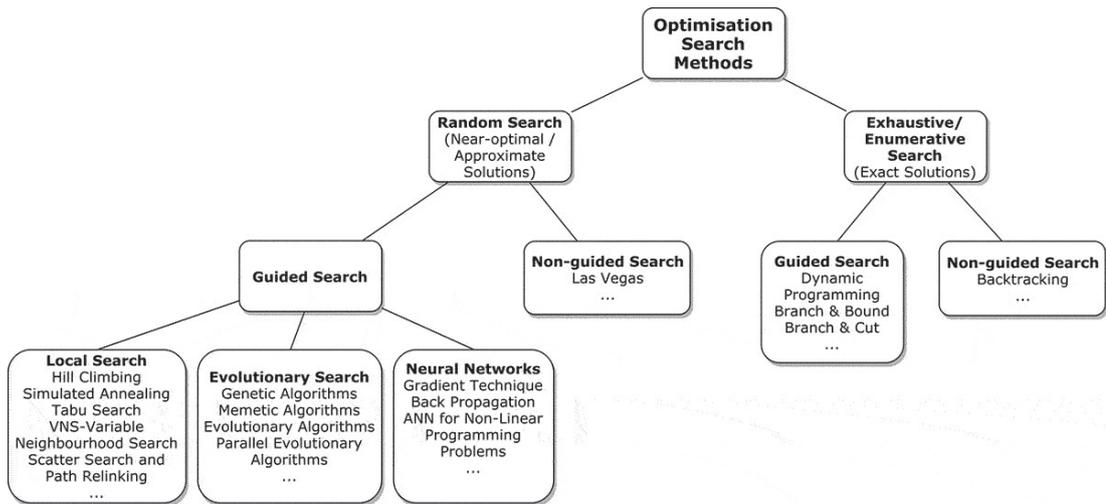


Figure 2.1: Heuristic and Meta-heuristic optimisation search methods [69]

jobs with assigned machines or people [70]. Figure 2.2 represents two ways in which Gantt charts may be shown for schedules.

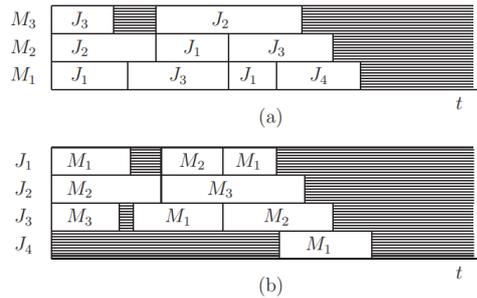


Figure 2.2: Example of Gantt chart output for completed schedule where (a) is from the perspective of M (Machine), and (b) is from the perspective of J Job [70]

Every schedule, however, has its own specific classification upon which a schedule is assessed and solved.

2.2.3 Overview of Scheduling Systems

The classifications of scheduling systems are divided into two categories: deterministic and non-deterministic, based on the problem's criteria [71].

Non-deterministic scheduling is where an optimal solution in polynomial time is not guaranteed and is undetermined. Heuristic and meta-heuristic methods are often asso-

ciated with this term, where a good enough solution is sought rather than a confirmed optimal solution. The theoretical concept of non-deterministic polynomial time is also referred to as NP [72, 73].

Uncertain scheduling, a type of non-deterministic scheduling, occurs when there is incomplete data or unexpected changes occurring within the schedule, which may result in partial or non-quantifiable information [72]. When these unexpected changes (uncertainties) occur due to randomisation exposed to one or more parameters in various conditions, such as machine breakdowns, processing times, and delays in the start and end times of jobs, this also results in stochastic scheduling [74].

Deterministic scheduling, by contrast, is defined by having no unknown or probabilistic variables, even where certain variables may be dynamic [71]. While deterministic scheduling is performed in a wide variety of methods, such as round robin scheduling [75], critical path methodology [76], assembly line scheduling [77], and employee shift scheduling [78], by far the most prevalent problem is job shop scheduling [79, 80] (although job shop scheduling itself is not always deterministic). The key characteristics of job shop scheduling are the utilisation of either single or multiple machines with a set of assigned jobs to be processed and the optimisation of the performance measures, where computational resources are not a limiting factor [81–83]. Additionally, all performing operations within the system are fixed, and a specific outcome is guaranteed [72].

One of the most influential studies into job-focused scheduling was published by E. Taillard in 1992 [84], establishing 260 benchmark scheduling problems using heuristic methods based on tabu search techniques to generate solutions. With the aim of computing a lower-bound makespan, they derived solutions for flow-shop, job-shop and open-shop schedules.

In 1995, Nagar et al. [82] developed a classification scheme for multiple and bi-criteria scheduling for a deterministic approach, highlighting the shop configuration for single and multiple machines and the applied solution methodology. They discovered conventional techniques were not successful, and therefore meta-heuristic approaches would need to be considered [62, 82].

Nagar et al. [82] classified job-shop scheduling as deterministic only when all parameters are known with certainty. In any other case the problem is considered stochastic. The classification was later broadened by Lin et al, as seen in Figure 2.3, adding the concepts of static (where no parameters change during the schedule) and dynamic (where parameters can change during the schedule) scheduling problems.

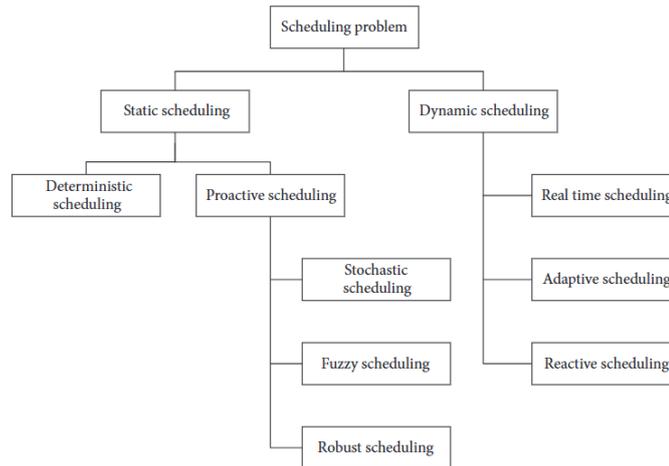


Figure 2.3: Schedule classification overview [85]

The classification of scheduling problems presented by Lin et al helped Amjad et al. to classify the machine based scheduling problems, such as the variations of job shop and flow shop, as shown in Figure 2.4.

Flow-shop scheduling is the serial arrangement of jobs with respect to machines for production flow with the aim of satisfying the criteria by finding the sequence of jobs [74, 87, 88]. Additionally, Assembly Flow-shop scheduling is an expansion of Flow-shop scheduling into two components: jobs during manufacturing can have more components in the machining stage, and jobs combined during the assembly stage [87].

Job-shop scheduling is more complex than flow-shop scheduling and is the assignment of a set of jobs, each with its own uninterruptible operation sequence, are assigned to specific machines for production [84, 89]. They are of high complexity due to their NP-hard combinatorial nature. In addition to this, flexible job-shop scheduling evolved when operations sequences are fixed but may be assigned to any machine that does not already have the job assigned, adding more challenges to solving the problem and

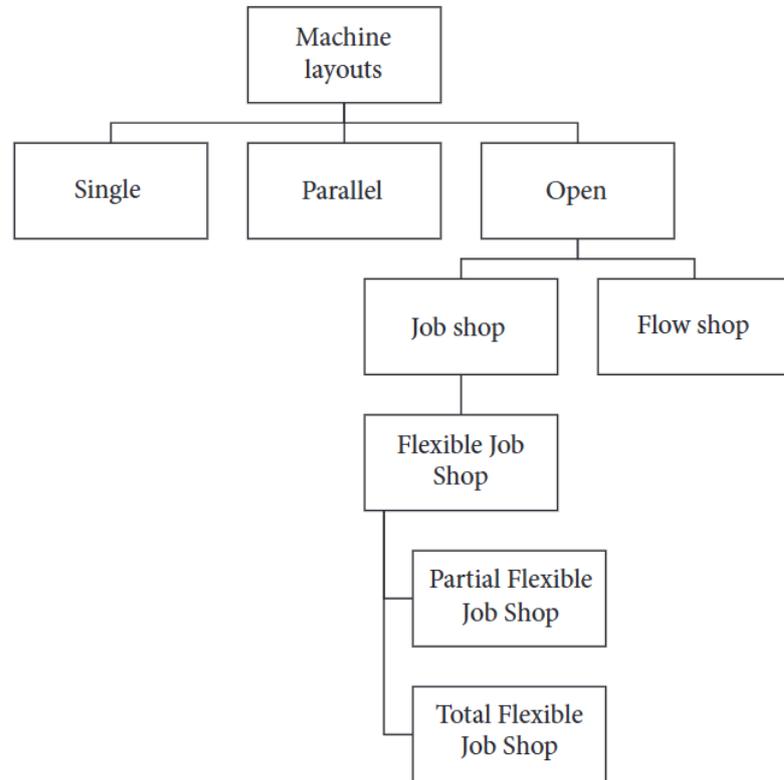


Figure 2.4: Classification of machine-based schedules, including the variations of job shop schedules [86]

resulting in alternate approaches being required. [80,90].

Open-shop scheduling, as shown in Figure 2.4, is a higher level from job-shop or flow-shop, where the sequencing of operations for the job is not predetermined, and the jobs must still be processed on different machines [86,91].

The evolution of these scheduling problems, shown by the Nagar et al. classification [82] being updated by Lin et al. [85] seen in Figure 2.3, has demonstrated them to be both deterministic and non-deterministic; based on the size and definition of the problem, and approach (or algorithm) used to solve the problem [74, 80, 91].

As the complexities of these scheduling problems increase, more constraints and optimisation criteria will be invoked. Đurasević et al. discuss the machine environment (α), constraints (β) and optimisation criteria (γ) for different variants of scheduling problems, collating expanded research building on Graham et al. [88] original classifi-

cations [92]. Figure 2.5 contains the overview of classifications of scheduling problems with respect to the machine environments, adding constraint and optimisation criteria to the outline provided by Amjad et al. in Figure 2.4 [86]; the full, extensive detail of the classification will not be discussed in this literature but can be reviewed in Đurasević et al.'s original work [92].

Depending on the availability and state of the variables, it may incur one of the two conditions for scheduling: offline scheduling, when all parameters are known in advance of the system operation, and online scheduling, where variables become available during the execution of the system [92].

The principles of the foundations of planning and scheduling systems are used throughout the derivation of satellite planning and scheduling systems. However, only an overview will be applied as the main focus of this thesis is on explainability and not scheduling.

2.2.4 Satellite Planning and Scheduling Systems

The general principles of planning and scheduling apply to space missions as much as they do in any other industry, ensuring that objectives are defined, stakeholders are identified, responsibilities are assigned, and activity requirements are outlined [93,94]. The scope of planning and scheduling processes can apply to both on-board and ground-based teams and systems, producing a combined Sequence of Events (SoE) maximising resource utilisation and free of conflicts [95]. While generalised planning and scheduling principles are commonly applied, there are also often mission-specific processes employed for such missions as International Space Station (ISS)-based, manned missions, space debris collection, and earth observation satellites [96–99].

2.2.4.1 Overview of Satellite Planning

Within the domain of satellite mission planning, it has been reported that it can take up to 7 years to plan and launch a satellite; however, most are planned and launched within 3 years [100]. Elsewhere, it has been presented that the satellite pre-execution planning lifecycle can take up to 12 months to complete, including specific functions

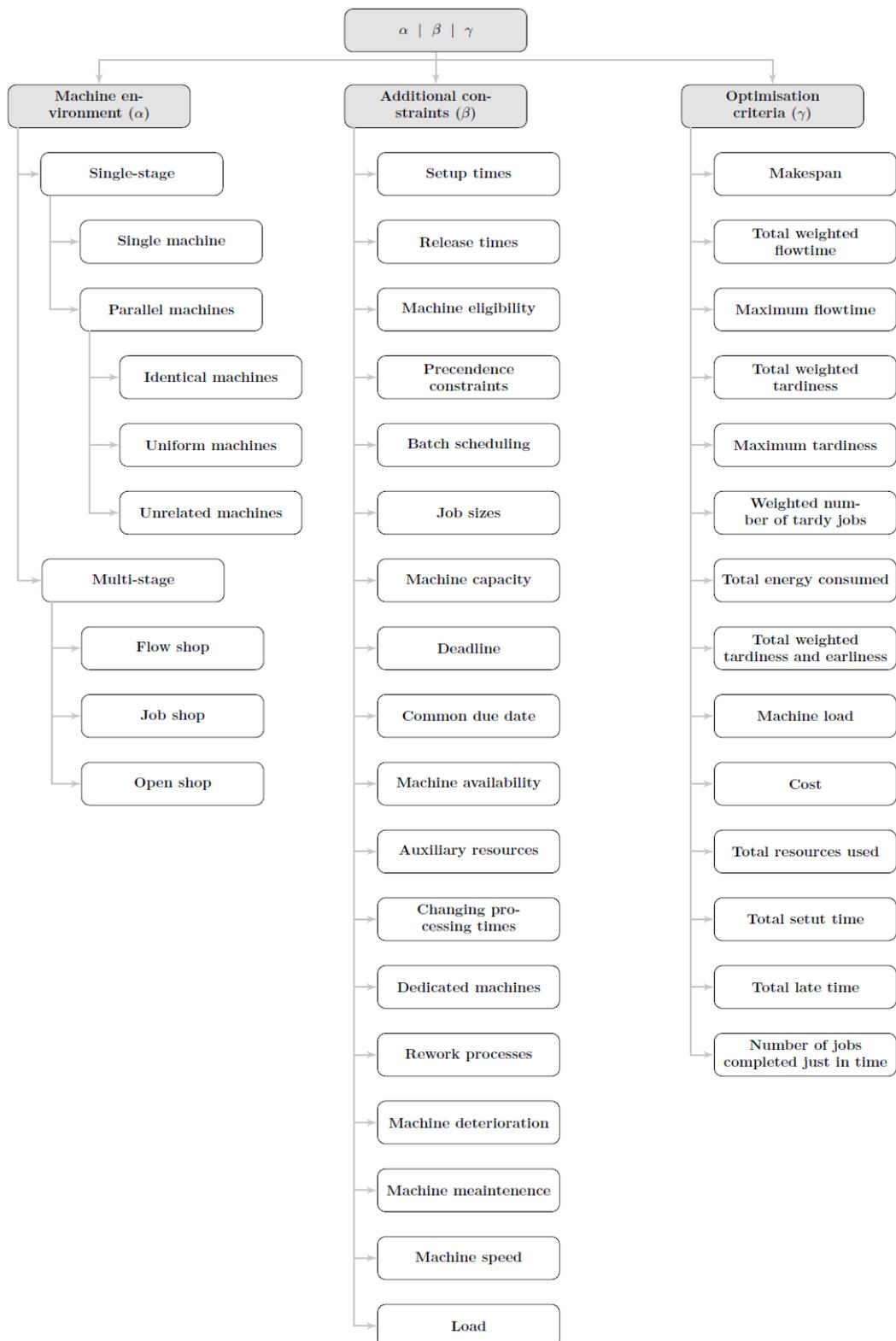


Figure 2.5: Scheduling problem classification overview with respect to machine environments and constraints [92]

such as selecting the launch vehicle, determining payload dynamics, trajectory and safety calculations [101].

An important distinction must be made between mission planning and satellite planning, as these terms have been observed to be used interchangeably in research [102]. Mission planning considers the full extent of requirements to achieve the mission, including but not limited to launch planning, the requirement for external resources, and human resource coordination [93]. Satellite planning is focused on the operation of the satellite system itself to achieve the mission and its objectives, including activities such as sensor calibration, position analysis, and power management [103].

Although for each mission, organisations may have their own definitions and planning parameters, there is a generally accepted principle approach of defining planning requirements by timescale, namely long-, medium-, and short-term [93]. Long-term planning establishes the primary objectives and the initial framework of requirements necessary to achieve the objectives, usually 20 weeks (or more) ahead of the expected execution. Medium-term planning builds on and refines the long-term plan(s), adding relevant details and validating previous calculations or decisions, usually around 8 weeks before execution. Short-term planning is performed about 7 days before the execution of the plan, completing the final verification of data and requirements and making any necessary last-minute adjustments [93, 102].

The fundamentals of satellite planning naturally aligns with the individual tasks that satellites are required to complete during a mission and operation. Therefore, satellite planning is often referred to as satellite task planning [104]. With the growth in popularity and capability of AI systems, this has been an area of increasing research, initially considered with Automated Scheduling/Planning ENvironment (ASPEN), introduced in 1997 [105, 106].

Since then, many more research initiatives have been explored, investigating the capabilities for specific satellite operations such as imaging, multiple-satellite systems (MSS), and communications [107–109].

Given the symbiotic relationship between planning and scheduling, the principles directly lead to satellite scheduling.

2.2.4.2 Overview of Satellite Scheduling

Satellite scheduling is a complex optimisation problem that utilises planning when allocating resources such as power, memory, and other requirements to achieve specified objectives. These objectives can range from missions for EO [110], communications [111] to deep space climate observation [112] missions. When comparing to Earth-bound manufacturing scheduling, additional constraints must be accounted for, such as visibility windows, orbital dynamics, and unpredictable environmental conditions of the satellite. As a result, minimising the operational cost while also maximising the satellite capabilities within these conditions is vital for the longevity of the mission [113–115].

As a critical factor for mission operations, satellite scheduling directly impacts the efficiency and effectiveness of satellite utilisation. When performed effectively, satellite scheduling ensures its tasks, some of which may include observation durations and downlink opportunities, are executed at optimal times to establish continuous coverage, maximising data acquisition, and conserving resources while also achieving mission objectives and keeping risk conflicts to a minimum [69, 110, 115].

Every task allocation for a satellite scheduling problem requires resources to operate (including the allocation of the payload resources), and based on the mission objectives, some tasks are prioritised over others, having an influence on the results, especially those of emergency importance that would require a near real-time to real-time resolution which requires a ground management, daily maintenance and control of resources [114, 116].

As the scale of satellite missions increase, so do the satellite scheduling problems, which, therefore, require scheduling systems to aid in the development of space missions through ground stations. One of the core challenges with communication between the ground station and the satellite is the optimisation of traffic, which is said to be NP-hard with the aim to minimise the total data downloaded [114].

Applications of satellites vary based on their mission objectives and payloads, some of which are multipurpose and used to accommodate different customers' needs [117]. Large-scale satellite missions would involve projects from large organisations such as European Space Agency (ESA), National Aeronautics and Space Administration (NASA),

Japan Aerospace Exploration Agency (JAXA), Canadian Space Agency (CSA), each with their own and shared ground stations [69].

Satellite scheduling missions can include EO missions that cover a wide range of projects and scientific missions, two of which are weather monitoring (Meteosat Second Generation (MSG) for cloud cover forecasting [118, 119]) and land observation (Landsat for land monitoring some of which are the detection of wildfires, biomass changes, land change [120]) [69]. Another example is the Observatory mission, used for exploring space environments such as the Deep Space Climate Observatory (DSCOVR) that monitors solar wind [121]. The mission objectives will determine the path of the satellite for example, an EO satellite may be in LEO for Advanced Land Observing Satellite (ALOS) [122], Geostationary Orbit (GEO) for Geostationary/Low-Earth Orbiting Radar Image Acquisition System (GLORIA) [123], Medium Earth Orbit (MEO) for weather and environmental forecasting [124], and High Earth Orbit (HEO) for GPS operations [125] [126, 127], as some examples.

2.2.4.3 Overview of Satellite Scheduling Software Approaches

There is computational complexity in solving satellite schedules, and thus, there is a probability that these may not be solved optimally due to their NP-hard nature. Therefore, as mentioned previously, heuristic or meta-heuristic methods may be used more for deriving feasible optimal solutions [69].

Utilising the Rolling-Horizon Optimization (RHO) strategy, Wu et al. [116] proposed a scheduling model, priority model and scheduling algorithms for prioritising tasks. Their scheduling model was constructed with several constraint conditions while the priorities of tasks were focused on targets, imaging, Track, Telemetry & Control (TT&C) and data transmission. Within their task scheduling model, a satellite scheduling problem process under the multiple constraint conditions was represented as a cyclical process shown in Figure 2.6. Additionally, their task priority model is based on the order of importance of the target images to be acquired through their determined weight coefficient and calculated quantitative values. Algorithmically, the RHO strategy involves a rolling window that determines the response of the scheduling

system. It also includes a periodic triggering mode, with static and dynamic scheduling processes, to account for fixed and emergency conditions. For the proposed task scheduling problem, three techniques were applied, namely heuristic, Non-Dominated Sorting Genetic Algorithm II (NSGA-II), and a Genetic Algorithm (GA). Their results were shown to be effective with their proposed model and RHO strategy.

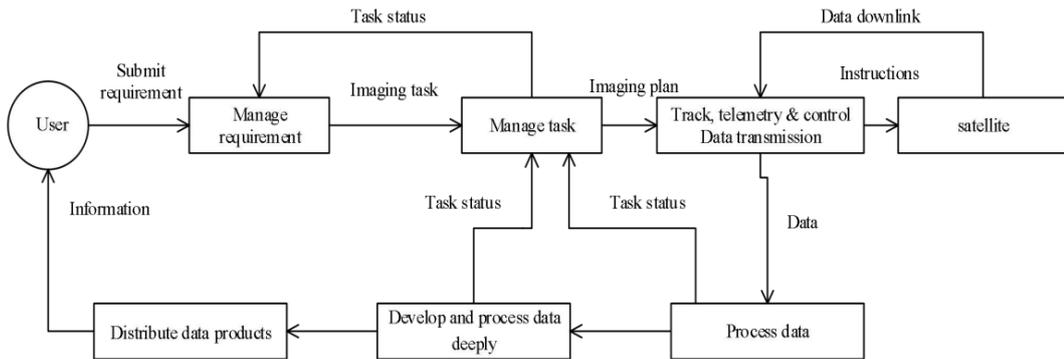


Figure 2.6: Satellite scheduling process [116]

ASPEN, developed for NASA, is an object-oriented software that is configurable, reusable, and also an application framework for planning/scheduling for rapid development of automated scheduling systems to allow for the execution of low-level action instructions. This reusable task scheduling system allows coding for flight rules, operable constraints, scientific experiment objectives, and other operational procedures [105, 128, 129].

The activity database is created, and its structure represents an activity entailing a start, an end, and a duration, allowing a hierarchical relationship between activities and enabling activities to use the required resources. Between activities, temporal and resource constraints are taken into account, and despite being done by other modules, a search algorithm is implemented [105, 128].

Satellite Tool Kit (STK), a 2-D and 3-D simulation software, is used for space missions' life cycle, including testing, manufacturing, operations, design, and other applications. It is effective for assessing the solution to the problem, and its main functions entail calculating the attitude of the satellite, orbit generations, and accessibility windows for land-bound objects and other airborne crafts and objects, including

physical constraints of the satellite [114, 129].

Through the lens of traditional scheduling, an example of the flow shop scheduling concept was applied in two stages using the Mixed Integer Linear Programming (MILP) for the Multi-Satellite Observation and Data-downlink Scheduling Problem (MSODSP) for a data downlinking and multi-satellite EO problem. This problem entailed three objective functions to minimise the makespan, the total weighted completion time and the success rates of all tasks while considering uncertain weather conditions. With scheduling constraints, each satellite can only observe one target at a time within the available window for observation, followed by data transmission and a data downlink that can only be scheduled within the available window for transmission. Using simulated data of real LEO satellites, the designed model with up to 20 tasks spanning across 8 days successfully solved the scheduling problem [130].

An example of applying the concepts of the flexible job-shop problem to satellite scheduling by deriving a mathematical model, including its constraints for activity onboard planning, was investigated by Liang et al [131]. A heuristic approach was taken to achieve the order of precedence for real-time planning, referred to as the scheduling process. Based on the sequential logic constraints, scheduling activities onboard due to their resources represents a similarity to the flexible job shop problem, where the conditions must be satisfied at any point in time concerning the limitations of machines. Figure 2.7 shows an example of the activities with their respective resource and logic constraints arranged with respect to time.

The search methods shown in Figure 2.1 are based on the objectives of the mission, which may be either deterministic or non-deterministic according to the problem and whether it needs to be resolved online or offline may require a more complex approach to solving the problem which may include AI tools to assist.

2.3 Advanced Scheduling Techniques

Advanced scheduling techniques, including AI, have fundamentally changed how scheduling is done through the introduction of empowering automated tools. Vassev et al. [132]

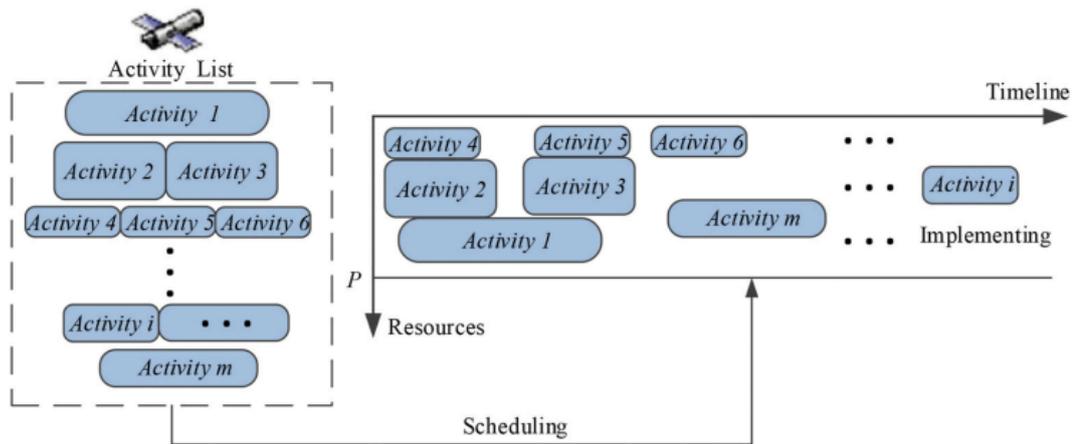


Figure 2.7: Onboard activity planning with respect to scheduling [131]

explored the differences between automation and autonomy, stating that automation is the process of software/hardware used step-by-step to replace a manual routine process. By contrast, autonomy is emulating human procedures instead of executing a sequential process. Laborie et al. [133] developed new algorithms to solve the challenges between resource and precedence constraints of activities versus their temporal positions for constraint-based scheduling. They analysed existing approaches used for timetabling and interaction techniques between activities through disjunctive constraints by analysing each pair of activities, edge finding that may be for both discrete and unary resources, and energetic reasoning involving the comparisons of energy resource requirements during a specified time interval. The new algorithms presented are energy precedence constraints, which focus exclusively on discrete resources and can adapt to dynamic resources during a search, and balance constraints, which consider the reservoir resource and calculate an upper and lower bound for the reservoir level immediately before and after each event in the precedence graph.

Constraint Handling In Prolog (CHIP), is a type of logic programming language that utilises constraint-solving techniques in tandem with Prolog (a tool created in 1975 for logic programming to solve small combinatorial problems [134, 135]) [135] was extended for the first time for tasks without fixed durations, and resource allocations to find an optimal solution for a 10 jobs x 10 machines problem while also ensuring

there was no overlap of jobs. [136]. Additionally, a well-known CP solver for over 20 years, used for solving scheduling problems, is the IBM ILOG solver, a part of the IBM ILOG CPLEX Optimisation Studio suite. Its performance is utilised across several industries: transportation, manpower scheduling, manufacturing, and network scheduling. It may entail piecewise-linear and step functions, with logical, sequence, resource no-overlap temporal, span, and alternative constraints. RL was utilised to build on the principles of CP to achieve encouraging results when solving various benchmark job shop scheduling problems [137]. It has also been noted to be used with satellite communication scheduling and EO Satellite scheduling [138]. When solving scheduling problems, the choice of search methods includes Blind Search, Stochastic Local Search, Algorithm A* and Variations [139], Large Neighbourhood Search, Depth First Search, and Random Seed [138].

2.3.1 AI Techniques for Satellite Planning and Scheduling

As AI solutions have advanced research has begun to determine the capability and suitability for AI techniques to support space missions and operations, including satellite planning and scheduling.

The high degree of complexity, dynamic nature and the need to derive optimal solutions through autonomous decision-making in satellite planning and scheduling have led to AI becoming an essential tool [132]. Currently, AI techniques have efficiently handled uncertainties, including large search spaces and dynamic environments [140].

Four autonomy levels of execution were explored for ESA mission operations for the autonomy problem, real-time ground control, pre-planned mission operations onboard, adaptive mission operations onboard, and goal-oriented mission operations onboard. [132].

Zheng et al. presented a solution to execute onboard re-planning during potential observation and communication emergencies for a Multi-Satellites System (MSS) through implementing a Hybrid Dynamic Mutation Genetic Algorithm (HDM-GA) and combining it with other techniques used in unmanned aerial and underwater vehicles that handle the insertion, deletion and modification of data [141]. The authors used a

reference mission for Orbit Low-Frequency Antennas for Radio Astronomy consisting of one primary (Mother Satellite (MS)) and eight secondary (Daughter Satellites (DS)); they considered three re-planning scenarios where an architecture was developed to solve the problem. The first scenario (*A*) considered the space environment that could impact the size of the fleet through physical impact from meteoroids and other objects. In contrast, scenarios two (*B*) and three (*C*) focused on the malfunction of components resulting in observation and communication issues. They initially generated a general planner, followed by a decision maker that enables the MS to examine the conditions of the DS through an emergency communication channel due to an error signal received. During recovery, the replanner uses a Cyclical replanning and a near-real-time planning method developed from a GA, which are utilised based on the time required for replanning and scheduled tasks that will be executed in the future, reducing the reliance on ground station operative intervention. An overview is shown in Figure 2.8.

Heuristic algorithms have also been applied to an Agile EO Satellites scheduling problem based on the Temporal Conflict Network (TCN). This model detected and resolved conflicts between tasks where overlaps and temporal constraints were identified. The heuristic algorithm also prioritised high-value tasks, thus resolving schedules by adjusting them where necessary. The basic TCN was constructed using a Node, representing the visible time windows, and an edge, represented as a weight, where two would exist between nodes with an overlap of windows. This helps calculate a path for related tasks to be completed in sequence and maximise schedule efficiency. The study introduced an improved TCN which includes additional exclusive relationships of the same tasks with different time windows that are more pronounced and, therefore, include more edges that allow for longer, more efficient paths [142]. A comparative representation is shown in Figure 2.9.

2.4 eXplainable Artificial Intelligence

As AI over the years has created remarkable results, becoming a frequently used tool for solving complex problems through techniques such as LLMs, decision-making, schedul-

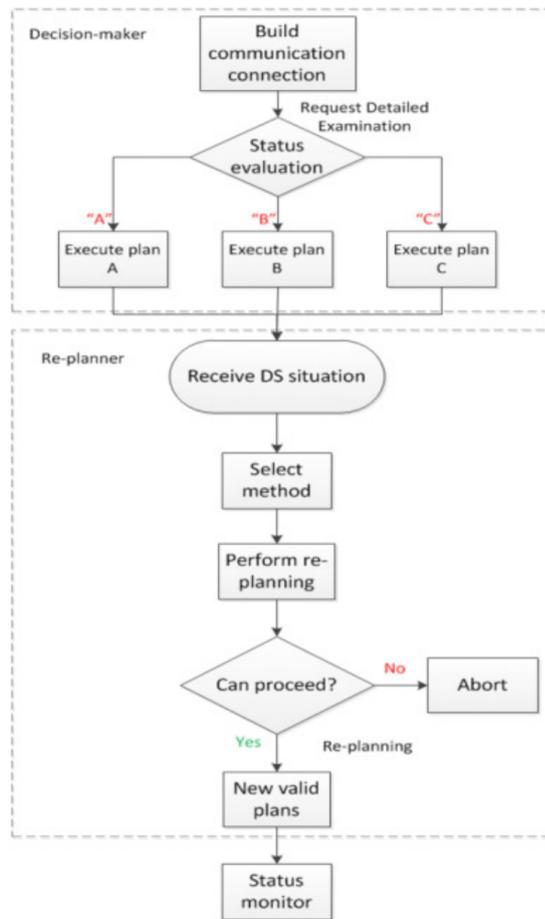


Figure 2.8: Overview of an onboard decision maker and re-planner between the mother and daughter satellite [141].

ing and image recognition. A lack of transparency often leads to trust, accountability and interpretability concerns [143,144], leading to the emergence of XAI. XAI is defined as a collection of ML techniques aimed towards enhancing models to be more interpretable while maintaining high-performance levels for prediction accuracy. Additionally, it enables users to understand, trust and manage their interactions effectively with increasingly AI systems. XAI incorporates insights from the psychology of explanations from social sciences to improve and achieve its approach to interpretability and user understanding [145].

Explainability, however, aims towards achieving the reasoning behind ML models, making them understandable and transparent with a focus on uncovering the logical

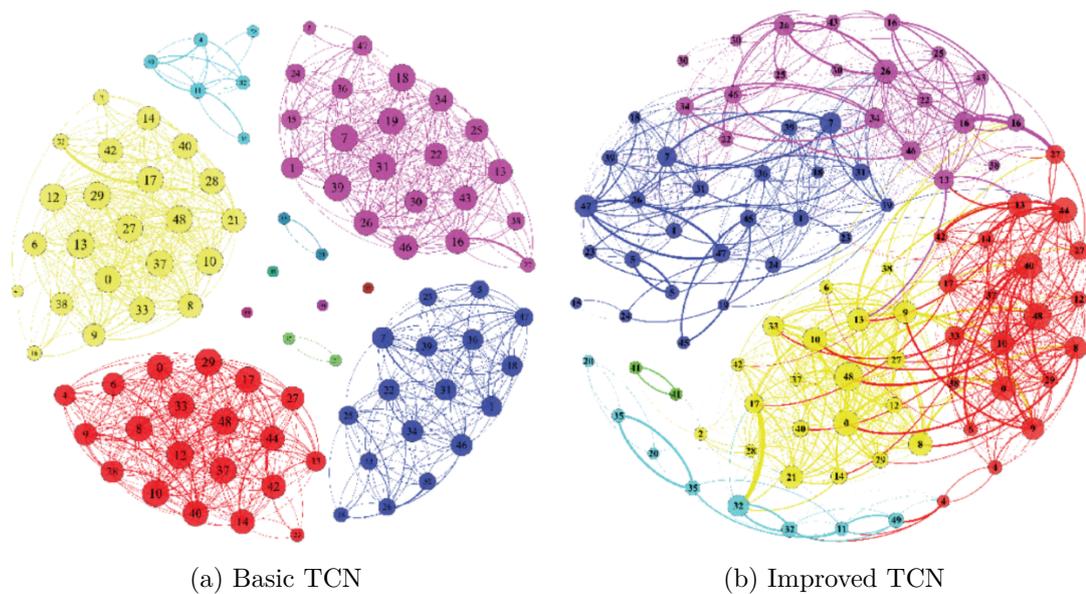


Figure 2.9: A visual representation of 50 nodes representing the basic and the improved Temporal Conflict Network (TCN) [142]

processes of why the system has arrived at a decision [146]. Explainability is often used interchangeably with interpretability [147–149]. Despite the close relationship in their definitions, these concepts differ. Interpretability is defined as the “*ability to explain or to present in understandable terms to a human*” [144, 150].

To understand explanations, it was highlighted that they must be social and require interactive communication with users [148]. Four maxims, quality, quantity, relation, and manner, were established to effectively exchange information for explanations regarding logic and conversation. Quality is necessary to ensure the explanations are high-quality and supported by robust evidence. Quantity, however, ensures the right amount of information is provided, striking a balance without overwhelming the users. Additionally, the Relation is based on only relevant information regarding the current discussion. Finally, Manner focuses on how the information is delivered versus its content [151].

2.4.1 A brief overview of Machine Learning

ML is achieved through algorithms that enable data and its properties to be analysed and its actions or predictions determined through statistical tools without being programmed to do so. Therefore, computers are trained to “learn” and “think” like humans with the introduction of more data, making these algorithms dynamic [152–154]. With a broad classification into four categories, ML algorithms consist of supervised, unsupervised, semi-supervised, and RL [152].

Supervised Learning involves the process of learning and mapping between the input and output variables according to the sample input-output pairs where, the output is predicted for the unseen data by applying the mapping [152,155].

Unsupervised Learning, however, is the method of receiving and analysing unlabelled inputs without receiving target outputs or rewards. [152,156].

Meanwhile, Semi-supervised learning is considered a hybrid of both supervised and unsupervised learning, which aims to improve the prediction results rather than only using the data from the model [152,157].

Lastly, RL is developed where perception and action establish a connection between an agent and its environment. Therefore, an agent’s training occurs in an interactive environment where a reward and/or punishment is received based on whether or not the attempts are successful [153,158].

2.4.2 Trustworthiness

The advancements in AI and ML have brought forward human trust in AI (Human-AI) interfaces. However, one of the key factors is how much trust a human has in these types of systems [159]. Trust is examined around how humans trust each other, known as interpersonal trust, a bi-directional transaction between two parties [160]. Human-AI trust, highlighted by Jacovi et al., is established when the human has developed confidence in an AI system’s capabilities to perform to specific expectations, including uncertain situations. When trustworthiness is attained in the model fulfilling the contract, individuals willingly accept vulnerability to the model’s response [159,160]. Additionally, trust in AI systems is warranted when the AI has demonstrated reliability

and trustworthiness through its ability to uphold its responsibilities, resulting in the human’s trust evolving. However, where the trust isn’t based on reliability, it is deemed unwarranted. The authors later highlighted that “*Trust in the AI model based on trust in the AI developer is an instance of interpersonal trust by proxy, and not Human-AI trust*” [160].

This statement does not highlight the importance of transparency, which is crucial in trustworthy AI systems, by clearly providing information on how decision-making processes operate, including their data [161]. A system is therefore considered transparent if the model itself is understandable [145]. Despite the growth of these complex systems that may lead to enhanced performance, there is often a cost of transparency. Keeping this in mind during the development of the design principles can drive more equitable decision-making, ensuring developers use crucial variables to generate model predictions [162].

To build and maintain trust relies on our ability to receive and interpret an explanation, which makes interpretability a crucial factor in establishing trust, especially for systems where the internal calculations and decisions are not accessible.

2.4.3 Interpretability

ML models, when deployed in the real world, interpreting their decisions becomes a critical component, as their role is to ensure the models are accurate and understandable [147]. Research has shown interpretability may not be required or possible for some ML models; for example, Doshi-Velez et al. stated Ad servers do not require interpretability [144], while Czerwinska stated certain models such as Supported Vector Machines (SVM) or highly complex Neural Network (NN) may be uninterpretable [163]. Conversely, alternative research has called out the importance of interpretability, even within complex models, particularly for mission-critical fields such as medicine, automation, and finance, where understanding a system’s decisions and reasoning is crucial for enabling adoption, trust, and accountability [164–166]. Rudin champions the development of interpretable AI models for these high-stakes areas, stating that reliance on post-hoc methods for traditional models may prove inadequate in achieving ethical

AI implementations and applications [167]. When working with Deep Learning (DL) methods, Chakraborty et al. discussed the existence of a complementary relationship between the elements of Bayesian reasoning and Deep Learning (Bayesian DL) within a uniform probabilistic framework towards achieving interpretability for NN [168].

There are two major types of interpretability: Perceptive interpretability, which is most relevant to this thesis, and interpretability by mathematical structures [147].

Perceptive Interpretability is the ability to provide interpretive data of ML models through means of observing neuron activation patterns (signal interpretability); input features highlighted related to their respective outputs (saliency) [169], and logical explanations formulated that may indicate causality (verbal interpretability). This approach enhances clarity by emphasising visual evidence. Saliency methods, however, quantify the importance of input features through probabilities, heatmaps or super-pixels. Overall, perceptive interpretability focuses on abstracting tasks, highlighting the most relevant aspects that explain the solution [147, 170].

Interpretability by mathematical structures involves mathematical models or data-oriented approaches that analyse clustering, dependencies and perturbations. This generated output necessitates deeper cognitive data and background knowledge for accurate interpretations. It combines various methods that explain mechanisms within deeper layers of NN algorithms. These would include testing techniques like t-Distributed Stochastic Neighbour Embedding (t-SNE) and Singular Vector Canonical Correlation Analysis (SVCCA), which help in achieving accurate predictions by identifying relevant subspaces [147, 170].

Taxonomies of interpretability are essential for understanding the trade-offs between various interpretability methods. These methods vary based on the type of algorithm and may be model-specific if there is a restriction in the application to a specified family; otherwise, they are model-agnostic [150]. Despite this, it has also been noted that explainability is often referred to as being model-specific, while interpretability refers only to model-agnostic methods due to the limited consensus in the research field [149]. Another factor to consider is the data type format (images, text, or tabular, for example) where these methods may be used. Figure 2.10 represents an overview of the

interpretability methods, including model-specific methods, highlighting the objectives they were designed to fulfil and steps taken to achieve them.

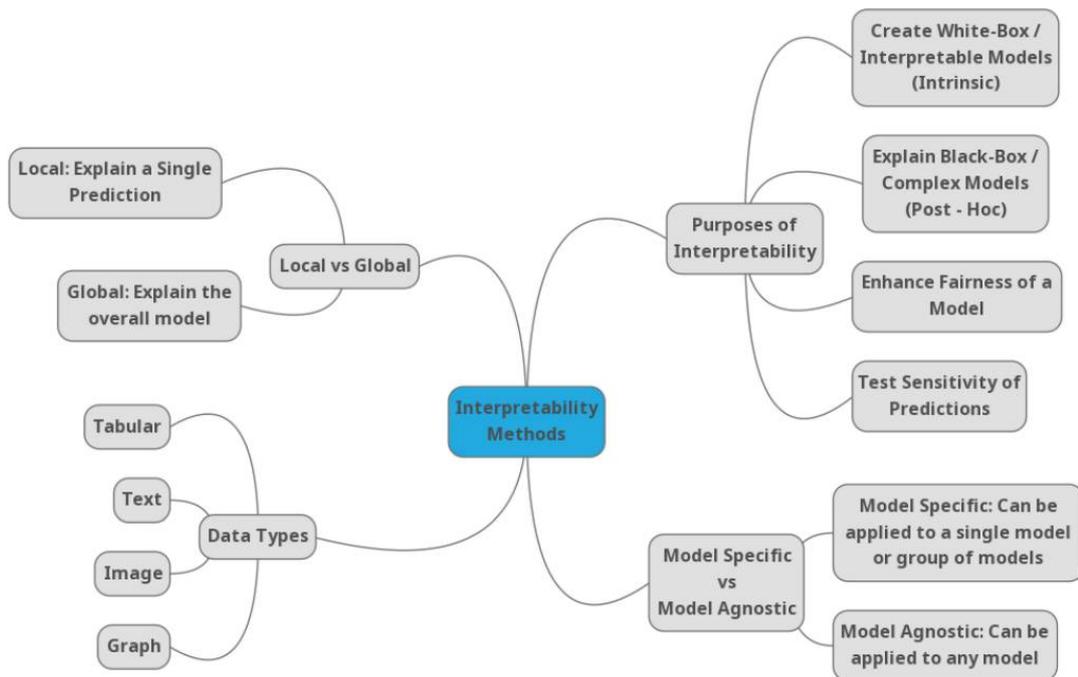


Figure 2.10: Taxonomy overview for interpretability [150]

2.4.4 Techniques and Applications

XAI has three main purposes: (i) *Justification*, providing the rationale behind decisions made; (ii) *Control*, to sanction identification and correction of inaccuracies, and (iii) *Improvement*, enabling the refinement of AI models to improve performance. May either be done independently or combined to enhance the overall capability of an AI system and ultimately build trust with users [171]. The conceptualisation of trust in AI has been seen as a contract with its users, where trust is maintained, where the system behaves as expected; otherwise, it is lost [172]. Therefore, XAI techniques can be instrumental in verifying AI performance and ensuring that satellite scheduling systems operate within expected parameters.

XAI contains a wide range of techniques targeted towards achieving more interpretable and transparent AI systems to users, building trust, especially in high-stakes

Chapter 2. Background

domains [144, 173, 174]. While some techniques have matured and are widely used in various domains, others are innovatively evolving to enhance model interpretability. Each XAI technique is categorised to represent different aspects of explainability.

1. Model-Agnostic Techniques - These techniques for interpretability are model-free and independent of the internal structure of the model. They may also be local (to explain specific decisions or instances within a specified range of data [175]) or global explainers (a rational overview for the entire dataset [175]) and are used to interpret Artificial Neural Network (ANN)s [148]. They are used for a post-hoc approach and may be applied to any trained model as it is not related to any architecture [145, 176]. Some of these techniques are shown in Table 2.1:

Table 2.1: Examples of Model-Agnostic XAI Techniques

Technique	Description	Example Use Case
Local Interpretable Model-agnostic Explanations (LIME)	Provides predictions of any classifier by learning an interpretable model around the prediction locally [145, 177]	Patient diagnosis [178]
SHapley Additive exPlanations (SHAP)	Based on Shapley values (present in game theory) and calculates contributions of every feature for a specific prediction [179]	Financial Credit Scoring [180]
Counterfactual Explanations (or Contrastive Explanation)	show how modifications in input features impact the output, establishing the decision boundaries [181]	Fraud Detection [182]
Partial Dependence Plots (PDPs)	Depict the relationship between features and produced outcome, without altering any other features [183]	House pricing [184]
Accumulated Local Effects (ALE)	Aim to build and improve on PDPs by incorporating the effects between the features as well, providing more holistic results [185]	Customer purchase patterns [186]
Individual Conditional Expectation (ICE)	Depict how changes to features affect the individual responses [187]	Quality assessment [187]
Feature Importance	Assesses the model-wide impact of each feature by measuring their respective contribution	Housing data analysis [188]
Sensitivity Analysis	Focuses on the uncertainty in a model and how sensitive the model is to variances, which can be applied both globally and locally [189]	Genetic Testing [190]
Anchors	Build on the LIME technique, aimed at providing greater precision and contextual awareness in explanations [191]	Image Labelling [191]

2. Model-Specific Techniques are designed to be used with specific ML models, and are also referred to as intrinsic methods [148]. This means the technique is implemented within or as part of the model during the training process [176]. Example techniques are shown in Table 2.2.
3. Rule-Based and Symbolic Techniques leverage logic-based systems, symbolic rea-

Chapter 2. Background

Table 2.2: Examples of Model-Specific XAI Techniques

Technique	Description	Example Use Case
Saliency Maps	Used in computer vision, extract the input features of interest and visualise them in a model’s predictions [192]	Computer vision [192]
Gradient-weighted Class Activation Mapping (Grad-CAM)	Creates heatmaps on feature importance and is commonly used to interpret Convolution Neural Networks (CNN)-based models, as they enable insight into which input areas are most influential in predictions [193]	Medical Data Analysis [193]
Layer-wise Relevance Propagation (LRP)	Applies to structured NNs to understand classification decisions that redistribute the models’ output back into the NN through purposely designed local propagation rules [194]	Multi-dimensional data analysis [195, 196]
Deep Learning Important Features (DeepLIFT)	Interpretable technique proposed by [197] that explains the predictions of Deep Neural Networks	Medical Modelling Assessment [198]
Decision Tree Explanations	Used for decision tree models to aid interpretability [199]	Medical diagnosis models [200]
Random Forest Feature Importance	Non-linear classification and regression technique which is centred on constructing groups of decision trees [201, 202]	Cybersecurity protection [203]

soning, and decision trees to provide explanations interpretable by humans, driven by knowledge or data [204–206]. These techniques are flexible and can be adapted to integrate with many varieties of models and provide global and local explanations [207, 208], with some examples shown in Table 2.3.

Table 2.3: Examples of Rule-Based and Symbolic XAI Techniques

Technique	Description	Example Use Case
Decision Rule Extractions	Captures decision points within a system to help provide insight into the influencing factors to a particular decision [209]	IoT device analysis [210]
Neuro-Symbolic Reasoning	Builds on Neuro-Symbolic Computing to utilise neural networks to reveal logic from within the model [211, 212]	Next-generation networks [213]
Fuzzy Rule-Based Explanations	Multi-valued approach to logic statements from linguistic variables [214]	Unmanned aerial vehicles [214]

4. Contrastive Explanations focus on answering the question ‘why P instead of Q ?’ and can be supplemented with a counterfactual approach, which considers the outcome of taking an alternative decision by analysing data inputs and the resultant output [215]. Table 2.4 highlights some examples.
5. Visualisation Techniques can be a means of visualising system data, as well as providing explanations for visual outputs [219]. A varied approach can be taken to graphically represent data or image-based explanations, with the benefit of making explanations independent of language and highly interpretable [220]. Some

Chapter 2. Background

Table 2.4: Examples of Contrastive Explanation XAI Techniques

Technique	Description	Example Use Case
Contrastive Explanations	Compares outcomes to derive the reasoning behind selecting one over another [216]	Investigation of Explainable AI Planning (XAIP) [217]
Counterfactual Explanations	Explore the decisions <i>not</i> made a system to build justification for the selected outcome [216]	Medical prediction and diagnosis models [218]

examples of these techniques are shown in Table 2.5.

Table 2.5: Examples of Visualisation XAI Techniques

Technique	Description	Example Use Case
t-SNE	Adapts the Stochastic Neighbour Embedding (SNE) technique to utilise simpler gradients and a Student-t distribution instead of a Gaussian to calculate similarities [221]	Medical imagery [222]
Uniform Manifold Approximation and Projection (UMAP)	Similar to t-SNE, used to capture low-dimensional information from raw datasets without following linearity assumptions [219, 223]	Customer data analysis [224]
Model Debugging Tools	Applies different techniques to create visual explanations, including, but not limited to, sample generation and weight comparison, feature contribution assessment, and global average pooling (GAP) [225]	Medical diagnosis [226]

6. Argumentation-Based Techniques, sometimes referred to as computational argumentation, is built on the principle of supporting or attacking conditions that influence decisions and outcomes [227]. By identifying and quantifying these conditions, explanations can be formed from the insights gained [228]. Example approaches are shown in Table 2.6.

Table 2.6: Examples of Argumentation-Based XAI Techniques

Technique	Description	Example Use Case
Argumentation Frameworks	Analyses the influencing factors of a decision or outcome, often assigning weights and quantitative values to determine the relative strength of a decision [229, 230]	Medical support models [231]
Dialogical Explanations	Allows for an interactive approach for users to have conversational explanations provided [232, 233]	Understanding recommender systems [233]

7. Natural Language Explanations techniques aim to create human-like written or graphical explanations, which can be conversational and easily understandable to users at all levels of expertise. There is a broad range of versatile approaches that can be utilised in almost all areas of AI application and research, building on decision tree, rule-based, salient mask, feature importance, and probing tech-

niques [234, 235]. The use of Natural Language Explanations has been applied directly to LLMs to help develop trust and transparency, with applications across many industries and interests, including robotics, fake news detection, and time series forecasting [234, 236, 237].

8. Knowledge Graph-Based Techniques create relational connected node graphs of datasets to facilitate explanations, which can be used for both visual and textual explanations. Data and information can be gathered from multiple sources to build large, interconnected graphs to support models in explaining the influencing factors to decisions and outcomes [238, 239]. These techniques have been explored for their capability of enhancing performance for business process analysis, drug repurposing, and recommender systems [240–242].

2.5 Overview of Large Language Models

AI systems have become more prevalent for businesses and personal users alike, none more so than the field of NLP, where LLMs have rapidly become one of the most well-known capabilities of AI. High-profile systems, such as ChatGPT from OpenAI or Large Language Model Meta AI (LLaMA) from Meta, have become household names in recent years, with thousands of research papers exploring the systems’ capabilities for real-world applications.

Fundamentally, NLP was first explored during the birth of AI systems themselves, with the creation of the Enigma machine and its later decrypting counterpart, Colossus; although it was not until 1957 that NLP as we might know it today was established, when Noam Chomsky developed syntactic structures. The next major breakthrough occurred in the 1980s, with research up until this point focusing on such utility as machine translation and object or token-based sentence structuring, as the increased utility and capability of ML paved the way for more advanced and effective research on NLP techniques [243, 244].

The critical outcome from the breakthrough in the 1980s was the development of Language Models (LMs), which aim to predict word and phrase structure based on an

unannotated history [245–247]. As the understanding of LMs and system performance advanced, the use of neural network technologies and deep learning techniques led to increasingly large datasets and training regimes [248], building on the capabilities enabled through the development of Long Short-Term Memory (LSTM) [249].

All the research on NLP over the last few decades culminated in two breakthrough developments from Google, the first being Google Neural Machine Translation (GNMT), which built on the capability of LSTM and utilised Recurrent Neural Network (RNN)s with 8 layers to foster gradient flow, with training datasets containing many millions of sentence pairs [250,251]. The second was the creation of the Transformer model, which was the first language model that solely utilised the concept of attention mechanisms within the system to generate even more performant results [252].

The advent of the Transformer model was the catalyst for the most recent major revolution in NLP, as the introduction of Generative Pre-trained Transformer (GPT) by OpenAI in 2018 paved the way for what is now known as LLMs, which are trained on and have access to huge amounts of words and phrases, and are able to respond to vastly varied natural language tasks and queries. [253]. The introduction of different LLMs in the last 6 years is succinctly summarised in Figure 2.11, noting that GPT-1 was the first, initially introduced in 2018 [254]. As of August 2025, there are more than 10 actively available and high-performing LLMs, many of which have undergone multiple iterations, with ChatGPT and GPT-4 arguably being the most well-known and utilised [255].

To assess the effectiveness of LLMs, several evaluation metrics have been established, as shown in Table 2.7, centred around core principle measures of accuracy, precision, and recall, with many specific metrics derived into three categories of Multiple-Classification, Token-Similarity, and Question-Answering [256]. Within the Multiple-classification category, Accuracy measures the percentage of true responses from a system as a direct indication of the model’s capability [257]; Precision measures the proportion of relevant components within the returned response [258]; and Recall measures the percentage of positive predictions made, based on all available positive examples within the dataset [259].

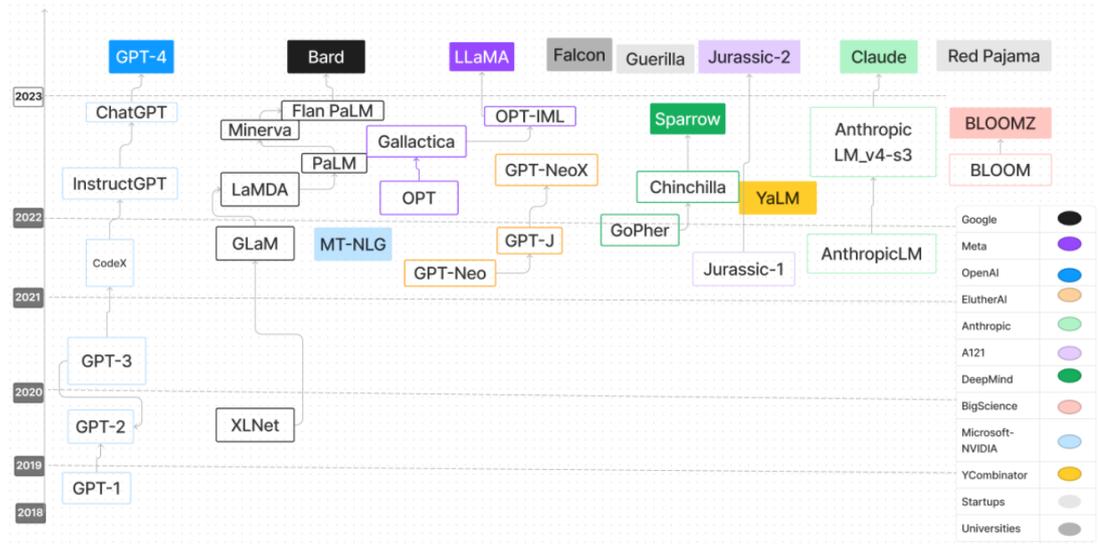


Figure 2.11: Growth of LLMs in recent years [254]

The measures of precision and recall are also combined to give an F1 score, which provides a balanced representation of response quality [256]. F1 scores, and the precision and recall components, can be further classified into two separate sub-metrics of Micro-F1 (assigning equal weight to every instance) and Macro-F1 (assigning equal weight to each class) [256]. The precision and recall measures are then combined to generate the F1 score using Equation 2.1 [260]:

$$F1 = 2 \frac{P \times R}{P + R} \tag{2.1}$$

The result is scored between 0 and 1, where 0 is considered completely unrelated and useless as a response, and 1 is perfectly aligned with reference material and, as a result, human-like.

For Token-Similarity metrics, which are focused on assessing performance at the token level, build on the combinatorial metrics of precision, recall, and F1 by adding the criteria of n -grams. An n -gram is a reorganisation of token sequences to assess the co-occurrences of tokens [256]. Several variations of the approach have been introduced, namely BLEU, ROUGE, and METEOR. BLEU advances on the metric of precision and excludes the measure of recall, returning high scores even for partial results. ROUGE

Table 2.7: Most popular LLM evaluation metrics by summary category

[256].

Metric	Purpose
Multiple-Classification	
Accuracy	Measures the number and percentage of correct responses or predictions
Recall	Measures a model’s ability to return phrases from example datasets
Precision	Measures a model’s ability to return relevant individual components
F1-Score	Combines Recall and Precision to provide an overall language quality score
Micro-F1	Combines Recall and Precision with equal weight for each instance
Macro-F1	Combines Recall and Precision with equal weight for each class
Token-Similarity	
Perplexity	Measures the appearance probability of a reference sequence
Bilingual Evaluation Understudy (BLEU)	Expands on Precision to measure generated tokens from reference material
Recall-Oriented Understudy for Gisting Evaluation (ROUGE)	Applies n -gram at the F1 level for determine alignment with reference material
Metric for Evaluation of Translation with Explicit Ordering (METEOR)	Derived from ROUGE, aiming to remove the negative impact of synonyms and word variants
Bidirectional Encoder Representations from Transformers (BERT)score	Combines Recall and Precision with pre-trained BERT embeddings to score against reference material
Cosine similarity	Measures the similarity between responses and/or reference material
Question-Answering	
Strict Accuracy	Measures the proportion of entirely correct predictions
Lenient Accuracy	Measures the top 5 responses for correct predictions
Mean Reciprocal Rank	Measures the average of both optimal and suboptimal correct predictions

applies n -gram at the F1 level and therefore includes the recall component, favouring more complete responses. METEOR is derived from ROUGE, aiming to remove the negative impact of synonyms and word variants on the performance measure [256,257].

Additionally, within the Token-Similarity category is BERTscore, which also builds on the measures of precision, recall, and F1, but instead of utilising n -grams draws on pre-trained BERT embeddings. The embeddings are initially loaded, and a fixed-length vector is found between each token of LLM returned texts and the reference material. A similarity measure is performed based on a combined sequence of vectors to calculate the BERT F1 score [256,257].

Cosine similarity is also used to determine the amount of similarity between statements X_G and X_A , measuring the dot product of two texts, represented as vectors, divided by the product of their magnitudes, as shown in Equation 2.2 [261]:

$$S_{(G,A)} = \frac{X_G \cdot X_A}{|X_G||X_A|} \quad (2.2)$$

The cosine similarity scores will range between -1 (no similarity whatsoever) to 1 (completely identical) [261].

The Question-Answering metrics are derivatives of accuracy, defined as Strict Accuracy, Lenient Accuracy, and Mean Reciprocal Rank [256]; however, in many cases of research, the assessment of accuracy is measured and presented as a singular specific metric [257]. These metrics are all scored from 0 (worst) to 1 (best), except for Lenient Accuracy (which should be less than 5) [256].

Although the aforementioned metrics cover a broad spectrum of aspirational qualities of an LLM, the use of human-centric metrics, such as Likert scales, is prevalent in the domain-specific fields, such as healthcare and legal services [262–264]. Moreover, the evident need for human expert evaluation and validation of LLM responses led to the investigation of how this could be provided as a framework to enable more consistent and less performance-impactful results [265].

By aggregating massive amounts of data and parameters (in the hundreds of billions) an LLM is able to facilitate more sophisticated tasks, consistently outperforming smaller-scale Language Model (LM)s [266]. The models are enabled by the Transformer architecture [252] and deep neural networks comprising multi-faceted attention layers. Two alternative Transformer scaling laws have been presented by two of the leading institutes in LLM development; KM scaling by OpenAI, [267], and Chinchilla scaling by Google [268]. Each law presented the relational performance and scalability in association with model size, dataset size, and amount of training compute. While both laws conclude an increase in performance as the allocated resources increase, KM scaling favours an increase in model size over increasing the dataset size, whereas the Chinchilla law asserts that the two sizes should be increased by equal scales [266].

Recent research has highlighted emergent abilities of LLMs, which separate them

from smaller models where these abilities are not observed, identifying capabilities with in-context learning, instruction following, and step-by-step reasoning [266, 269]. In-context learning is the ability for the LLM to gain additional insight from the provided prompt query, which has been sub-categorised into zero-shot, one-shot, and few-shot learning (or prompting), dependent on the number of demonstrations given at that time [270]. Instruction following or tuning, also called fine-tuning or Finetuned Language Net (FLAN), is achieved by providing additional tuning datasets to the LLM that aim to enhance zero-shot performance [271]. Step-by-step reasoning, such as the Chain-of-Thought (CoT) approach, adds additional thought process steps, such as breaking down the steps in math problem solving, to natural language queries as a form of prompt optimisation to facilitate more accurate responses from the LLM [272].

Consequently, LLMs are seen as generalised problem solvers, while also exhibiting the capability to handle specialised tasks as well (through such techniques as CoT) [266]. Additionally, LLM are being further enhanced through the integration with calculators [273] and internet search engines [274] to overcome challenges with advanced calculations or for accessing unknown information. However, due to the extensive amount of pre-training data involved, there remain concerns that accidental or inevitable low-quality training data will result in biased or harmful responses, which is currently being mitigated by implementing human feedback within the learning loop [275].

The broad and generalized capabilities of LLMs have drawn a lot of interest from both the commercial, civil, and academic communities with extensive research being conducted in the medicine, education, agriculture, finance, engineering, entertainment, legal, and marketing industries [254, 276–291]. The discussed research findings and purported performance of LLMs provide credible reason to investigate the ability for such models to support the advancement of scheduling systems and provide human language explanations for scheduling decisions.

2.5.1 Large Language Models with scheduling

With the wide utility of LLMs has shown, exploring how this could be applied to the advancement of scheduling techniques in various domains has been researched.

A comprehensive survey was conducted, assessing over 100 research papers on the capabilities of LLMs for the use in and advancement of Automated Planning and Scheduling (APS). The results of the survey found that the most common purpose for the research was plan generation, which 53 (42%) papers investigated, with more complex concepts of multi-agent planning or heuristics optimisation only featuring in 4 and 8 studies, respectively. The authors concluded that in order to realise the potential of LLMs for APS, more targeted research into training plans and datasets, integration with specialist language and concepts, and establishing suitable performance metrics to facilitate enhancements and innovation [292].

The use of LLMs for manually operated, non-automated scheduling systems has also been investigated for such functions as conference [293], construction [294], and systems task scheduling [295]. In every case, the research underlined the potential that LLMs represented through greater or comparable performance to existing methodologies, while acknowledging the results were modest and further research is required to build on the initial results.

The Job Shop Scheduling Problem, as mentioned previously, was applied to a purposefully pre-trained and fine-tuned LLM in one study to compare results to other neural approaches. The results demonstrated that the performance was consistent with other techniques; however, they highlighted the current limitations with the requirement for dedicated pre-training and the lack of interpretability that the experiment offered [296].

2.6 Background Conclusion

The background considered the difference between planning and scheduling, outlining that, in general terms, planning is focused on *what* activities need to be completed and scheduling will determine *when* the activities will be performed. The purpose of scheduling was found to be calculating the optimal solution to the required activities, based on the available resources and timescales; however, if an optimal solution cannot be derived, the best feasible solution can be selected. Schedules can be solved through

such techniques as tabu search, simulated annealing, or GRASP.

Schedules were found to be either deterministic, containing no variable constraints or resources, or non-deterministic, which contain variable conditions. One of the most prevalent scheduling problem examples is the job shop scheduling problem, which is characterised by the definition of machine environment, job constraints, and the optimisation criteria. The flexibility and broad use case of scheduling problems principles were found to generally fit with applications in satellite scheduling requirements.

Satellite planning and scheduling were examined, where planning will outline the overall mission requirements, while scheduling will focus on managing available resources to maximise mission objectives. Different approaches have been investigated to develop a satellite scheduling model, including RHO and ASPEN, with both heuristic and meta-heuristic also being explored for MSODSP and an onboard real-time scheduling which applied the concept of flexible job shop.

More advanced scheduling techniques were examined for their capability, with CP solver, RL, and stochastic local search solutions discovered to show promise for enabling more complex scheduling in future applications. These advances were seen to be important in meeting the growing demand for capable automated solutions to satellite scheduling, as bespoke algorithms like HDM-GA for a MSS are gaining interest for real-world implementation.

The rapidly growing field of LLMs, from the roots of NLP, have increased in popularity and capability, with industry-leading models of ChatGPT and LLaMA containing billions of words and phrases. The measuring of effectiveness of LLMs has been widely explored, adopting assessments of accuracy (also referred to as correctness), precision (selecting the right words), and recall (putting the words in the right order). Precision and recall can also be combined to give an overall language quality measure known as F1, which is the basis for many automated language assessment models, such as BertScore, ROUGE, and BLEU. However, in many cases, human experts are also included in performance assessments of LLMs to provide adequate insight to domain-specific areas, such as medicine. Developing even more accurate and reasoned responses from LLMs has also been considered through such approaches as CoT prompting or

dedicated pre-training.

The overall performance of these models has established them as generalised problem-solving and question answering, and they are being investigated for solutions within conference schedules, construction projects, and IT system scheduling. However, trust and transparency of automated systems have been highlighted as a barrier to adoption, which has led to the field of XAI to facilitate understanding in how and why AI make decisions and choose outcomes. Explainability and interpretability were discovered to sometimes be used interchangeably; however, these are distinct terms; explainability is the extent to which an explanation can be provided, while interpretability is the extent to which a response can be understood. Both of these concepts are required for effective XAI, along with system transparency, which is the measure of information made available on decision-making and reasoning. Many differing techniques of XAI can be implemented to generate explanations, with some specialised for visual-based explanations and others designed to examine decision trees.

Upon completion of this background, while there is research investigating the use of AI systems for satellite missions, operations and scheduling, no existing research (at the time of writing) was found to be examining the use of XAI within these systems. This gap allows for novel study into the potential capabilities, challenges and benefits of developing XAI for satellite scheduling models. For this research, a combination of example-driven, local, and post-hoc techniques in natural language explanations (achieved through the integration of an LLM), argumentation frameworks, and knowledge graphs was applied in novel methodologies to examine the performance in accuracy, comprehension, and quality of language. The aim is to provide explanations for a satellite scheduling problem that can assist in building trust with human operators.

Chapter 3

Satellite Scheduling Problem

3.1 Chapter Overview

This Chapter focuses on utilising concepts mentioned in Chapter 2 in deriving a satellite schedule used throughout the rest of this thesis, where XAI techniques will be applied. Therefore, it is not focused on creating a novel solution to scheduling problems nor to test the depth of the schedule proposed in this chapter; it is aimed at creating a baseline for an autonomous decision solution to build an explanation layer.

This chapter is separated into four sections:

- The literature review introduces the implications of constraints for mission objectives and CP solving techniques applied in achieving the satellite schedule used throughout this thesis.
- The satellite schedule problem was defined using a specified time period, which included tasks, constraints, and an objective function to generate a simple satellite schedule.
- Complexities were added to the definition of this baseline satellite schedule, where specific targeted areas with additional constraints were included.
- Further analysis was then explored to assess the recovery of a schedule where stochastic failures occurred during execution of a predefined task.

The first two points were used to assess the potential of XAI techniques when applied.

The schedule was created using a heuristic approach, followed by using a CP solver to develop a more feasible solution. The satellite’s recovery was then assessed through a scholastic failure of a task. The methodology, results, discussions and applications were adapted from *C. Powell and A. Riccardi, “Towards explainability of on-board satellite scheduling for end-user interactions”* [37] as well as *C. Powell and A. Riccardi, “On-board re-planning of an earth observation satellite for maximisation of observation campaign goals”* [38] published in the proceedings of International Astronautical Congress (IAC) 2021 and 2022.

3.2 Literature Review

As explored in Sections 2.2.3 and 2.2.4.2, all methods of solving a satellite schedule require the identification and inclusion of constraints that determine the parameters by which a satellite must operate. For example, some constraints will restrict when certain activities can be executed, or how often they can be executed, in support of maximising delivery on campaign goals [297]. The mission objectives for each satellite will determine the tasks to be executed, the resources required, the respective orbit and the opportunities for data retrieval by ground stations in fixed locations [298–300]. Schedules are derived by Ground Station Operators (GSO) and are uploaded to the orbiting platform; however, manually generated mission schedules have their limitations, as they can be labour-intensive. Satellite scheduling problems are often aligned with industry agnostic scheduling problems, such as job shop scheduling [301, 302].

Consequently, different scheduling techniques are being investigated to assist with scheduling, to reduce human intervention and enhance the robustness in responding to unexpected events, with one of the leading areas of research being on CP [137, 303–305]. CP has resulted in the development of CP solvers, which have shown to be capable of producing higher quality results when compared with alternative methods in recent

studies for EO satellites [305, 306]. Replanning in the event of randomised failure has seen studies utilise advanced genetic algorithms to meet near real-time responsiveness, potentially suitable for both small and larger scale satellite systems [141]. The projection of research in the field of satellite operations reveals a growing demand for automated solutions and, therefore, facilitating an XAI solution to validate system reasoning and develop trust is critical for assuring real-world applications in such satellite missions as distributed constellations and space communications [307–309].

3.3 Satellite Schedule problem definition

A simple EO satellite schedule was created using 5-second intervals across 24-hour time horizons to reflect an offline (on-ground) heuristic and job-shop scheduling approach, followed by the implementation of a CP-SAT solver from OR-Tools [310] reflecting an online scenario to optimise this solution and later investigate how XAI could be applied. In creating the schedule, constraints and realistic assumptions were made around the actions shown in Table 3.1.

Mathematical formulations were created using the constraints and assumptions with the following constants of the memory profile shown in Table 3.2.

Across the scheduling discretised time horizon T , any possible action $\{a_p, a_r, a_d, a_e\} \in A$ is represented in a binary decision matrix X shown in (3.1) where a_p represents taking an image, a_r for processing an image, a_d for downlinking an image, and a_e for idle time in all actions A .

$$X \in \{0, 1\}^{T \times A} \quad (3.1)$$

The constraint equations were formulated as follows:

$$\sum_{a \in A} X_{i,a} \leq 1 \quad \forall i \in 1, \dots, T \quad (3.2)$$

$$p_i = \sum_{j=1}^i X_{j,a_p} - \sum_{j=1}^i X_{j,a_d} \frac{D_m}{I_m} \quad \forall i \in 1, \dots, T \quad (3.3)$$

Table 3.1: Overview of simple satellite schedule constraints

Action	Symbol	Constraints
Taking of images	a_p	<ul style="list-style-type: none"> • Taking an image utilises approximately 2.6 GB of memory to accommodate all thirteen channels combined, with each requiring a size of 200 MB of Sentinel-2A's instrument utilizing it's swathe width. • An image will take 5 seconds to be taken, which includes instrument setup or movements prior to taking the image. • Images can only be taken during sunlight exposure when over land and a field of view is greater than $100km^2$. • The maximum number of images cannot exceed 80% of the satellite's onboard total memory.
Processing of images	a_r	<ul style="list-style-type: none"> • Onboard processing rate per image is assumed to be 50 MB/s and therefore utilises onboard memory. • For every instance of processing, due to the processing rate, it will only process a fraction of an image. • Image processing may only take place when there is at least 1 unprocessed image onboard and, therefore, must not process any more images than what has been taken at any point. • During processing, the memory utilised for image taking will not be removed and is therefore reserved to allow re-processing.
Downlinking of images to ground stations	a_d	<ul style="list-style-type: none"> • The downlink data rate used for this research is 280 MB/s (half of that used on Sentinel 2). • The process of downlinking images must not exceed the number of processed images. • Downlinking must only occur when there is access to the ground station • During the execution of downlinking, the onboard memory of the specific image taken and the image processed is released from memory and, therefore, will only downlink an equivalent amount of processed data, removing twice as much memory onboard.
Idle	a_e	Has no constraints and only occurs when no other action can be scheduled.
Additional Constraints		
Action Limit		Only one action is to be executed each time interval.
Memory	M	At any point in time, the overall memory must not be saturated with a saturation point of 80% to leave the remaining 20% for instrument calibration.

Table 3.2: Memory profile for actions

Name	Symbol	Description	Memory
Maximum memory	M_{max}	Maximum available memory onboard (80% of the available memory).	1920 GB
Image memory	I_m	Memory required to store an image.	2.6 GB
Processing memory	R_m	Memory utilised at every time instance during the processing of an image in the considered unit of time (5s).	250 MB
Downlinking memory	D_m	memory released for downlinking of processed images at every time instance. Each occurrence equates to a fraction of an image in the considered unit of time (5s).	1400 MB

$$r_i = \sum_{j=1}^i X_{j,a_r} \frac{R_m}{I_m} - \sum_{j=1}^i X_{j,a_d} \frac{D_m}{I_m} \quad (3.4a)$$

$$r_i \leq p_i \quad \forall i = 1, \dots, T \quad (3.4b)$$

$$d_i = \sum_{j=1}^i X_{j,a_d} \frac{D_m}{I_m} \quad (3.5a)$$

$$d_i \leq r_i \quad \forall i = 1, \dots, T \quad (3.5b)$$

$$m_i = I_m(p_i + r_i) \quad \forall i = 1, \dots, T \quad (3.6a)$$

$$m_i \leq M_{max} \quad \forall i = 1, \dots, T \quad (3.6b)$$

Equation (3.2) represents that only one action can be executed at any time instance i across the time horizon T . Variables p_i , r_i , d_i , m_i were created to represent the total number of remaining images taken in memory (3.3), processed images in memory (3.4a)

and (3.4b), and images downlinked (3.5a) and (3.5b); and the total memory used at any instance m_i (3.6a) and (3.6b). Additionally, it is important to note that (3.5a) is substituted in (3.4a) and (3.3) to remove the total number of images downlinked up to the specified instance in time.

An objective function (3.7) was created using equations (3.3) - (3.5a) to generate the schedule, with a focus on maximising the sum of the three actions at any instance i .

$$\max \left(\sum_{i=1}^T X_{i,a_p} + \sum_{i=1}^T X_{i,a_r} + \sum_{i=1}^T X_{i,a_d} \right) \quad (3.7)$$

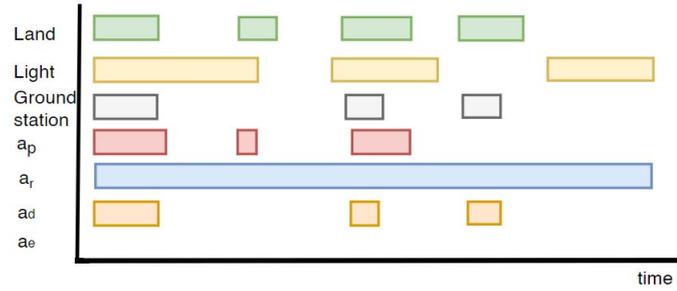
This objective function was designed to minimise idle activity, ensuring that the schedule prioritises any of the other actions to maximise the amount of images taken, processed, and downlinked aligning with the primary purpose of an EO satellite.

An example of how schedules are created, Figure 3.1a, represents all the possible execution times for the respective actions prior to the application of the constraint equations, only considering the environmental conditions. While in Figure 3.1b is a reflection of the Job-shop concept showing where actions a_p , a_r , a_d , and a_e do not overlap while also emphasizing the constraints that:

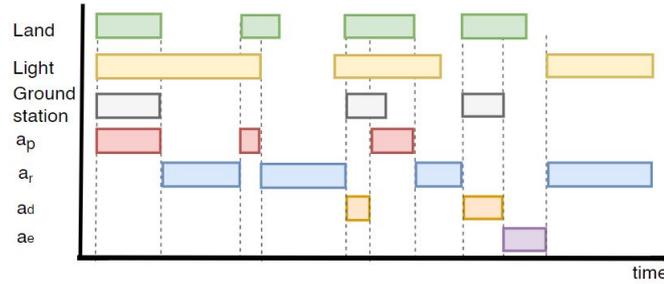
- a_p (shaded red) is possible with sunlight exposure (yellow) and land visibility (green).
- a_r (shaded blue) may happen anytime.
- a_d (shaded orange), only occurs with the line of sight with ground stations (grey).
- a_e should happen when no other action occurs, as shown in Figure 3.1a, a_e wasn't scheduled as a_r can be seen to be in execution across the time horizon.

3.3.1 Satellite Schedule Expanded Problem

The derived satellite schedule was later expanded to include predefined N areas of interest on Earth from the mission objectives that need to be observed, processed and downlinked to simulate real-world requirements. These areas are randomly selected



(a) An example schedule showing the possible action executions considering the environmental conditions.



(b) An example schedule generated using the environmental conditions and constraints applied to each action.

Figure 3.1: A reflection on the implications constraints have on a satellite schedule

to add complications to the problem. Additionally, constraints for power and image quantity requirements for each area of interest were also introduced.

Furthermore, the consideration of different onboard satellite architectures should be applied to examine different memory profiles and functionality, along with the targeted downlinking of specific images.

Each possible action is aligned with the originally derived satellite schedule, with constraints taken from Table 3.1 that must be satisfied before it can be executed. To reiterate, these are:

- Image-taking (a_p) - can only take place when the satellite is within light range (LR), has visibility of the target (VT), and has enough onboard memory (M).
- Process (a_r) - can only take place when there is an unprocessed image in memory (I_{mem}).
- Downlinking (a_d) - occurs when access to the ground station (GS) is possible and

there is a processed image (An_{mem}) in memory.

- Idle (a_e) - has no constraints and only occurs when no other action can be scheduled.

Each action, except idle, consumes onboard power P , which requires an execution time; however, during idle, it allows for power to be restored. In line with this, each mission goal (G) determines how many images are required from each area of interest, which corresponds to the total number of downlinked images required for each area of interest.

3.3.2 Satellite Schedule Action Recovery

Enabling satellite image recovery due to an anomaly occurring onboard relies on short-term scheduling to reduce the compromise of the mission's goals. Based on the random events that occur during a scheduled execution, the amount of computing resources would be based on the mission goals and the duration required for recovery from the anomaly, on the condition that recovery is possible.

For example, if an incident occurs during orbit B , cancelling an action a , where the following scheduled actions are independent of that cancelled action and can execute for the rest of the orbit, this will allow time for recovery on board to take place. However, if action a in orbit B at time instance i is critical for the next action, the demand for computing resources is increased until a solution is derived. If this is not solvable within the time for the next action, based on the criticality of this action for the mission, the system will then seek the next opportunity for that affected action to be re-executed by the satellite and, in return, have time to compute for the next opportunity of this activity to take place. When replanning occurs on board, the mid-term schedule will be updated, resulting in a short-term schedule. Here, the short-term schedule would attempt to recover from the failed action. Figure 3.2 shows a scenario where action c (highlighted yellow) failed during the execution of the mid-term schedule, resulting in the short-term schedule generated before later recovering the same action c and attempting to match the mid-term schedule by resuming to action b (highlighted

green). This means when action c is cancelled, the satellite schedules other actions until it can reschedule the cancelled action before continuing to action b (highlighted green).

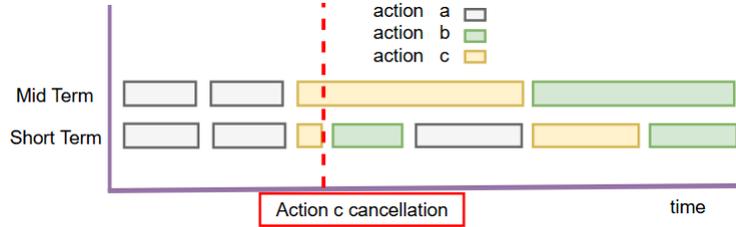


Figure 3.2: Short-term schedule created to recover from the cancellation of action c (yellow) during the execution of the mid-term schedule.

The cancellation of an image processing action is simulated due to the unusability of the captured image, necessitating a retake of the original image. The satellite’s orbit must be calculated to determine the next viable opportunity. Using this information, the system predicts the satellite’s path to estimate when the area of interest will be revisited, creating a short-term schedule for executing the action. This process is not limited to a single instance but can accommodate future occurrences, progressively building a database of opportunities for re-executing previously cancelled actions. As a result, 20% of the system’s overall memory is temporarily reserved for recovery until the action is completed, after which a regenerated schedule of actions utilises the remaining 80% of the memory.

The area of interest based on the satellite’s field of view, swathe, and path may not always be fully visible. An assumption was therefore made to ensure that a minimum of 50% coverage is attainable for the removed image to constitute a viable opportunity to schedule the task for re-execution. Therefore, recording the date, time, and satellite coordinates when the image was taken is crucial. Here is an example in Figure 3.3 illustrating the satellite’s position at different times. The processed instance is cancelled at position a on day n and is traced back to the time t when the satellite was at position b on day $n - \delta t$, capturing the initial image over the area of interest. The corresponding coordinates are extracted and stored. The satellite’s onboard system then calculates a predicted path starting from day n , identifying future opportunities.

When the satellite, at position c on day $n + \delta t$, has a field of view overlapping the area of interest by approximately 50%, the deleted image can be scheduled for retaking while considering constraints such as sunlight and shadow conditions.

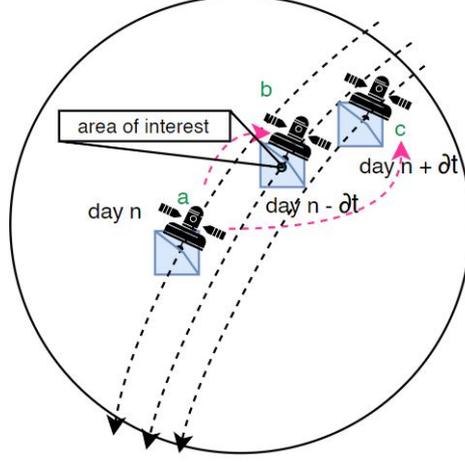


Figure 3.3: Illustration of image processing cancellation at position a on day n , showing the corresponding field of view. A pink arrow indicates the backtracked position of the satellite at b on day $n - \delta t$. The future predicted opportunity for image retaking is represented by the satellite at position c on the predicted day $n + \delta t$.

Due to the constraints, to process an image a_r , requires the occurrence of this action across several instances using the satellite schedule problem outlined in Chapter 3.3, equation (3.4a). To determine how many instances (R_{num}) need to be deleted from memory if the processing at time instance i was impacted. Equation (3.8) was created with the assumption that each processing stage contains sequential steps in analysing an image. This enables the calculation of the memory value of each image at each instance i .

$$R_{num} = \left\lceil \frac{r_i - \lfloor r_i \rfloor}{\frac{R_m}{I_m}} \right\rceil \quad \forall i = 1, \dots, T \quad (3.8)$$

3.4 Methodology

This section outlines the procedures taken using the information from Chapters 3.3, 3.3.1, and 3.3.2 in achieving a simple heuristic schedule, a feasible towards optimal solution, the capturing of targetted locations, and the process of deriving a schedule

through replanning due to a stochastic failure of the processing task.

3.4.1 Heuristic Schedule Creation

For the simple satellite schedule, its locations were consolidated and organised into a table under the assumption that image processing occurs when the satellite is positioned over the sea or over land in shaded conditions. It is therefore essential that image capture occurs when the satellite is over land and not in shade, while downlinking occurs when connections to ground stations are available.

Image capture and downlinking actions were prioritised in relation to processing, ensuring that these actions occurred sequentially without overlapping. This approach enabled the development of a heuristic method to calculate the memory requirements for each action, starting from the first image capture.

Memory requirements were calculated based on the sequence of actions. If the memory usage approached its limit, further image capture or processing actions were halted until a downlink occurred to free up space. Conversely, if the memory level dropped below zero, downlinking would be paused until a new image was captured and its processing completed. Algorithm 1 illustrates the heuristic approach used to calculate memory requirements under these constraints.

Algorithm 1 Heuristic approach for memory calculation

```
1: while Search through list of actions do
2:   Look for the first image to be taken
3:   if image has already been taken then
4:     calculate how many images are required to be processed (using equations 3.3
       and 3.4)
5:     if images have been processed then
6:       calculate how many downlinks can occur (using equations 3.4 and 3.5)
7:     else
8:       Look for the next instance for image processing
9:     end if
10:  else
11:    Do not start the calculation until the image is taken
12:  end if
13: end while
```

3.4.2 Solver Schedule Creation

The heuristic schedule was combined with environmental conditions (shade, land, and ground station visibility) derived from the collected data for each day from STK. This dataset was expanded to uniform time intervals, using the minimum start and maximum end times of all daily activities as the time horizon. Boolean encoding was applied to represent actions over time, as detailed in Chapter 3.3. The data from the heuristic schedule was provided as a first guess despite the feasibility to the solver in deriving the schedule to assess if the solver is capable of improving what was provided.

The objective function was formulated to maximise the number of actions completed onboard, thereby optimising the frequency of each action. Due to computational resource limitations, out of over 15000 time instances for every 24 hours, the total dataset was processed in sequential batches of 3000 time instances.

Integer variables, representing images taken and processed, were derived from their respective Boolean variables and used to calculate memory allocation. The integer variable for downlinking was employed solely to track the number of images downlinked and the corresponding memory freed. An overview of the process is shown in Figure 3.4

3.4.3 Targeted Areas of Interest

Building on the schedule generated during Chapter 3.4.2, additional and modified constraints were added to generate a more detailed and realistic schedule, which targets specific areas of interest for multiple observations through the taking of images based on the defined goals of each targeted area. An external decision support system was used to supplement the schedule with probabilistic weights for every possible action across each time interval.

The conditions for the execution of each of the actions remained identical to the originally derived schedule, with the exception of the additional conditional constraint for the visibility of a specified target for image-taking. The coordinates of N targets were randomly selected for a generalised approach and then fixed throughout the schedule generation.

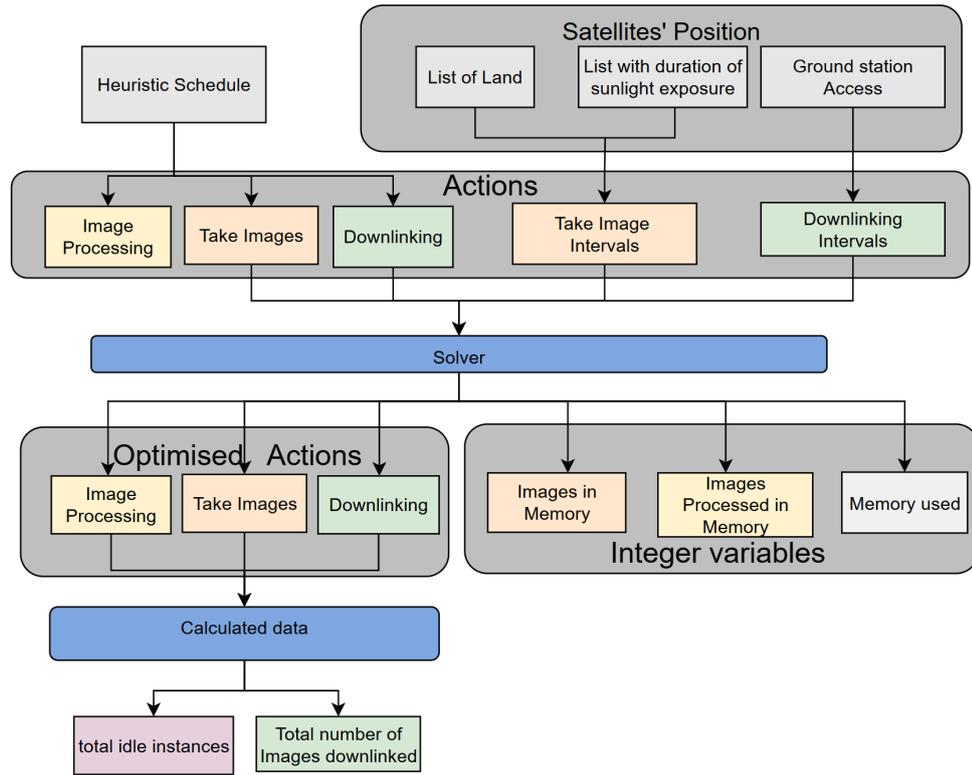


Figure 3.4: Schedule generation and solving process overview, where initial actions are optimised through the solver and further calculated for processing data.

Power consumption was added for each action, where a_p consumes 6% of available power for each execution, a_r consumes 5%, a_d consumes 6%, and the remaining a_e recovers 1% of power provided the satellite is within the light range; with the schedule starting with the satellite at 100% power.

In line with the power consumption, each action requires a fixed number of time intervals to complete execution. a_p and a_d both require 10 intervals to fully execute, while a_r requires 25, with a_e lasting 1 single time interval for each scheduled occurrence. It is important to note that the execution of any action by the satellite cannot be interrupted in any way and will always complete after the required time intervals.

Satellite mission objectives were also simulated through the addition of a required quantity of images for each of the areas of interest set as a mission Goal (G). The number required for each Goal was randomised between 0 and 10 for each targeted area of interest during the schedule generation. Once an image was downlinked, following

being taken and processed, the corresponding Goal number would reduce by 1, until reaching 0, at which point no further images for that area of interest are required.

The calculation of memory allocation upon the completion of each of the actions was implemented on a slot-based approach. Two sets of 30 independent memory slots were made available for both image-taking and processing actions, where images are stored until they can be downlinked. Once an image-taking action has been completed, a single memory slot for taken images is consumed, allowing for the satellite to schedule a processing action on that image, upon the completion of which leads to a single memory slot for processing to be consumed. After a processed image has been downlinked, both memory slots are cleared and made available for use again. If all available memory slots are consumed, then no further image-taking or processing action can be completed until a downlinking action is completed to make memory slots available again.

All other conditions, as presented in Chapter 3.4.2, remained the same to generate the schedule.

3.4.4 Rescheduling from a stochastic failure

A solution was implemented for the mid-term schedule to simulate the effects of cancelled actions being recovered in a short-term schedule. This involved introducing a random chance of image loss during execution to account for potential issues such as data corruption, cloud coverage, or other image quality problems. Figure 3.5 outlines the steps for the algorithm's recovery before generating the short-term schedule.

Fig. 3.5 details a summary of the processes of the algorithm involved with recovery before the short-term schedule can be generated.

The key steps in the recovery process are:

1. Cancellation Identification and Memory Update - On day n , a cancelled processed action is identified at instance i for image $Im_{\#}$. A backwards search removes subsequent processed instances from memory unless it was the first for that image.
 - An additional search identifies and removes the corresponding image taken previously on day $n - \delta t$, marked in red.

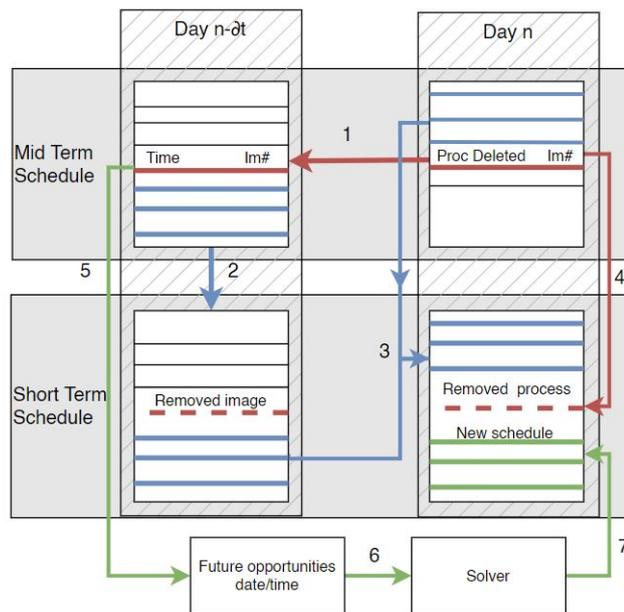


Figure 3.5: Steps required to recover when the next opportunity for an image can be retaken. Red represents changes that have negatively impacted the schedule; blue represents already executed tasks, and green represents the steps for recovery by the system that positively impacted the schedule, creating the short-term schedule.

- 2–3. The memory profile and objective function are updated, highlighting executed actions (shown in blue) with corresponding timestamps for streamlined tracking of future cancellations.
4. Flagging for Traceability - A flag is placed in memory to record the time and location of the cancelled instance. This flag is carried forward to the short-term schedule for day n , depicted with a red arrow, enabling traceability.
5. Future Search for Re-execution Opportunities - The time and coordinates for the cancelled image $Im_{\#}$ are extracted, and a forward search (FS) is conducted along the satellite's predicted path. Using a swathe value of 290 km (e.g., Sentinel-2) [311], the search ensures the satellite will pass within 145 km of the area of interest, covering at least 50% of the original image.
- 6–7. Feeding Data to the Solver - Once the next opportunity is identified, the solver integrates this information.

- For specific intervals Y (where a_p is possible for execution), the solver uses an adapted objective function (3.10) to prioritise the retake task.
- For other intervals where $i \notin Y$, the default objective function (3.7) applies. Sufficient memory must be available onboard to retake the image unless the ground station operator deems the area of interest unnecessary.

$$\max \left(\sum_{i \in Y} X_{i,a_p} \right) \quad (3.10)$$

Algorithm 2, based on the steps shown in Figure 3.5, generates the short-term schedule. During the execution of future actions, any cancelled instances trigger the algorithm to repeat, creating a closed-loop system for dynamic re-planning and recovery.

Algorithm 2 Recovery for task execution

- 1: **while** Search through the list of processed actions for the respective image. **do**
 - 2: Remove each processed action from memory, locate the original image with coordinates and delete the raw image from memory.
 - 3: **end while**
 - 4: Update the memory and objective profile.
 - 5: **while** Search through the calculated predicted path for coordinates. **do**
 - 6: calculate the overlap distance for every future coordinate with the coordinate of the removed image.
 - 7: **if** Coordinates overlap by at least 50% **then**
 - 8: Check that they satisfy environmental constraints.
 - 9: **if** Coordinates are in light and over land. **then**
 - 10: Store time as an opportunity for retaking and feeding to the solver with the updated objective.
 - 11: **else**
 - 12: Continue to search.
 - 13: **end if**
 - 14: **else**
 - 15: Continue searching for predicted coordinates.
 - 16: **end if**
 - 17: **end while**
-

3.5 Results and Discussion

3.5.1 First Guess vs Solver

To best reflect a real-world scenario in creating an EO satellite schedule, five months of Sentinel-2A's data (the dates of December 1, 2020, at 11:00:00.000 to May 18 2021, at 11:00:00.000) were collected using AGI-STK [312]. This was used to identify the orbit frequency of 14.3 orbits per day at a rate of 100 minutes, a revisit frequency of ten days, and an altitude of 786 km. The duration of exposure and access to several key factors were:

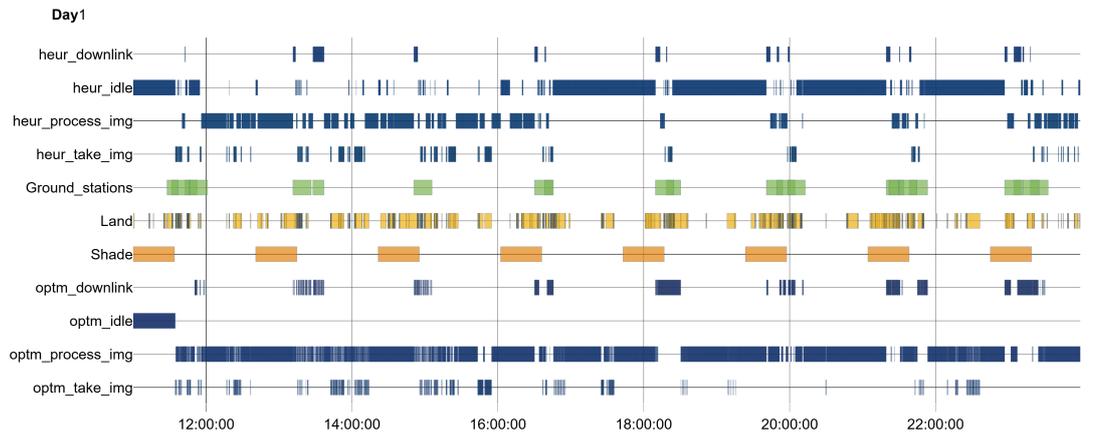
- Satellite coordinates - both latitude and longitude on a second interval.
- Penumbra start and stop times - for light and shade exposures.
- Land visibility - to record all durations land is visible to the satellites' instruments.
- Ground Station access - the durations for ground stations within a line of sight for the satellite. The considered ground stations and communications were:
 - XBand to EUMETSAT_Maspalomas (27°45'48"N latitude, 15°38'01"W longitude)
 - XBand to Matera_Station (40.6486°N latitude, 16.7046°E longitude)
 - XBand to PrudhoeBay (70°18'24"N latitude, 148°43'57"W longitude)
 - XBand to Svalbard_STDN_S22S (78.22875°N latitude, 15.39964°E longitude)

Using the coordinates and revisit frequency of ten days of Sentinel-2A in sun-synchronous orbit, fixed time intervals for light/shade exposures, land visibility, and communication access to ground stations were computed.

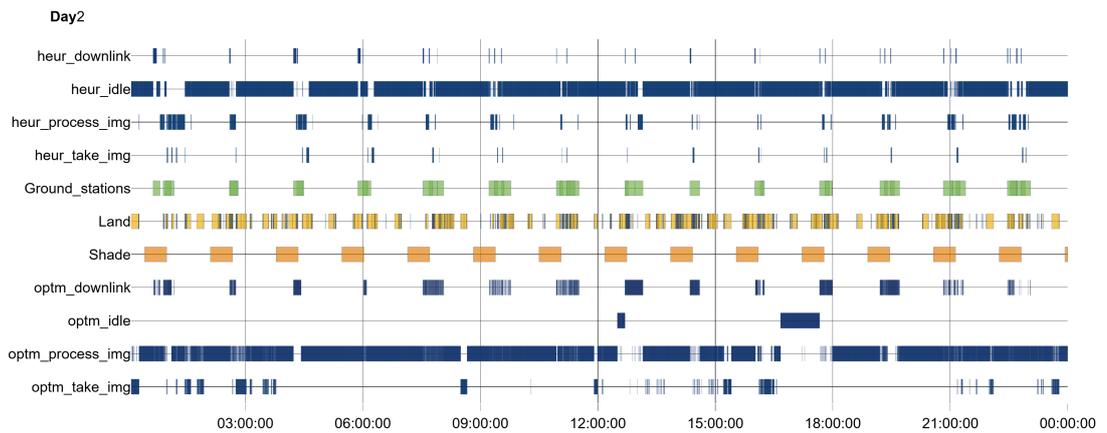
Initially, the schedule was calculated using the heuristic approach starting with Day 1 (1st of December 2020) midday with an assumed initial memory of 0 GB, which was hinted to the solver with all the constraints to optimise the schedule. Figures 3.6a and 3.6b represent the distinction between the two schedules for the first two days with the

Chapter 3. Satellite Scheduling Problem

respective environmental constraints. The top four rows of the Gantt chart contain the schedule using the heuristic approach, namely, downlink, idle, process, and take of images. Secondly, the next three rows represent penumbra/shade, land visibility and ground station access; and last, the final four rows are represented in the same order as the heuristic schedule but were solved using Google OR-Tools [310].



(a) Action instance comparison between both schedules for Day 1



(b) Action instance comparison between both schedules for Day 2

Figure 3.6: Action instance comparison between both schedules for Days 1-2

The variances between the heuristic and solver-based approach, shown in Table 3.3, represent a seven-day scheduling period (1st of December 2020 - 7th of December 2020), highlighting the differences in the actions, image taking, processing and downlinking of images, and instances where idleness occurs.

Table 3.3: Results for Days 1-7

Days	Total Img Taken		Total Img processed		Total Img Downloaded		Total Idle Instances	
	Heur	Solver	Heur	Solver	Heur	Solver	Heur	Solver
1	910	691	295	656	277	622	4746	416
2	280	1299	144	1224	215	1020	15029	851
3	209	1124	113	1204	176	1167	15511	956
4	235	1195	112	1166	170	1421	15511	816
5	238	1336	129	1262	184	1234	15294	0
6	205	1114	122	1211	172	1080	15428	1063
7	228	1241	106	1164	168	1133	15588	1342
Total	2305	8000	1021	7887	1362	7677	97107	5444

3.5.1.1 Image Acquisition

The heuristic approach across the seven days was scheduled to capture 2305 images while the solver has a total of 8000 images, resulting in a 247% increase. Looking at Day 1, though the solver took fewer images than the heuristic approach, this would be due to maximising the schedule period holistically rather than maximising the image acquisition per day, which could lead to bottlenecks in achieving the scheduling of the other actions, processing and downlinking. Therefore, reducing the number of images taken on this day would allow for smoother operations across the entire horizon and ensure that memory is not depleted.

On Day 2, there was a 364% increase in images scheduled to be acquired using the solver, resulting in 1299 images as opposed to 280 images scheduled to be taken from the heuristic approach. Similarly to Day 5, the numbers taken were 238 for the heuristic and 1336 for the solver. Overall, across all the remaining days, is a reflection of the solver's capabilities in improving the initial state of the schedule from the heuristic level.

3.5.1.2 Image Processing

Looking into image processing for this satellite scheduling problem and comparing the heuristic with the solver shown in Table 3.3, represents a bottleneck in maximising the objective function, a total of 1021 images were processed out of a total of 2305 images taken using the heuristic approach. Additionally, 7887 images were processed of the

8000 images taken from the solver (which is 98.5% of all images taken). For the total images taken across the seven days, resulting in a 673% improvement when compared to the heuristic method. From observing the trends, Day 3 presents the largest difference of the other days when comparing 113 images scheduled to be processed as opposed to 1204 from the solver, reflecting a 967% increase. The heuristic method may have either failed to schedule sufficient processing time or not optimised the sequence of image processing tasks.

3.5.1.3 Image Downlinking

One of the aspects of mission success is the downlinking of required data, where the solver's approach over a seven-day period has an increase of 464% with a value of 7677 versus the heuristic with a value of 1362. The most notable difference in scheduled images to be downlinked is on Day 4, where the solver has 1421 images scheduled compared to 170 for the heuristic. It can be noted that the solver maximises data transfer opportunities, ensuring that the captured and processed images are successfully transmitted to the ground.

3.5.1.4 Reduction in Idle Instances

A large number of idle instances is reflected in the heuristic approach, where scheduled opportunities were underutilised. With a total of 97107 instances with heuristic and 5444 using the solver, this represents an overall 94% reduction. The most outstanding improvement was observed on Day 5, where 0 idle instances occurred, thus further proving the solver approach is more efficient than a heuristic method.

A visual representation of the heuristic and solver results from Days 1 and 2 is shown in Figures 3.7 and 3.8. These graphs represent the memory being utilised (on the right) in purple, with the number of images taken, processed and downlinked at every instance (using the left of the graph) in colours blue, orange and green respectively. This means, when images are taken and processed, the downlink remains stagnant while the memory increases in line with the utilisation of these actions. Additionally, when downlinking occurs, the memory reduces while the number of images taken and processed remains

Chapter 3. Satellite Scheduling Problem

stationary. It can be observed that the heuristic approach attempted to capture as many images as possible, which then plateaued throughout the rest of the schedule, with few opportunities for processing and downlinking due to memory saturation. The solver showed an enhancement in system efficiency across all key performance indicators and, therefore, is considered a more global, mission-optimised strategy across all days.

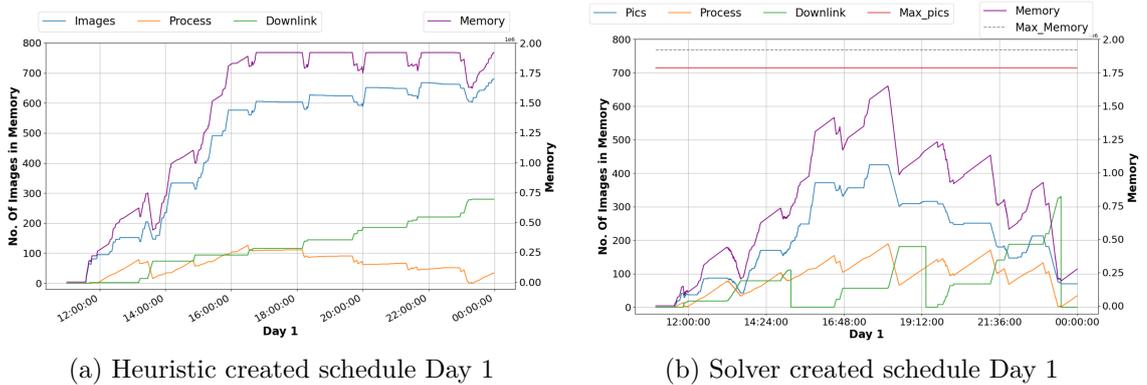


Figure 3.7: Day 1 Memory allocation comparison between the two methods. The number of images taken, processed and downlinked is on the left with respect to the memory utilised, shown on the right.

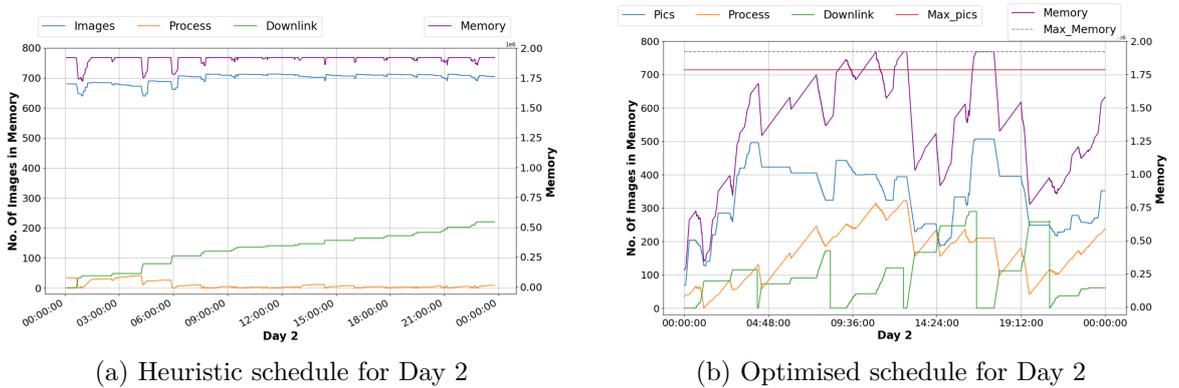


Figure 3.8: Day 2 Memory allocation comparison between the two methods. The number of images taken, processed and downlinked is on the left with respect to the memory utilised, shown on the right.

Overall, the solver has demonstrated a strong ability to dynamically adapt to scheduling constraints, significantly enhancing the satellite’s operational efficiency. Compared to the heuristic approach, the solver achieved a 247% increase in total images taken, a 673% increase in images processed, and a 464% increase in images downloaded while successfully reducing idle instances by 94%. This led to the question

of what would happen if one of these actions were to fail due to an anomaly occurring on board.

3.5.2 Targeted Areas of Interest

This section discusses the results utilising an alternate approach that focuses on specific targets of interest, with additional constraints applied to observe the variances between the two types of schedules created. The areas of interest are pre-selected as opposed to the previous subsection, where any land greater than $100km^2$ is observed.

Initially, ten targets of interest were selected, with each target given a goal of n number of images to be acquired. Each image of each target must then be processed before being downlinked, followed by the goal number being updated. With a separate memory slot to record the number of images taken and the processed images in memory, these are cleared after the downlinking of the processed image has occurred.

It was required that 31 actions be created, where each of the ten targets is assigned three actions, with the addition of an idle period where the satellite is unable to execute any alternative action. Figure 3.9 represents a snippet covering a 14-minute window showing the environmental conditions and the respective actions. All targets were grouped, followed by their respective scheduled actions of taking images, processing and downlinking. It can be observed that a_e occurred quite frequently throughout the selected time period, which is due to the specified targets limiting the frequency for image taking to be executed. These Targets are not represented individually, as there were instances where two or more were visible at the same time; thus, for simplicity, one colour (yellow) was selected, representing where at least one Target is visible. In this scenario, it can be seen that processing and downlinking occurred at various instances during the shade, and downlinking only occurred when a ground station was visible, as per the action constraints. Similarly, image taking only occurred when in sunlight exposure (the gaps between shade), thus complying with the constraint from the problem definition.

In support of Figure 3.9, Figure 3.10 represents the memory slots utilised and the recorded power consumption for the same duration. From analysing the patterns of the

Chapter 3. Satellite Scheduling Problem

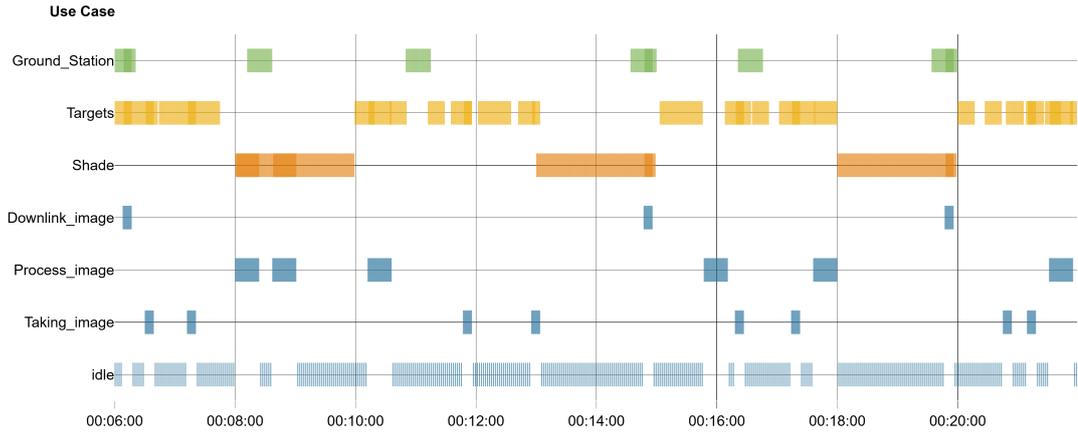


Figure 3.9: Schedule snippet representing targets of interest with all actions shaded blue, and environmental factors shaded green, yellow and orange.

memory utilisation, it was observed that when the images were taken and the memory increased, the power gradually decreased, with a similar pattern seen with processed images in memory. Likewise, when the memory values were reduced by means of downlinking, this action also utilised power, and the value gradually decreased. It is important to note the durations required for each action to complete execution, as prior to the memory increasing by one, the action will have already started executing, thus utilising the power before the image or process acquisition is completed, reflected in the memory slot once finished.

3.5.3 Rescheduling

This section presents an in-depth evaluation of the scheduling behaviour when processing tasks are interrupted, removed, or rescheduled due to stochastic failure using the initial schedule created by the solver. The memory profile and database updates in different scheduling scenarios were analysed, as well as the proposed recovery approach for image retakes and the subsequent schedule regeneration.

A short-term schedule during the simulation was created for the mid-term schedule for days $n - \delta t$ and n as shown in Figure 3.5 to allow for data comparability and traceability. Consequently, a file summarising future opportunities for image reacquisition was created, listing the corresponding dates and times for prioritisation based

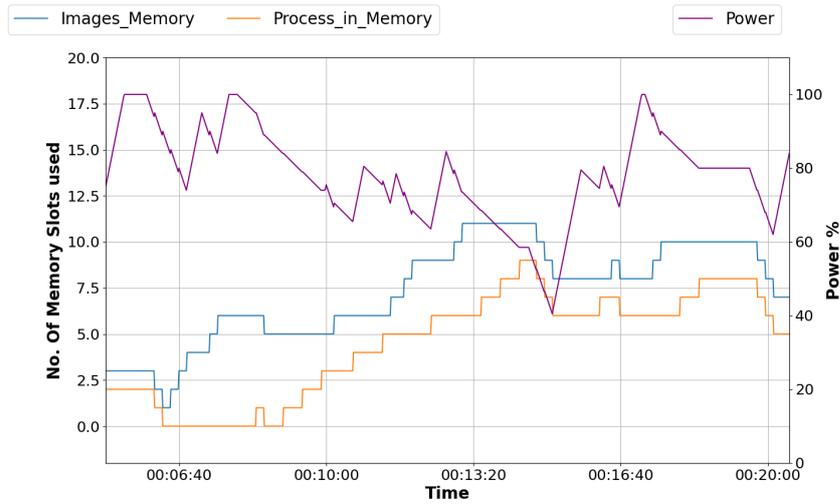


Figure 3.10: Corresponding Memory and Power plot for Figure 3.9

on the objective functions 3.10 and 3.7. These opportunities extended from at least 24 hours to several days into the future, achieving approximately 100% full coverage by the tenth day for an Earth observation (EO) satellite in a sun-synchronous orbit, with a 90-minute orbital period and an altitude of 786 km.

Additionally, since each instance i was set to 5-second intervals according to the satellite’s orbital path, the field of view could overlap the area of interest across multiple instances, resulting in several opportunities being considered along the same path. A 1-second interval was considered for improved accuracy; however, this would significantly increase computational demand and resource requirements.

3.5.3.1 Impact of Processing Cancellation on Memory Profile and Database

To illustrate the impact of task cancellations on onboard memory and scheduling, Table 3.4 presents a comparison between two datasets for day n : the mid-term and short-term schedules. It highlights the transition of task execution states and the subsequent updates to memory and processed instance values. Initially, all processed actions were scheduled for execution. However, at a randomly selected start time of 71 seconds (outlined in a red box), a processed instance was cancelled, resulting in the removal of the memory profile for previously successful processed actions of the same image.

This changed their task execution states from ‘Y’ to ‘N’ (also outlined in red), which is shown in the short-term schedule. The short-term schedule was generated with updated values for onboard memory, processed instances in memory, total processed instances, and the image number being processed (outlined in blue). Equation 3.8 was used as a reference to determine the number of processed instances that needed removal, as described in Section 3.3.2. These results were explored to represent an example of the method and were not executed as an exhaustive testing of the methodology.

Table 3.4: A representation of the day n memory profile and the effect in the database when a processed instance was cancelled from the mid-term schedule during execution at 71 seconds, creating an updated list, deleting the previous instances related to the image.

Time Stamp (s)	Task	On-board memory (MB)	Images in memory	Total number of images	Processed instances in memory	Total processed instances	Image number being processed	Total downlinked instances	Total downlinked images	Task Executed
Mid Term Schedule										
41	Process	1581796	351.91	1992	2571.6	20234	1881.88	3154	1642.7	Y
46	Process	1582046	351.91	1992	2572.6	20235	1881.97	3154	1642.7	Y
51	Process	1582296	351.91	1992	2573.6	20236	1882.06	3154	1642.7	Y
56	Process	1582546	351.91	1992	2574.6	20237	1882.16	3154	1642.7	Y
61	Process	1582796	351.91	1992	2575.6	20238	1882.25	3154	1642.7	Y
66	Process	1583046	351.91	1992	2576.6	20239	1882.34	3154	1642.7	Y
71	Process	1583296	351.91	1992	2577.6	20240	1882.44	3154	1642.7	Y
Short Term Schedule										
41	Process	1579108	350.91	1991	2571.6	20234	1881.88	3154	1642.7	Y
46	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	Y
51	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	N
56	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	N
61	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	N
66	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	N
71	Process	1579358	350.91	1991	2572.6	20235	1881.9	3154	1642.7	N

3.5.3.2 Effects of Image Removal from Mid-Term Schedule

Following the removal of the processed task, the system conducted a backwards search to identify the image that had been deleted during processing and remove its associated memory from the onboard storage and historical records. In this instance, image number 1883 was located at an execution timestamp of 79,441 seconds on Day $n - \delta t$ and subsequently removed from the mid-term dataset. This modification was reflected in the short-term schedule, as shown in Table 3.5 (outlined in red). To maintain consistency in future tracking, the next executed image was relabelled, ensuring synchronisa-

tion with the number of processed instances. As a result, the removed image memory was cascaded throughout the previously executed schedule until the cancellation. This forward-tracking adjustment is evident in Table 3.4, where the number of images in memory and the total number of processed images are reduced by one compared to the initial dataset, along with their corresponding memory values (outlined in orange). Notably, in both Tables 3.4 and 3.5, the downlinked data profile remained unaffected.

Table 3.5: A representation of day $n - \delta t$ depicting what happens in the database when the located scheduled image has been removed from the mid-term data, represented in the short-term data.

Time Stamp (s)	Task	On-board memory (MB)	Images in memory	Total number of images	Processed instances in memory	Total processed instances	Image number being processed	Total downlinked instances	Total downlinked images	Task Executed
Mid Term Schedule										
79436	Take Image	1103916	265.83	1882	1585.2	18990	1766	3108	1618.8	Y
79441	Take Image	1106604	266.83	1883	1585.2	18990	1766	3108	1618.8	Y
79446	Take Image	1109292	267.83	1884	1585.2	18990	1766	3108	1618.8	Y
Short Term Schedule										
79436	Take Image	1103916	265.83	1882	1585.2	18990	1766	3108	1618.8	Y
79441	Take Image	1103916	265.83	1882	1585.2	18990	1766	3108	1618.8	N
79446	Take Image	1106604	266.83	1883	1585.2	18990	1766	3108	1618.8	Y

3.5.3.3 Recovery Algorithm for Image Retake Opportunities

The recovery algorithm proposes alternative retake opportunities based on coverage feasibility to mitigate image loss. Table 3.6 presents an example of the output data, indicating potential opportunities for image reacquisition along with their respective coverage percentages based on the predicted satellite path and operational constraints. The results show that the earliest opportunity to retake the image occurred on the same day (n) that the processing failure was detected, corresponding to day 3 in this simulated scenario. Over a 10-day period from the initial image acquisition on day 2 ($n - \delta t$), a total of 92 potential opportunities were identified for reacquisition. Among these, three opportunities provided coverage exceeding 99% of the area of interest. Given the

extensive dataset, only a representative subset of these opportunities is presented for clarity.

Table 3.6: Proposed times for image retake opportunities from the recovery algorithm.

Image retake times		
day	Time stamp (s)	Coverage %
3	16846	50.20
...
5	19216	55.66
5	19251	68.65
5	19256	57.95
...
6	17451	65.49
6	17456	60.19
...
8	19821	60.92
8	19826	66.05
...
9	18026	69.97
9	18031	79.05
...
12	79436	99.54
12	79441	99.54

3.5.3.4 Regenerated Schedule from the Solver

After executing the solver with the variable objective function, the short-term schedule was analysed to verify whether the scheduler implemented the output from the recovery algorithm. As shown in Table 3.7, the solver successfully processed the instructions and rescheduled the prioritised task for both suggested time intervals, with comparisons highlighted in red boxes.

Figure 3.11 compares the mid-term and short-term schedules generated by the solver. Prior to process cancellation, the data remained consistent across both schedules. However, after cancellation, Figure 3.11a illustrates variations in the plotted values, despite some similarities. In this figure, the blue line represents images in memory; the yellow line denotes processed images in memory, the green line indicates downlinked images (which reset due to continuous accumulation), and the purple line represents total memory at any given time. Fig. 3.11b highlights the point where the

Chapter 3. Satellite Scheduling Problem

Table 3.7: Section of regenerated schedule from the solver due to the predicted coordinates from the recovery algorithm.

Time Stamp (s)	Task	On-board memory (MB)	Images in memory	Total number of images	Processed instances in memory	Total processed instances	Image number being processed	Total downlinked instances	Total downlinked images	Task Executed
Mid Term Schedule										
16836	Take Image	1907986	369.63	2147	3688.2	22829	2123.2	3418	1780.2	N
16841	Take Image	1907986	369.63	2147	3688.2	22829	2123.2	3418	1780.2	N
16846	Take Image	1907986	369.63	2147	3688.2	22829	2123.2	3418	1780.2	N
16851	Take Image	1907986	369.63	2147	3688.2	22829	2123.2	3418	1780.2	N
16856	Take Image	1907986	369.63	2147	3688.2	22829	2123.2	3418	1780.2	N
Short Term Schedule										
16836	Process	1524574	422.95	2348	1583.8	22315	2075.4	3702	1928.15	Y
16841	Take Image	1527262	423.95	2349	1583.8	22315	2075.4	3702	1928.15	Y
16846	Take Image	1529950	424.95	2350	1583.8	22315	2075.42	3702	1928.15	Y
16851	Take Image	1532638	425.95	2351	1583.8	22315	2075.42	3702	1928.15	Y
16856	Process	1532888	425.95	2351	1584.8	22316	2075.52	3702	1928.15	Y

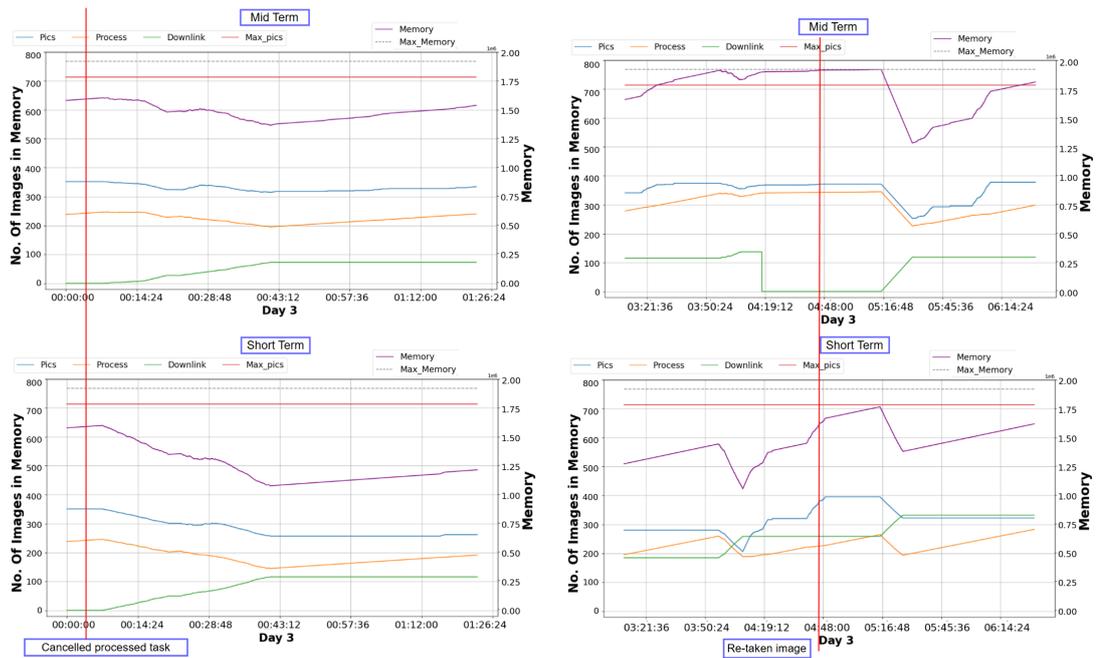
image was rescheduled for retaking on the same day, with variations in the plotted data resulting from the earlier process cancellation.

3.6 Conclusion

This research explored CP techniques for generating offline schedules for short-term and mid-term planning of an EO satellite in a sun-synchronous orbit. A heuristic schedule was created as an initial input to the CP solver, which then optimised the sequence of imaging, processing, and downlinking tasks while considering environmental and operational constraints.

The solver was able to make effectiveness and efficiency improvements in all areas, increasing the number of images taken, processed and downlinked, in addition to minimising the amount of idle time scheduled. This was also achieved through a more consistent and manageable use of resources, proving that the CP solver was proficient in producing more productive schedules, which can generate more output from a satellite

Chapter 3. Satellite Scheduling Problem



(a) Process Cancellation comparison between mid-term and short term schedules (b) Image retake comparison between mid-term and short term schedules

Figure 3.11: Day 3 before and after the process task removal, and image retaken on the same day. Purple - overall memory, Blue - Images in memory, yellow - Processed images in memory, green - Downlinked images in memory (frequently resets to zero)

schedule.

This was then built upon to create an expanded schedule for capturing specific areas of interest from across the globe. Targets with a randomised number of required images were added to the schedule, along with additional constraints for satellite power and an alternate memory management approach, used to generate a solution. The additional constraints and requirements created a unique schedule, which demonstrated the challenges with adding complexity to a scheduling problem. The results revealed a high amount of idle time being scheduled when alternative actions may have been possible, suggesting that even sophisticated solvers may require further research and enhancement to appropriately handle highly detailed and real-world mission goals and requirements. The schedule, however, was able to identify when the targeted areas of interest were available and avoid overutilisation of memory to ensure satellite operations continued without bottlenecking resources.

Chapter 3. Satellite Scheduling Problem

Additionally, this Scheduling Problem (SP) was simulated as an existing EO mid-term satellite schedule to evaluate a recovery algorithm's performance. When a processing task was randomly cancelled, a backwards and forward search algorithm identified the affected images and predicted future opportunities for retaking them. The recovery algorithm autonomously generated an updated short-term schedule, prioritising image retakes through an adaptive objective function. Compared to the mid-term schedule, the recovered schedule achieved the objectives while ensuring efficient and autonomous decision-making without human intervention. This demonstrates the system's capability in fault or failure tolerance, reducing the risk that mission goals and objectives are missed or under-delivered, allowing for missions to operate successfully for longer and with greater confidence.

Chapter 4

Explainability - Knowledge Graphs

4.1 Chapter Overview

This chapter explores the application of KGs to the satellite scheduling problem introduced in Chapter 3, assessing their suitability as an XAI technique. It details the design and implementation of a KG, followed by an analysis of the results. This Chapter is separated into three sections:

- The literature review discusses the applications of KGs with AI systems and their applications for XAI techniques.
- A thorough overview of the KG schema is provided, along with how it is created for the simple EO satellite scheduling problem.
- Followed by an evaluation of the application of the schema and how it is used.

The findings presented in this chapter are based on research published in “*Question Answering Over Knowledge Graphs for Explainable Satellite Scheduling*”, by *C. Powell, and A. Riccardi*, which appeared in the proceedings of the International Astronautical Congress (IAC) 2023 [41], as well as an extended version published in the Journal of Aerospace Information Systems of the American Institute of Aeronautics and Astronautics (AIAA) in 2025 [43].

4.2 Literature Review

Integrating KGs with AI systems has become very useful in augmenting decision-making processes demonstrated across various industries. Their adaptability establishes semantic reasoning, contextual understanding, and improved data organisation, making them valuable when complexities in relationships are required to be modelled and interpreted.

In the medical domain, KGs are used for building awareness by structuring patients' details to model drug interactions and assist in clinical decision-making. They allow for automated reasoning of medical diagnosis and provide visualisation of patients' histories, explain drug-drug interactions, and thus reduce errors in healthcare systems [313].

In marketing focusing on travel planning for tourism businesses and travellers, KGs have been adapted to support structuring and integrating tourism-related content and service packages. This would enable businesses to enhance content discoverability and improve direct booking channels by generating structured data for search engines [314].

In Additive Manufacturing (AM), KGs have been used with ML for extracting predictive AM based insights to improve manufacturability through constructing design rules. KGs facilitate knowledge extraction that is automated, where large-scale datasets are analysed by ML algorithms [315].

Additionally, in electrical grid control systems, KGs support the automation grid management through the grid day-ahead scheduling. They facilitate collecting, organising, and visualising large-scale grid data, enabling knowledge extraction, domain-specific phrase mining, entity recognition, and equivalent mapping to enhance decision-making and operational efficiency [316].

Though these applications display an accomplished use in KGs with domain-specific knowledge for enhanced decision-making, their application to planning and scheduling, particularly for satellite operations, remains under-explored. The increased demand for satellite data and autonomous space operations has triggered the need for onboard reasoning using the information extracted from a KG containing specific mission details to determine mission specification suitability based on its hardware characteristics [317].

An emerging area of research is the integration of KGs with XAI to attain effective results in data insights. XAI aims to provide information on AI models' decision-making processes by generating recommendations and ensuring interpretability for human operators. A survey on the use of KGs for XAI was explored, where 40 articles between 2018 and 2021 were focused on healthcare applications, with most AI methods focused on NNs, followed by NLP driven models [318].

Another research explored how KGs in applying different XAI methods, discussing the challenge, approaches, limitations, and opportunities. The study concluded that KGs could be utilised to make XAI more adoptable for large-scale systems, specifically calling out the opportunity in the area of automated planning and scheduling to provide greater context to complex terms and actions [319].

4.3 Knowledge Graph Schema

A satellite schedule is made of several interconnected components that influence the ability of tasks to be scheduled for specified timelines. To better represent these components, a structured data modelling framework known as a KG was explored. KGs are generally made up of three fundamental elements: Entities, Relations, and Attributes, to organise the linked data systematically. For this scheduling problem, the environments, events, objects or components are represented as Entities that make the foundation of the KG. Relations, however, define the relations between two or more entities, classifying their associations, while Attributes record specific data points as values related to both Entities and Relations. These elements are interconnected through edges, representing relationships between nodes, such as ownership, usage, access, containment, etc. [320]

To structure the onboard data of the satellite schedule as a KG, each action is defined as an entity that is paired with its respective attributes and relations. Five primary entities were created to represent not only the actions but also the environment, coordinates, and memory unit, representing the information stored in the memory at each instance in time t across the schedule horizon.

Chapter 4. Explainability - Knowledge Graphs

- *Satellite* - The primary entity, containing an ID number (traceable to time instances), is connected to all the other entities through their respective relations:
 - *Installation* - linked to the *Memory_Unit* entity
 - *Schedule* - linked to the *Action* entity
 - *Localisation* - linked to the *Environment* entity, which contains its' own attributes *start* and *end*.
- *Action* - Provides information on the scheduled action and the time instance of execution *i*. Which includes attributes such as the *name* of the action scheduled (image taking, processing, downlink, including idle), *memory_requirement* that represents the memory used or released for the scheduled action, and the scheduled execution time *a_timestamp*.
- *Memory_Unit* - Represents the status of the hardware memory onboard at the current moment at time step *i*; two attributes represent the memory status, one for the maximum capacity (80%) of the overall memory and one for the current capacity.
- *Environment* - Responsible for the state of the environmental conditions at the current time, and includes four attributes: status of *land_visibility*, sunlight illumination (*Daylight*), *latitude* and *longitude* coordinates.
- *Ground Station* - Overlaps with the *Environment* entity through the *Station Access* relation, is necessary for tracking the relative position between the satellite and the ground station at the currently scheduled instance *i*.

The *Memory_Unit* entity has a relationship with four additional entities designed to model the information stored in memory from previous time instances. They are:

- *Image* - models the images saved in memory (*im_size*) and encompasses several attributes, including the number of images ready to be processed and the total number of images captured up to time instance *i*.

- *Processed* - models the processed images still in memory and encompasses several attributes, namely the memory usage for each processed image (*pr_size*) and the total completed at time instance *i*.
- *Downlinked* - models the number of images scheduled to be downlinked and encompasses several attributes, namely the memory freed after each completed downlink (*dl_size*) and the overall number of images transmitted up to time instance *i*.
- *Idle* - models the idle action (this is created in the schedule when no other action can occur), it includes a single attribute representing the cumulative instances of idle actions (*total_idle*) up to time instance *i*.

Following the definition of the KG schema, the population of the database with the satellite schedule data containing all actions across the time horizon is required.

4.3.1 Population of the Schema

To populate the KG with the satellite schedule data, the data must be arranged in accordance with the predefined KG schema. Each entity is assigned its respective attributes, and relationships between entities are established accordingly. The process was implemented using Algorithm 3 with each schedule data entry mapped to the schema shown in Table 4.1.

Algorithm 3 Schedule data migration to KG

- 1: Identify the entity to populate
 - 2: Insert data to attributes within each entity
 - 3: **if** State conditions are required and met **then**
 - 4: Move state values to attributes
 - 5: **else**
 - 6: Set alternate value to attribute
 - 7: **end if**
 - 8: **if** Relationship exists **then**
 - 9: Create the containment relationship type with all entities or attributes within the relationship itself
 - 10: **else**
 - 11: Proceed to the next entity
 - 12: **end if**
-

Table 4.1: Table Representing the scheduled data points mapped to each KG attribute at every time instance.

KG Attribute	ID	latitude	longitude	land_visibility
Scheduled data	n	latitude	longitude	land_access
KG Attribute	daylight	name	a_timestamp	pr_in_memory
Scheduled data	day_access	extracted_action & action_possible	start_time	processed_instances_in_memory
KG Attribute	total_processed	total_sent	im_size	pr_size
Scheduled data	total_processed_images	total_downlinked_images	im_size	pr_size
KG Attribute	dl_size	total_idle	memory_requirement	im_in_memory
Scheduled data	dl_size	total_idle_instances	action_memory	pics_in_memory
KG Attribute	im_total_taken	current_capacity	start	end
Scheduled data	total_pics_count	memory_used	start_time	end_time

Once the satellite-scheduled data has been migrated, the retrieval process depends on the nature of the queried data. The data type and the required information extraction method determine the specific querying approach.

4.4 Results

This section presents the findings in creating the schema for the KG, the efficiency of the data migration and the query performance.

4.4.1 Validating the schema

The populated KG was validated against the schema as defined in Chapter 4.3 to ensure the satellite data was accurately represented. Figure 4.1 represents a visualised reflection of the KG schema within TypeDB studio [321]. It correctly shows the interconnection between the edges of the components with their respective relations, entities, and attributes. The entities are pink rectangles, while the relations are orange diamonds, and the attributes are blue ovals. This means that a schema is initially required before data can be migrated. The data is categorised into groups for the schema, where, at any point in time, there is an overlap or similarity in value, which will be represented as a connection.

Statistical analysis was done to verify the data stored within the KG. To do this, an entire dataset for a day (17277 time instances) were migrated from a 24 hr period (Day 3) through the execution of a migration code. A query was then executed to retrieve all data, resulting in a total of 155493 entities, 75213301 relations and an average of

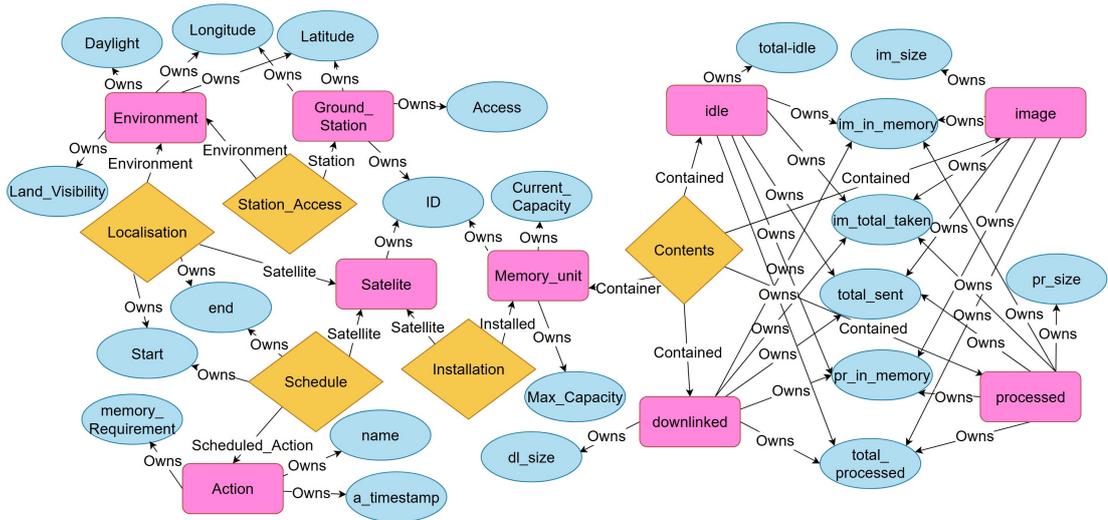


Figure 4.1: KG schema created, displaying an overview of the satellite scheduling problem at time instance i with entities as pink rectangles, relations as orange diamonds, and the attributes as blue ovals.

149661 attributes. These include overlapped variables such as the same memory values at different time intervals, containing overall memory, images, and processed image in memory, as well as all downlinked related activities.

Therefore, after the schema was created, the scheduled data was migrated to the KG followed by code executions to explore how it may be used for satellite scheduling and how the data may be retrieved.

4.4.2 Retrieval of data from the schema

Various codes were created and executed to analyse and test key aspects of the KG of all the components. These codes focused on including entity retrieval, complex pattern matching and relationship traversal, enabling in-depth exploration to answer queries or concerns on the stored information. The data retrieved from the KG may be used for, but not limited to:

- Retrieval of actions for specific time instances i and time ranges.
- Exploring the memory usage trends with respect to time i .
- Assessing the visibility of ground stations with coordinates between specified time

intervals.

- Assessing shared data between the interconnected properties across different times.
- Evaluating the relationships between different scheduling activities

Tables 4.2 and 4.3 represent types of queries that could be asked of the system, and their corresponding KG based solutions. The queries and codes created were manually done to initially assess the possibilities of whether the queries can be answered through code.

In Table 4.2, the first query involves one of the fundamental questions around retrieving an action for a particular time, this four line code, looks directly for the time $\$at$ and the action $\$a$ with its attributes *name* ($\$n$), and action time stamp ($a_timestamp, \$at$) followed by the retrieval of the name of the scheduled action. In response, this generated the action scheduled to be processed. Note: The symbol '\$' represents a variable which is a requirement for the correct syntax in TypeDB.

To assess the memory profile (the second question), all actions contained within the *Contents* of the *Memory_unit* were called with all other interconnected properties to retrieve all data related to the queried time. The generated KG in Figure 4.2 shows an image taken from typeDB studio, where the memory profile of the action *process* overlaps with itself, where the images in memory were the same. It can be seen that the contents only represent the memory state and the relationships within the memory itself, where the connections are indirectly linked to the scheduled time, but the status of the memory profile.

The third question inquires about a time range, two environmental conditions, and its coordinates. The code, similar to that of question 2, now has a time range for start time $\$st$ followed by a different 'get' statement while keeping all the relationships defined for the schema the same. Shown in Figure 4.3 contains three images taken from TypeDB studio; Figure 4.3a shows the full results representing the daylight and ground station conditions to be true, while land visibility for 3 instances is *False* and the other *True* with their coordinates at each time instance. Two zoomed images, Figures 4.3b and 4.3c contains the details showing land visibility to be *False* with 3 branches

Table 4.2: Example 1 for KG code for types of queries

Number	Query	Code - Solution
1	What action was scheduled at time 05 : 41 : 46?	<pre> match \$at = 2020-12-03T05:41:46; \$a isa action, has name \$n, has a_timestamp \$at; get \$n; </pre>
2	What action is scheduled at time 01:23:21, and what other action did the memory profile overlap with?	<pre> match \$st = 2020-12-03T01:23:21; \$a isa action, has name \$n, has a_timestamp \$at; \$ret isa \$ret-type; {\$n contains 'processed'; \$x isa processed, has \$ret;} or {\$n contains 'image'; \$x isa image, has \$ret;} or {\$n contains 'idle'; \$x isa idle, has \$ret;} or {\$n contains 'downlink'; \$x isa downlinked, has \$ret;}; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$sat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$t(\$a,\$sat) isa schedule; \$l(\$mem,\$sat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$sat,\$env) isa localisation, has start \$st, has end \$en; get \$n, \$at, \$ret, \$pt, \$ac, \$lv, \$d, \$lat, \$lon, \$l, \$sa, \$t, \$loc, \$id, \$mem, \$g, \$cc, \$st, \$en; </pre>
3	Was the ground station accessible and land visible between times 20 : 48 : 46 and 20 : 49 : 01? Also, what were the coordinates?	<pre> match \$st >= 2020-12-03T20:48:46; \$st <= 2020-12-03T20:49:01; \$a isa action, has name \$n, has a_timestamp \$at; \$ret isa \$ret-type; {\$n contains 'processed'; \$x isa processed, has \$ret;} or {\$n contains 'image'; \$x isa image, has \$ret;} or {\$n contains 'idle'; \$x isa idle, has \$ret;} or {\$n contains 'downlink'; \$x isa downlinked, has \$ret;}; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$sat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$l(\$mem,\$sat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$sat,\$env) isa localisation, has start \$st, has end \$en; get \$ac,\$lv,\$d,\$lat,\$lon,\$loc,\$env,\$st,\$en,\$g; </pre>

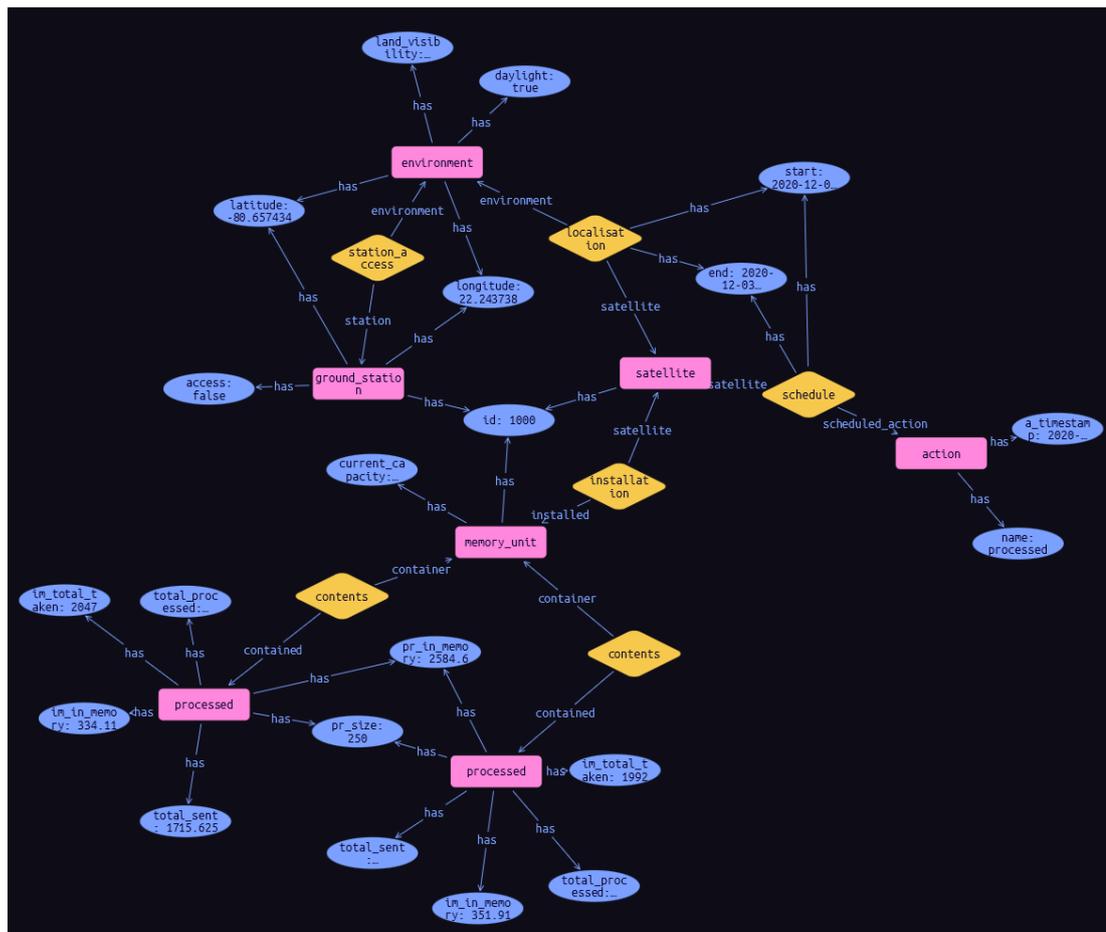
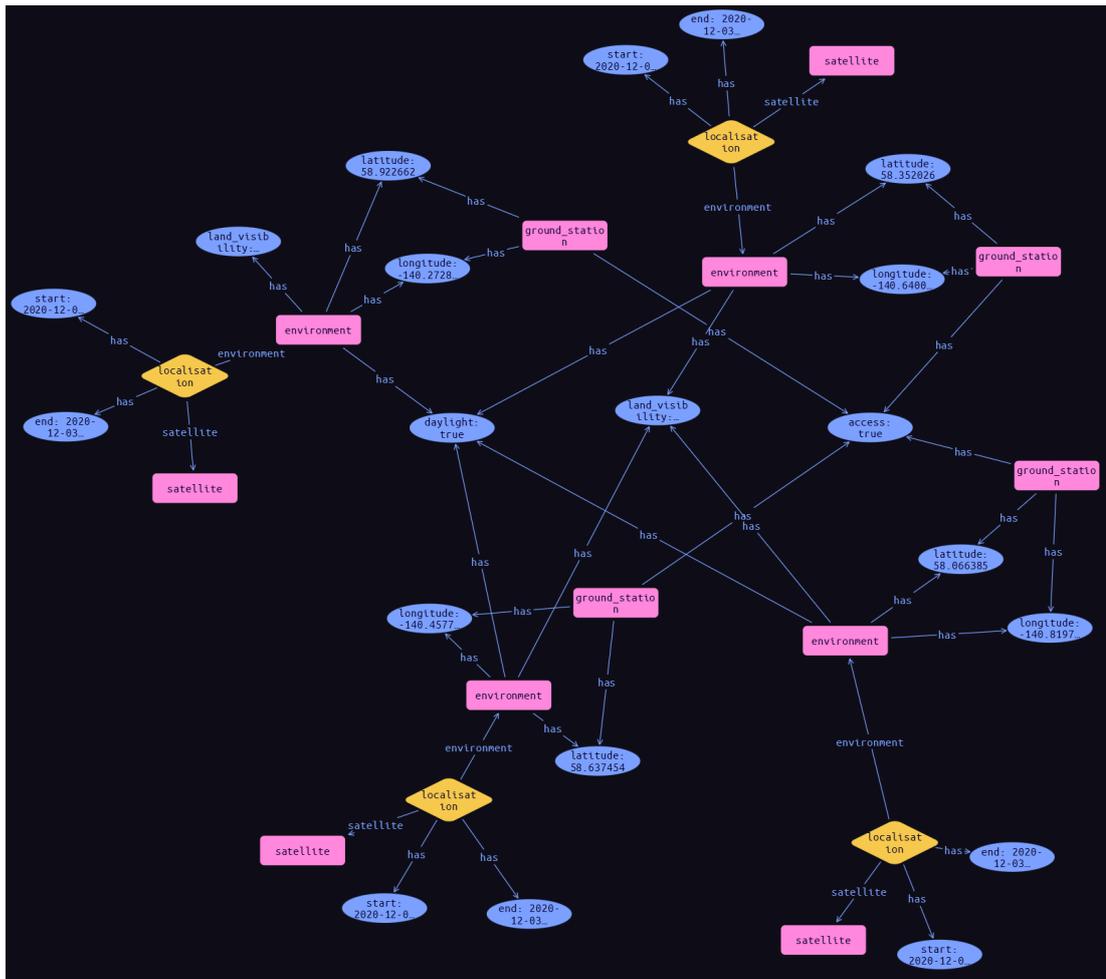


Figure 4.2: KG displaying the overlapping of actions with respect to the memory profile taken from typeDB studio for question 2.

pointing to the respective times 20 : 48 : 51, 20 : 48 : 56, and 20 : 49 : 01; and land visibility as *True* for time 20 : 48 : 46.

Similarly shown in Table 4.3 are two further queries with their respective code examples. Question 4 inquires about the overall memory and the actions scheduled during those times, like question 3 in Table 4.2, a range is provided in the code; however, a new value (*?value1*) is created and used for the lower range of ‘*Current_Capacity*’ (*\$cc*) with ‘*\$max*’ as the upper range (as max is the 80% limit of the overall onboard memory). Based on what the user would like to see would determine the retrieve function, therefore *Option 1* provides only the actions with their respective time stamps as requested by the question shown in Figure 4.4a, while *Option 2* provides more details

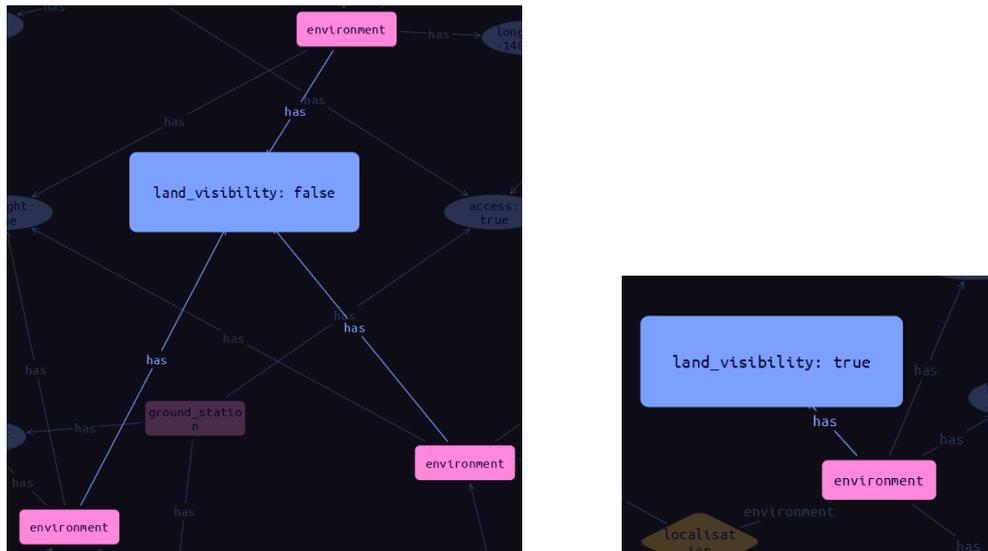


(a) KG displaying the environmental factors with the satellite's coordinates.

Figure 4.3: KG displaying the detailed environmental factors with the satellite's coordinates for question 3. (Figures *b* and *c* are on the next page)

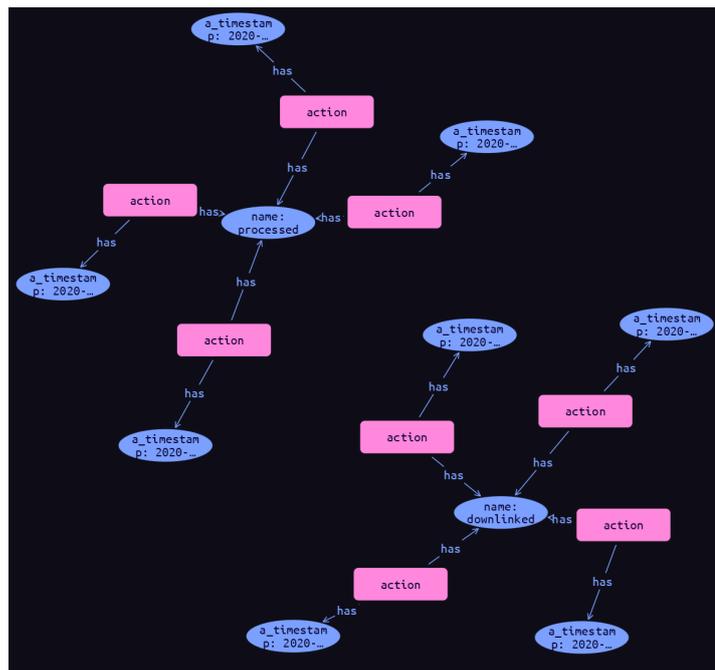
on the interconnected properties for the solution, enabling the user to see what action and respective time is connected to the specified memory with its value. This generated a much larger graph, as shown in Figure 4.4b, showing two separate sets of interconnected properties. It was observed that two actions were scheduled: downlinking and processing, each occurring at 4 times (21:58:41, 21:58:46, 21:58:51, and 21:58:56) for downlinking and (21:58:21, 21:58:26, 21:58:31, and 21:58:36) for processing.

Finally, question 5 inquires about the number of instances the downlinking has occurred within a specified time range and what was the onboard memory $\$cc$ at those



(b) KG highlighting the land visibility state where False. (c) KG highlighting the land visibility state where True.

Figure 4.3: KG displaying the detailed environmental factors with the satellite's coordinates for question 3. (Figure *a* is on the previous page)



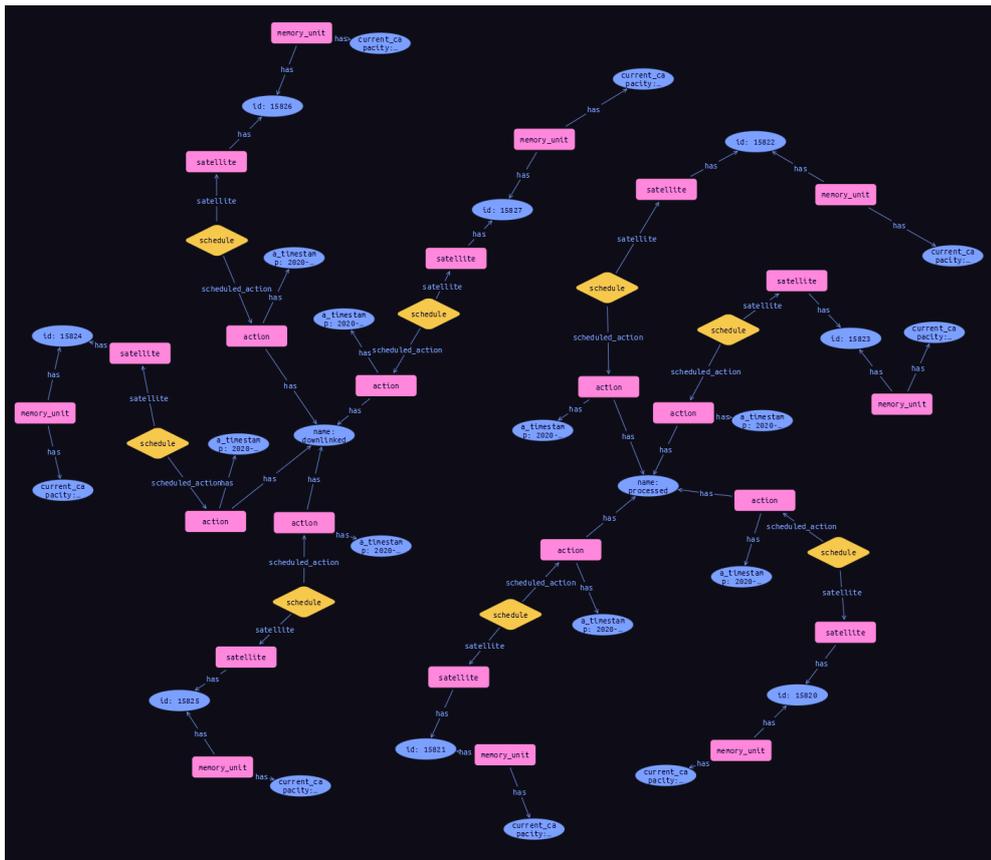
(a) KG representing Option 1 results for question 4

Figure 4.4: KG showing result options for question 4. (Figure *b* on the next page)

Table 4.3: Example 2 for KG code for types of queries

Number	Query	Code - Solution
4	Between times 21 : 58 : 21 and 21 : 59 : 00 at what times was the memory within 79% – 80% capacity (1896 – 1920 GB) and what were the scheduled actions?	<pre> match \$st >= 2020-12-03T21:58:21; \$st <= 2020-12-03T21:59:00; \$a isa action, has name \$n, has a_timestamp \$at; \$ret isa \$ret-type; {\$n contains 'processed'; \$x isa processed, has \$ret;} or {\$n contains 'image'; \$x isa image, has \$ret;} or {\$n contains 'idle'; \$x isa idle, has \$ret;} or {\$n contains 'downlink'; \$x isa downlinked, has \$ret;}; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$sat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; ?value1 = 1896000; \$cc >= ?value1; \$cc <= \$max; \$t(\$a,\$sat) isa schedule; \$l(\$mem,\$sat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$sat,\$env) isa localisation, has start \$st, has end \$en; Option 1: get \$at,\$a, \$n; Option 2: get \$at,\$a, \$n, \$cc, \$mem, \$id, \$sat, \$t; </pre>
5	How many instances of downlinking occurred between times 20:32:00 and 20:33:50 and what was the onboard memory?	<pre> match \$at >= 2020-12-03T20:32:50; \$at <= 2020-12-03T20:33:50; \$a isa action, has name \$n, has a_timestamp \$at; \$n == 'downlinked'; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$sat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$t(\$a,\$sat) isa schedule; \$l(\$mem,\$sat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$sat,\$env) isa localisation, has start \$st, has end \$en; get \$at,\$a, \$cc,\$mem,\$id,\$sat,\$t,\$n; </pre>

times. Only *'downlinked'* was inquired with the respective time range in this case. Figure 4.5 provides a KG representation of downlinking occurring 3 times, represented as branches (20 : 33 : 01, 20 : 33 : 11, and 20 : 33 : 21), showing the primary interconnected properties.



(b) KG representing Option 2 results for question 4.

Figure 4.4: KG showing result options for question 4. (Figure *a* on the previous page)

The results of the executed queries are tabulated and presented in Table 4.4 to facilitate comparison and validation of the data and graph with the original schedule.

The overall data accurately reflects the results retrieved from the KGs generated by the executed queries. Through careful comparison of the original data from the schedule with the generated results, the information from each question, along with the code for each KG, was aligned with expectations. This, therefore, means the capabilities of the KG for the scheduling data, including constraints such as memory and environmental relations, have been validated. The graphs visually reinforce the consistency between tabular data, query execution, and the migrated satellite schedule data.

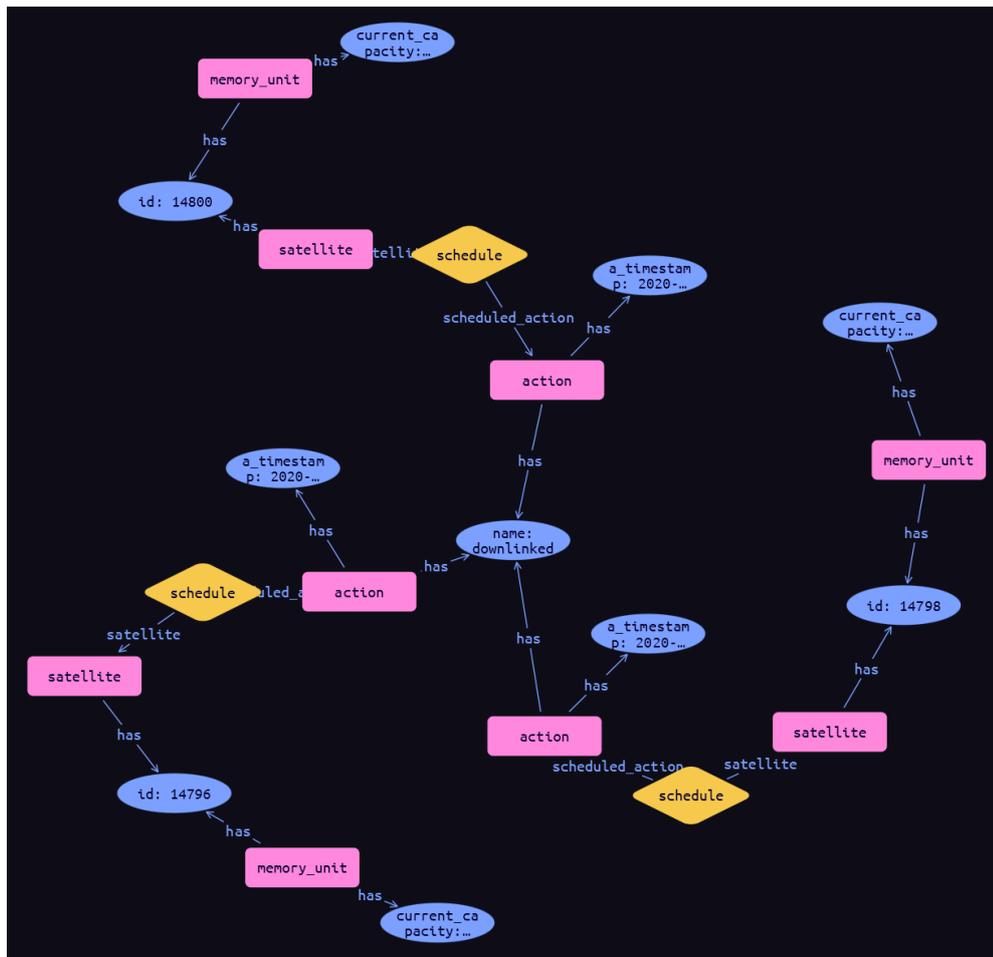


Figure 4.5: KG representation for the downlinking action and onboard memory for question 5.

Table 4.4: Tabular results for the execution of codes for questions 1, 2 (Figure 4.2), 3 (Figure 4.3), 4 (Figure 4.4), and 5 (Figure 4.5)

Number		Results for each question												
1	name	processed												
2	name	processed	a_timestamp	2020-12-03 01:23:21	pr_size	250	contents	contents	access	False	land_visibility	True	daylight	True
	latitude	-80.657434	longitude	22.243738	installation	installation	station_access	station_access	schedule	schedule	localisation	localisation	id	1000
	memory_unit	-	ground_station	-	current_capacity	-	start	-	end	-	total_processed	-	im_in_memory	-
	memory_unit	memory_unit	ground_station	ground_station	1536886.0	1536886.0	2020-12-03 01:23:21	2020-12-03 01:23:26	1883.091518	351.91	1956.008185	334.11		
	im_total_taken	1992	pr_in_memory	2584.6	total_sent	1642.708333								
	access	2047	land_visibility	2584.6	daylight	1715.625								
3	access	True	land_visibility	False	ground_station	ground_station	latitude	58.352026	longitude	-140.640047	localisation	localisation	environment	environment
	start	2020-12-03 20:48:56	end	2020-12-03 20:49:01										
4	a_timestamp	2020-12-03 21:58:21	action	action	name	processed								
		2020-12-03 21:58:26		action		processed								
		2020-12-03 21:58:31		action		processed								
		2020-12-03 21:58:36		action		processed								
		2020-12-03 21:58:41		action		downlinked								
		2020-12-03 21:58:46		action		downlinked								
		2020-12-03 21:58:51		action		downlinked								
		2020-12-03 21:58:56		action		downlinked								
5	a_timestamp	2020-12-03 20:33:01	action	action	current_capacity	1790722.0	memory_unit	memory_unit	id	14796	satellite	satellite	schedule	schedule
		2020-12-03 20:33:11		action	1788172.0	1788172.0	memory_unit	memory_unit	14798	satellite	satellite	schedule	schedule	
		2020-12-03 20:33:21		action	1785622.0	1785622.0	memory_unit	memory_unit	14800	satellite	satellite	schedule	schedule	
	name	downloaded												
		downloaded												
		downloaded												

4.5 Conclusion

This section explored the existing research and capabilities of KGs to determine the potential use case for satellite schedules and the benefits it may bring. The study demonstrated the capabilities of using KGs for satellite scheduling data and highlighted its ability to show meaningful insights from complex relationships through interconnected properties of nodes such as entities, relationships, and attributes. The example schedule used proved the capacity to map, display and report at least hundreds of thousands of data points, which would greatly benefit real-world investigations of satellite schedule data and displaying the interconnection nature of different resources and mission objectives or actions.

Through exploring the KG schema, example codes were tested and validated for memory trends and visibility conditions, showcasing the KG's effectiveness in structuring and accessing relevant information. This would enable operators to pinpoint particular times or data points of interest and construct adaptive maps to review necessary details, as well as being able to combine this with additional KGs to compare the results from other schedules to identify patterns of behaviour or potential issues to investigate.

The integration of tabular data with graphical representation further confirmed the reliability of the retrieval process, which supplements the graphical output well when presenting a very high number of nodes or data points. This enables KGs the flexibility to connect with external systems through different information formats and meet the needs of operators and any systems in use.

Chapter 5

Explainability - Comparative Analysis and Argumentation

5.1 Chapter Overview

This chapter represents the application of SCA, the concept of single instance comparative analysis introduced here, and Argumentation for XAI to simple satellite scheduling. Two variations of satellite scheduling are explored here, one outlined in Chapter 3 and the other described in this chapter is derived through a RL technique by [322]. This chapter is separated into three main sections:

- Chapter 5.2 for SCA is based on the publication “*Towards explainability of on-board satellite scheduling for end-user interactions*” from the proceedings of the International Astronautical Congress (IAC) 2021 [37]. It contains three sections:
 - The Literature Review discusses the comparisons of data formats across different scheduled data and how its use can be useful in contributing to XAI.
 - The methodology outlines how the comparisons between data are formulated.
 - Lastly, the results demonstrate its use case for comparing schedule data.
- Chapter 5.3 details the use of Computational Argumentation Techniques, where

AA is applied to the simple satellite schedule outlined in *Abstract argumentation for explainable satellite scheduling* published in IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA) 2022 [39] [peer-reviewed], and “*Explaining AI Decisions in Autonomous Satellite Scheduling via Computational Argumentation*” published in the proceedings of SPAICE 2024 [42] that investigates TAF application to the RL derived schedule [peer-reviewed].

This is made of three sections:

- The Literature Review is an expansion on Argumentation techniques where principles were applied in this thesis.
 - Exchange of properties for the Abstract Argumentation Framework (AAF), where two concepts SEP and PEP were applied to the satellite schedule derived in Chapter 3.3 followed by a discussion of the results.
 - Following AAF previously mentioned, TAF is then used but applied to the expanded satellite scheduling problem initially introduced in Chapter 3.3.1, upon which results are discussed.
- Chapter 5.4 summarises the findings of the chapter, concluding what has been learnt and any limitations or considerations identified through the experiments.

5.2 Singular Comparative Analysis

This section briefly explores the use of SCA to assess and compare the variances between two satellite schedules based on the heuristic approach, and the schedule generated by the solver explained in Chapter 3.

5.2.1 Literature Review

Satellite scheduling involves the allocation of limited resources considering constraints such as communication windows, power consumption, and orbital constraints stated in the Literature Chapter 2.2.4 and established in Chapter 3, where efficiency is crucial in maximising task allocation. The maximisation of resource utilisation has naturally

led to the investigation of automated capabilities, which could efficiently and rapidly manage satellite operations without GSO supervision or interaction, as examined in studies into satellite constellations and intelligent satellite operations [323–325].

However, the ability to prove that automated systems produce more effective and efficient results than manual methodologies has been an area of research recently, with investigations comparing the autonomous capabilities in generating design documentation and high school essay scoring [326,327]. These studies utilise comparative analysis, which is the formal practice of comparing the differences in performance between two or more approaches to a problem or area of study, used to derive explanations from the similarities and separations observed [328].

The approach of comparative analysis has been applied to compare results to scheduling problems in cloud computing [329,330], energy management systems [331], and EO satellites [332] utilising graphical, tabular and textual representations of the comparisons to observe the results.

A single study was identified to conduct a comparative analysis for satellite scheduling, assessing different RL approaches to producing schedules [332]. However, the study did not explore the explainability capabilities of comparative analysis, instead focusing only on the performance capabilities of the RL models, therefore highlighting a gap in existing research scopes.

The explanation capabilities of comparative analysis aim to enable insight between differing schedules, which can be conducted on singular time instances or across time horizons, by exposing distinctions and the benefit or disadvantage of making different decisions [328]. This can be investigated to aid explainability and interpretability of an automated system, utilising the principles of eXplanation User Interface (XUI) as explored in a recent study to help users better engage with and understand automated systems [333].

5.2.2 Methodology

SCA, applied across different satellite schedules for a single time instance, focuses on four main factors through mathematical expressions that examine which schedule is

more efficient under the constraints criteria. The comparison may be between two decisions at a single moment in time or an overall comparison. To compare the four factors of total images taken, processed instances, idle time, and total downlink instances, three equations were derived in support of generating explanations.

5.2.2.1 Comparison of the number of Images

Image acquisition for this satellite scheduling problem is key to achieving the mission objective. The number of images captured indicates the efficiency of utilising the observation windows. Equation 5.1 provides the comparison between two schedules where the number of images from schedule 1 ($X^{(1)}$) is compared with schedule 2 ($X^{(2)}$).

$$\sum_{i=1}^T X_{i,a_p}^{(1)} - \sum_{i=1}^T X_{i,a_p}^{(2)} \geq 1 \quad (5.1)$$

The first schedule is preferred if the number of images from $X^{(1)}$ exceeds $X^{(2)}$; otherwise, Schedule 2 is favoured.

5.2.2.2 Comparison of Processed instances

Processing of images ensures data is prepared before downlinked to the ground station. This stage is known to be the bottleneck as the downlinking is unable to proceed without any processed images. Due to the constraints, processing may occur at any time once there is enough onboard memory. Thus to assess and compare the differences between the two schedules, similarly to Equation 5.1, Equation 5.2 reflects the process in achieving the comparison.

$$\sum_{i=1}^T X_{i,a_r}^{(1)} - \sum_{i=1}^T X_{i,a_r}^{(2)} \geq 1 \quad (5.2)$$

5.2.2.3 Comparison of Idle instances

As stated previously in chapter 3, idle time is a representation of no other actions scheduled, which leads to an underutilization of the satellite's resources where an executable action may be possible. Therefore, an optimised schedule would minimise idle instances for operational efficiency. Equation 5.3 compares two schedules where the lower value indicates a better performance.

$$\sum_{i=1}^T \sum_{a_e}^A X_{i,a_e}^{(1)} - \sum_{i=1}^T \sum_{a_e}^A X_{i,a_e}^{(2)} \geq 1 \quad (5.3)$$

Using the objective function 3.7, the fewer idle instances the better; therefore, if Schedule 1 has less idle time, it is more feasible over the other. Likewise, if Schedule 2 has fewer idle instances than Schedule 1, this means Schedule 2 is preferred.

5.2.2.4 Comparison of Downlink Instances

Downlinking transmits data to the ground station when there is a line of sight, thus freeing memory onboard and increasing the capacity for other actions to be scheduled. A higher frequency of downlink events would be required to improve data throughput. Equation 5.4 provides the comparison of the downlinking events between two schedules.

$$\sum_{i=1}^T X_{i,a_d}^{(1)} - \sum_{i=1}^T X_{i,a_d}^{(2)} \geq 1 \quad (5.4)$$

Therefore, a schedule with a higher number of downlinks represents a better utilisation of the available opportunities.

By combining the four performance metrics, the quality of two schedules was assessed, reflecting the instantaneous results through a visual and graphical representation aimed towards developing a user interface to enable intuitive comparison, enhanced interpretability, and informed decision-making for scheduling optimisation.

5.2.3 Results

Two schedules were created under identical operation constraints (Chapter 3), where a comparative framework was developed to analyse the schedules based on their total number of images, idle instances, and downlink activities using the equations from Chapter 5.2.2. As shown in Chapter 3, there were significant differences and improvements made to the original schedule, which need to be interpreted and presented to an end-user to inform them of which schedule is closer to an optimal solution.

A proposed solution for representing these differences is shown in Figure 5.1, where a Gantt chart with potential operational decision support visualisation is presented that can be automatically generated. The user would be allowed to select any point on the schedule chart, and an immediate comparison would take place between the two schedule options, including textual statements and data to the user summarising the states of all actions and memory status at that selected time instance. The panel to the right of the figure provides the selected action and time along with the memory status, while the left panel contains the summary of schedule status (whether optimal or only feasible), the memory changes as a result of the scheduled action (in this example process_image), whether any instrument failure or memory saturation has occurred, and the next possible opportunity to downlink.

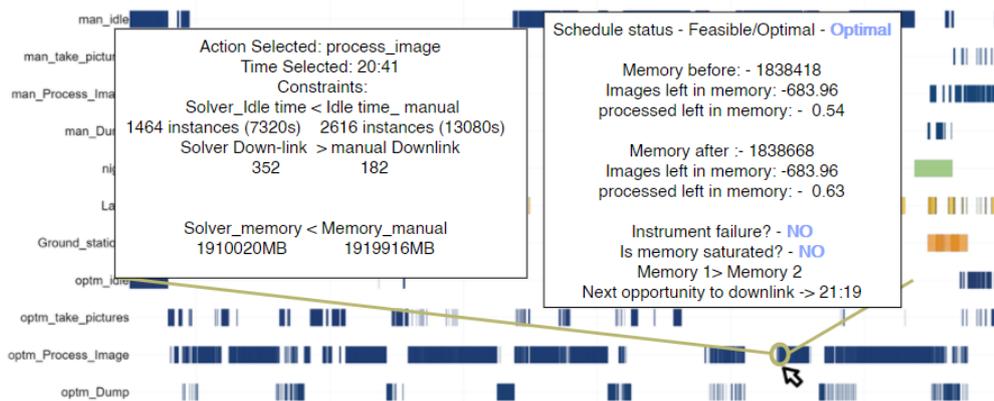


Figure 5.1: End User Interactions

The process of determining the best choice at any given point in the schedule outlined through this analysis of two different schedules led to the consideration of argu-

mentation as a means of performing deeper analysis and insight into which decision may be better or best.

5.3 Argumentation

This section explores the application of Argumentation techniques to two variations of schedules. Argumentation, as defined in Chapter 2.4.4, provides a structural approach to evaluating and explaining scheduling decisions. This technique is used to establish the reasoning behind the choices made, making them more transparent, interpretable, and justifiable. The schedule types and argumentation techniques applied are:

- Simple satellite schedule as defined in Chapter 3 with AA.
- A satellite schedule derived by a RL technique outlined in this chapter with the application of a TAF

5.3.1 Literature Review

As mentioned in Chapter 2.4.4, argumentation is the principle of analysing a decision or outcome to determine what influenced, whether positive, negative, or neutral, comprised of mathematical models [334] that have been defined around the objectives, containing elements of the problem presented as an Argumentation Framework (AF) [335]. An AF can be shown in different forms of logic, which facilitates both a graphical and quantifiable representation of the decision-making process, allowing for the examination of reasoning logic and is naturally effective as a means of explaining the reasoning for one choosing one activity over another [334–338].

The arguments within an AF can have the conditions of their acceptability grouped together as sets, the most commonly used groups being cores and remainders. The set of cores for an argument is the set of arguments required to be accepted to satisfy the original argument. Conversely, the set of remainders for an argument is the arguments required to reject the original argument [339].

The detected conflict relations, known as AA, can influence the conditions to either be improved or changed depending on the properties of the system. This can therefore

help explain how the conditions are affected [340], and thus an AAF has been stated to be created as pairs, containing both arguments and binary relations, where a relation is known as an attack [334–336, 341–343]. Rago et al. also stated standard labelling for attacks being ‘−’, with supports labelled as ‘+’, and neutral relations labelled as ‘0’ [344].

In addition to the AAF, there are several other types of AF, some of which are: Bipolar Argumentation Framework (BAF) - where attack and support relations occur [334, 343, 344] with two extensions: Quantitative Bipolar Argumentation Framework (QBAF) and Probabilistic Bipolar Argumentation Framework (PBAF) [334]; Structured Argumentation Framework (SAF) - where constraints are introduced in the form of preferences between arguments in accordance with the order of these arguments [334, 345]; and Abstract Dialectical Framework (ADF) that specifies the exact conditions by which an argument is accepted (or rejected) through the linking of dependencies, depicted as a directed graph [346]; and Incomplete Argumentation Framework (iAF) which introduces the possibility of unknown factors influencing arguments [347]; and TAF - where attack, support, and neutral relations occur to enable interactive recommendations, which allows for more complex and relatable argument structures [344, 348]. These relations were extracted to address the logical requirements based on how the predicted ratings for items and aspects impact each other. This synchronisation, in the Modgil et al. study [345], between the Recommender System (RS) and the explanations was considered an advantage with their approach, as users can trust explanations that detail how recommendations were created. Argumentative explanations involve, but are not limited to, conversational and visual explanations. These explanations form the foundation for interactions with users to explain recommendations and gain feedback, which can be accounted for in the RS to enhance its behaviour. Therefore, not only were the explanations varied and diverse, they also enable (to a limited degree) adaptable recommendation responses through extended use [344].

Furthermore Rago et al. [344] mathematically defined TAF as:

A tuple (X, L^-, L^+, L^0) where X is a set of arguments, and L^-, L^+, L^0

are binary relations over X . For $x, y \in X$, we say that x attacks y if $(x, y) \in L^-$, x supports y if $(x, y) \in L^+$, and x neutralises y if $(x, y) \in L^0$. With x as any of $-, +$ or 0 , for any $x \in X$, we will use $L^x(x)$ to denote $\{y \in X \mid (y, x) \in L^x\}$ the attackers, supporters or neutralisers, resp., of x .

This means whenever there is a conflict between x , and y (a set of arguments X), L is a binary relation produced based on whether there is an attack or support. To date, and to the best of the authors' knowledge, there are no applications of AA to real-life engineering problems, with several PhD research projects investigating the potential use cases of argumentation for XAI. [349] enriched AFs, by adding an additional support relation, to enable greater interpretability in visual explanations. Another project applied abductive explanations to sample instances, instead of the complete feature space, with the aim to improve the correctness and validity of generated explanations [350]. While the implementation for autonomously generated AAF-based view of a ML model's processes has also been explored, to assist with understanding reasoning and improving the model's performance [351].

The application of AAF allows for the examination of supporting and attacking conditions to an action within a schedule; however, to understand the overall impact of this or any alternative decision requires a calculation of the exchange of properties [352]. The exchange can be applied to a single exchange (SEP), where any one critical job is exchanged with another at any time within the schedule, or as a multiple exchange (PEP) in which any number of critical jobs can be exchanged [352]. The exchange of jobs has been applied to optimisation problems in scheduling, whether to reduce the makespan or increase the output, as studied for variations of the flow shop and job shop problems such as Mixed No-Idle Flow-shop Scheduling Problem (MNIFSP) [353], Distributed Job Shop Scheduling Problem (DJSSP) [354], and Distributed Permutation Flow Shop Scheduling Problem (DPFSP) [355].

To assess the XAI capabilities of AAF through the calculations of SEP and PEP for the EO satellite schedule defined in Chapter 3, the following questions were created as a benchmark for the scheduling system to answer:

- Is it better to interrupt sequences of repeated actions or maintain the sequence?

For example, should processing occur in the middle of an image-taking sequence or after?

- Is the time critical for task ‘x’, or is task ‘x’ critical for a certain time?
- Does the onboard memory impact the criticality of a task?
- Why execute an action over another, for example, take images vs down-linking vs processing on-board?
- Can the memory limit be relaxed for critical actions?
- If the conditions have been changed, how different will the revised schedule be?

5.3.2 Exchange of Properties

Two applications were investigated in assisting the generation of an AAF for satellite scheduling. These were SEP and PEP that have been adapted from Cyras et al, as mentioned in Chapter 5.3.1. [352] research that focused on minimising the last machine completion time to determine the minimum makespan of a schedule and the application to a nurse roster. SEP is defined as a single exchange of any critical job with another job between machines throughout a schedule to improve the given schedule. PEP, however, entails an exchange of multiple critical jobs with other jobs between machines to gain improvements. Equation 5.5 was created to enable both concepts by Cyras et al. scheduling problem [352] and the satellite schedule defined in Chapter 3.

$$v^* = \min_{a \in A} v_a \quad (5.5)$$

The memory value required by any action a (v_a) during any time instance results in the smallest memory value v^* by the considered scheduled action (as there is only one action requirement per instance). Due to a_e having no value, unless otherwise undergoes an attack from another action through SEP or undergoes a PEP scenario where it is exchanged with another action from another time interval. This means the smallest value is the value scheduled at that time instance to determine if an exchange due to an attack is possible based on the following definitions.

Definition 5.3.1 *An action $a \in A$ is said to be critical at time interval $i \in \{1, \dots, T\}$ if:*

$$X_{i,a} = 1 \wedge M_{max} - v^* \leq m_i \leq M_{max}$$

This means that an action is considered critical to a specified instance, where if replaced with any other action and propagated throughout the schedule would violate the onboard maximum memory constraint.

Definition 5.3.2 *SEP is satisfied by a schedule S iff for every critical action $a \in A$ instance i , it holds that, for any $a' \neq a$*

$$m_i - m'_i > v_{a'} - v_a \quad \wedge \quad m'_j \leq M_{max}, \forall j > i$$

Where m'_i and m'_j represent the value of memory at their respective time instances i and j , respectively, if action a' is chosen instead of action a at time instance i . Additionally, when an action v_a has been replaced by any of the other three actions $v_{a'}$, an alternate memory m'_i is created at that instance, and therefore, an alternate memory m'_j following this action is generated throughout the schedule, and while within M_{max} will satisfy SEP.

Definition 5.3.3 *PEP is satisfied by a schedule S iff for every critical action $a \in A$ at instance i , it holds that, for any $a' \neq a$, and any $j \neq i$:*

$$m_i - m'_j > v_{a'} - v_a \quad \wedge \quad m'_k < M_{max}, \forall k$$

Therefore, when an action v_a is exchanged with one of the three actions $v_{a'}$ at a different instance j , this results in two alternate memory values: m'_j reflects the memory state after the exchange and m'_k which is iteratively updated throughout each of the following actions to ensure M_{max} is satisfied, thereby satisfying PEP.

Overall, SEP ensures that substituting action a with a' at time interval i does not negatively impact the feasibility or criticality with respect to the memory constraint, for both at that instance and throughout subsequent time intervals. Similarly, PEP

guarantees that replacing a critical action with another does not violate the memory constraint across the entire time horizon.

In addition to SEP and PEP concept application from Cyras et al. [352], both negative and positive decision sets are defined as a subset of the decision space $T \times A$ (as previously defined in Equation 3.1 in Chapter 3.3), created using the action constraints from the satellite scheduling problem.

Definition 5.3.4 *The negative fixed decision set D^- is defined as:*

$$D^- = T^- \times A^- \subseteq T \times A$$

such that a feasible schedule S satisfies:

$$(i, a) \in D^- \rightarrow X_{i,a} = 0$$

This definition can also detect extreme scenarios such as memory saturation during time i with the next available downlink opportunity to occur at $j > i$. In this situation, the negative decision set is transformed into:

$$D^- = T^- \times A^- \cup \{(k, a) : i < k < j\}.$$

The positive fixed decisions can also be applied to this use case and may be actions implemented by the GSO in achieving an EO campaign goals.

Definition 5.3.5 *The positive fixed decision set D^+ is defined as:*

$$D^+ = T^+ \times A^+ \subseteq T \times A$$

such that a feasible schedule S satisfies:

$$(i, a) \in D^+ \rightarrow X_{i,a} = 1$$

Using these principles, a feasible schedule is derived from the existing schedule created by the solver in Chapter 3.

5.3.2.1 Abstract Argumentation (AA) with Exchange of Properties

In the concept of applying AA to satellite scheduling, where AA analyses the conflicts between two or more arguments [335, 336, 343], these arguments represent a particular reasoning of decisions within a schedule. The dynamic changes within a schedule due to AA occurring can influence the execution timing of scheduled activities, resulting in more arguments and attacks, creating a structured argument [356].

Integrating the principles SEP, PEP, and AA theory, an attack in this process is derived when an exchange occurs between two or more actions. The constraints created are therefore:

- Any action may attack another except itself.
- With SEP, the action a_e cannot attack another action but may be attacked by another action.
- With PEP, the action a_e can attack another action.
- For PEP, exchanges occur only between two distinct actions within the schedule, excluding self-substitutions.

Figure 5.2 represents an attack occurring on a scheduled action during a specific time period i with a current memory state of m_{i1} in schedule S . The individual memory values V_{ap} , V_{ar} , and V_{ad} with arrows coloured pink, blue and green, representing actions a_p , a_r , and a_d respectively. Using the memory constraints from Equation 3.6a to ensure there is no breach in M_{max} due to the change at m_{i1} , thereby complying with SEP. Consequently, an alternate memory m'_i is generated, cascading into future time steps updating the memory values of m'_j for instances $(m_{i2}, m_{i3}, m_{i4} \dots m_j)$ resulting in $(m'_{i2}, m'_{i3}, m'_{i4} \dots m'_j)$ stated in Definition 5.3.2.

To reiterate, when action a_e is scheduled for example, resulting in m_i , only actions a_p , a_r , and a_d containing V_{ap} , V_{ar} , and V_{ad} may attack this action as shown in Figure 5.2. Otherwise, in the condition where another action a_p , a_r , or a_d is scheduled, only any two of those actions may attack this action. For example, if a_r is scheduled, this will incur an attack from a_p , or a_d .

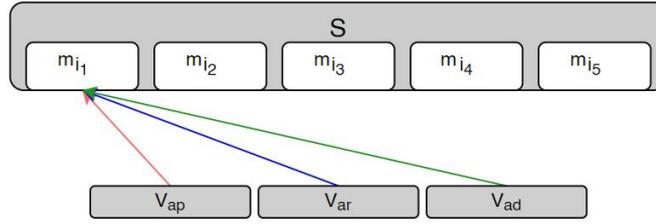


Figure 5.2: An overview of the argumentation process across one-time instance representing the occurrence of a single exchange of an action with memory variables V_{ap} , V_{ar} , and V_{ad} within a schedule S to alter the memory m_{i_1} at that instance.

However, for PEP, a representation shown in Figure 5.3 is an overview of how a pairwise exchange may occur in schedule S between two non-identical actions scheduled for execution at two different time intervals. The original memory states ($m_{i_2}, m_{i_3}, m_{i_4} \dots m_j$) remain unchanged before the action exchange, following which alternate memory states m'_j and m'_k defined in Definition 5.3.3, are generated on the condition M_{max} remains satisfied as stated in Equation 3.6b. As shown in the figure, what occurs during a pairwise exchange between actions within a Schedule S , impacting the existing memory $m_{i_1}, m_{i_2}, m_{i_3}, m_{i_4}$, and m_{i_5} , on-board a satellite by the respective memory values V_{ap}, V_{ar}, V_{ad} and V_{ae} of an action following an attack. This means following the interchange of the scheduled action a at i with another scheduled action will create the alternate memory m'_j ; the possibilities of exchanges are shown in colours pink, blue, green, and brown for values V_{ap}, V_{ar}, V_{ad} , and V_{ae} respectively. Therefore, an attack condition was created, leading to a decision $X_{i,a}$, the memory profile $m_j \forall j \geq i$, and the final objective function value.

When an action is considered to replace an existing scheduled action at a specified instance, it is known as a binary attack represented as (\rightsquigarrow). This occurs only when the satellite coordinates, position and onboard memory are satisfied, which will determine the action exchange feasibility. A representation of this exchange is shown in Figure 5.4 where action a_p if scheduled at i is explored for attacks by a_r, a_d and a_e ; whereby a_r is analysed for attacks from a_p, a_d and a_e ; a_d by a_p, a_r , and a_e ; and a_e by a_p, a_r , and a_d with all actions are shown shaded in grey circles.

Additionally, shown in Figure 5.4, there is a differentiation between negative and

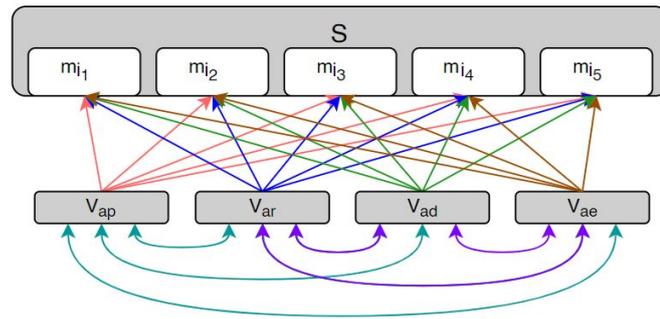


Figure 5.3: An overview of the argumentation process across each time interval representing the values (V_{ap} , V_{ar} , V_{ad} , and V_{ae}) of actions a_p , a_r , a_d , and a_e are exchanged with each other via bi-directional arrows in colours teal and purple, based on the action scheduled at i

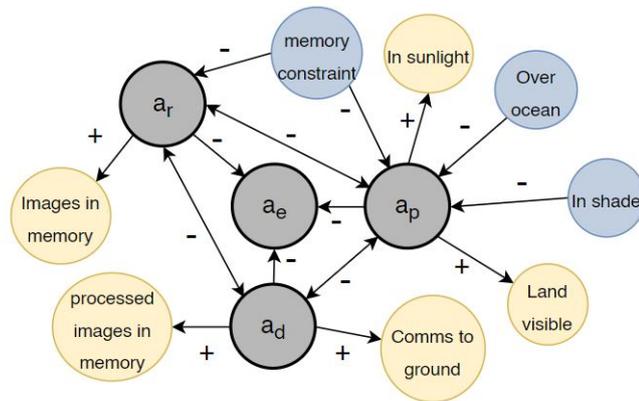


Figure 5.4: Conditions involved with attacks a_p (Image taking), a_r (Image Processing), a_d (Down-linking), and a_e (Idle time) during time i and the conditions affecting ‘-’ and supporting ‘+’ these actions from occurring.

positive arguments, where blue-shaded circles with arrows pointing away with negative ‘-’ symbols represent attack conditions that will hinder the execution of the action. Meanwhile, the arrows pointing towards the yellow shaded circles are considered supportive ‘+’ conditions that will encourage the action to be executed at the scheduled time due to the exception of the idle instance a_e that does not affect the onboard memory’s state as it has no value and is only scheduled when no other actions are feasible as stated in Chapter 3. This interplay of conflicts and support is derived from the BAF, as discussed in Chapter 5.3.1. The additional constraints, known as environmental factors, such as land visibility, ground station access, and sunlight exposure, dictate

which actions can be executed. Overall, this figure provides a structured visualisation of how external factors may influence the action execution in the schedule.

Combining the constraints with SEP and PEP, it can be noted, $a_{i,j}$ with the argument “An assignment of an action at instant j to time instant i ”, by applying the principles of AA to feasibility, the following definition is:

Definition 5.3.6 *The feasibility AF, $(Args_F, \rightsquigarrow_F)$ is defined as:*

- $Args_F = \{a_{i,j} : (i, j) \in (T \times A) \setminus D^-\}$
- $a_{i,j} \rightsquigarrow_F a_{k,l}$ iff $j \neq l, i = k$
- $a_{i,j} \rightsquigarrow_F a_{k,l}$ iff $m_q > M_{max}$ for $q > k$ if $k < l$, or for $q > l$ if $l < k$.

This, therefore, means that in addition to the action assignment within the decision matrix (first point), the second point indicates an action is competing for the same time slot due to the constraint of only one action allowed at each time interval. The third point signifies that an action challenges another only when the memory is breached, either if the duration k is insufficient for the action completion while it is already scheduled to be executed before time q or if the action is scheduled to complete before the allocated duration of k while it has occurred before q . In such scenarios, an alternate action would be preferred to prevent the memory breach and optimise time utilisation.

5.3.2.2 Results and Discussion

Utilising one day’s worth of data from the satellite schedule created in Chapter 3, the concepts of SEP and PEP were applied to evaluate their impact on memory constraints and feasibility. This section is separated into two sections SEP and PEP, with each presenting the analysis of how the framework can influence schedule optimisation, conflict resolution, and overall mission performance. These findings were discussed, showing the effectiveness in maintaining feasibility while complying with the operational constraints.

5.3.2.2.1 SEP

Each action scheduled was attacked by another action except itself and a_e . During the event of an attack, the memory profile was altered from that point onward, and therefore depending on the alternate memory with respect to the memory constraint at j' may be either feasible (in support of SEP) or infeasible (a violation of SEP) as stated in Definition 5.3.2.

A two-hour period was selected from the day (equating to 1440, 5-second interval data points), and shown in Figure 5.5 to produce a visual of a Gantt chart where actions (a_p , a_r , a_d , a_e) are scheduled. These actions were attacked by the other actions, and where feasible and infeasible are also shown. Each infeasible action exchange is a result of a breach in M_{max} due to the cascade of the effect of the attack, thus a violation of SEP. Additionally, action feasibility is also shown where all constraints were met and the exchange was possible. It can be noted that there were no a_d infeasible results during this period, meaning that, outside of the opportunities where it was possible, it was unable to attack due to no visible ground stations. This also means that during this period, whenever it had the opportunity to downlink, it did, as it had no other constraints apart from ground station visibility and therefore unsaturating the onboard memory.

This time period is also reflected in the shaded region of Figure 5.6, representing the total memory utilised to the right with the number of images retained in the memory to the left for a_p , and a_r and also the number of images in memory after a_d has been executed. At 21 : 04 hrs, an attack occurred where action a_r was replaced by a_p , highlighted in the blue shaded region, which led to a memory breach at m'_j marked by a broken vertical line intercepting m_{Max} (a horizontal green line). It was observed earlier in the schedule that memory approached saturation; however, downlinking (coloured red) immediately followed, reducing the stored images and preventing this breach from happening. Due to the downlinking action always incrementing, this variable resets after 3000 instances to improve the readability of the graph.

As the existing scale makes it difficult to see the variance following the attack leading to the breach, Figure 5.7 provides a magnified view highlighting the moment of

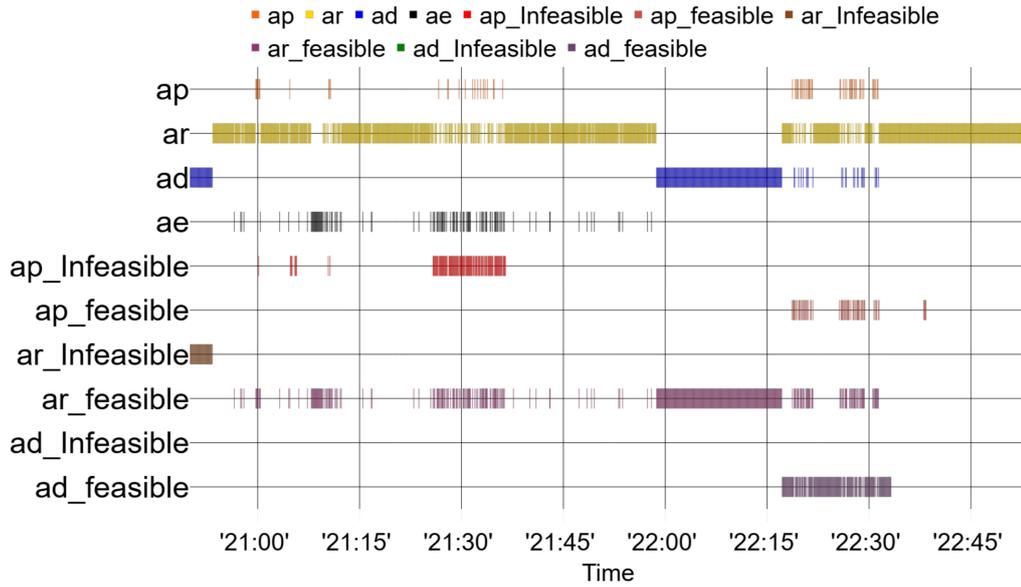


Figure 5.5: Summary of feasible and infeasible solutions when an action attack occurs on the original schedule for a_p , a_r , a_d , a_e

the attack at i . The infeasibility of action exchanges was quantified over the period of the same day against each scheduled action that was individually attacked (excluding itself). This was taken across 17278 data points with the infeasible attack percentages shown in Table 5.1 with the following observations.

- When a_r , a_d and a_e were attacked by a_p , this generated 74% (2897) infeasible cases with a_r , 16% (630) cases for a_d , and 6% (244) cases for a_e , out of the 3921 exchange possibilities with the remaining 4% returning feasible solutions.
- During the attack of a_p by a_r , all exchanges were possible resulting in a 0% infeasibility. When attacking a_d and a_e , however, infeasibility occurred 44% (1923) instances for a_d , and 10% (821) instances for a_e of the 4324 exchange possibilities remaining 37% feasible options.
- Lastly, when a_d attacked all actions, there were no infeasible solutions as the memory was always reduced by V_{ad} at every opportunity. This means an exchange was possible at any of the 2082 instance opportunities.

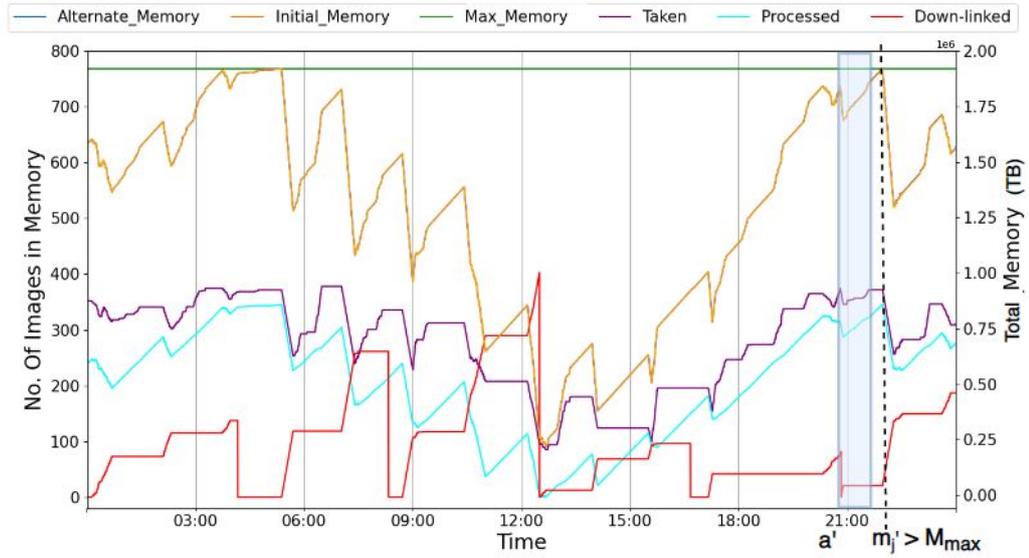


Figure 5.6: Results of a_p attack on a_r resulting in an alternate memory m'_i at time instance i at 21 : 04 hrs, followed by the updated memory m'_j exceeding the available memory on-board M_{max} shown as the horizontal green line on the right with all memory profiles referring to the axis on the right and action on the left.

Table 5.1: Percentages of action exchange infeasibilities over a period of 1 day

Action Exchanges	Scheduled Actions			
	a_p	a_r	a_d	a_e
	%	%	%	%
a_p	–	74	16	6
a_r	0	–	44	19
a_d	0	0	–	0

5.3.2.2.2 PEP

Further investigations proceeded with PEP to determine the impact on the feasibility of the same schedule. An $n \times m$ matrix of the schedule was created to allow the exchange of actions at different time intervals. This matrix is represented in Figure 5.8 covering a range of 70 seconds in 5-second intervals (times 02 : 12 : 31 – 02 : 13 : 41 hrs), taken from the day. With both time ranges represented in n rows by m columns, a grid of cells was created representing the actions (a_p , a_r , a_d , and a_e) as (0, 1, 2, –1) respectively. These actions would overlap, creating an attack of each action on the other for its respective time slot and triggering a pairwise exchange.

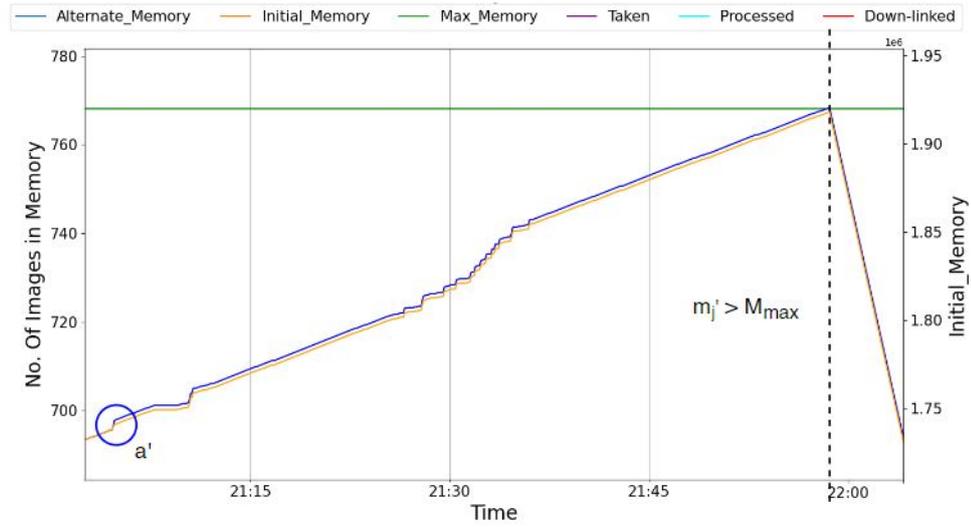


Figure 5.7: A magnified view of Fig. 5.6 showing where memory at time instance j' breached the maximum on-board memory M_{max} following an attack at instance i .

Where the overlapped actions were the same, they were shaded purple; however, where they were different, a feasibility check was performed at the first action's instance, generating a revised memory state, m'_j . The change in this instance was cascaded throughout, similarly to SEP, until the second action at the second time instance has been approached by the system on the condition that there has not been a breach in memory up to that point. Once the assessment of the memory of the second action was completed and deemed appropriate, this additional change was again carried throughout the rest of the schedule. When there was a breach in memory at any point and deemed infeasible, a violation in PEP occurred, and the cell was immediately shaded red, and the next exchange scan began. However, when the scan was successfully completed with both actions exchanged, this resulted in a cell shaded green, confirming feasibility satisfying PEP.

Shown in Figure 5.9, an extended time range (02 : 03 : 31 – 02 : 20 : 06) was selected, covering a total duration of 995 seconds encompassing the 70 second window shown in Figure 5.8. Within the expanded range, 20092 exchange opportunities of 5 second individual time instances were analysed. Among these instances, a total of 19509 involved an overlap of the same actions (shaded purple), 13286 infeasible exchanges (shaded

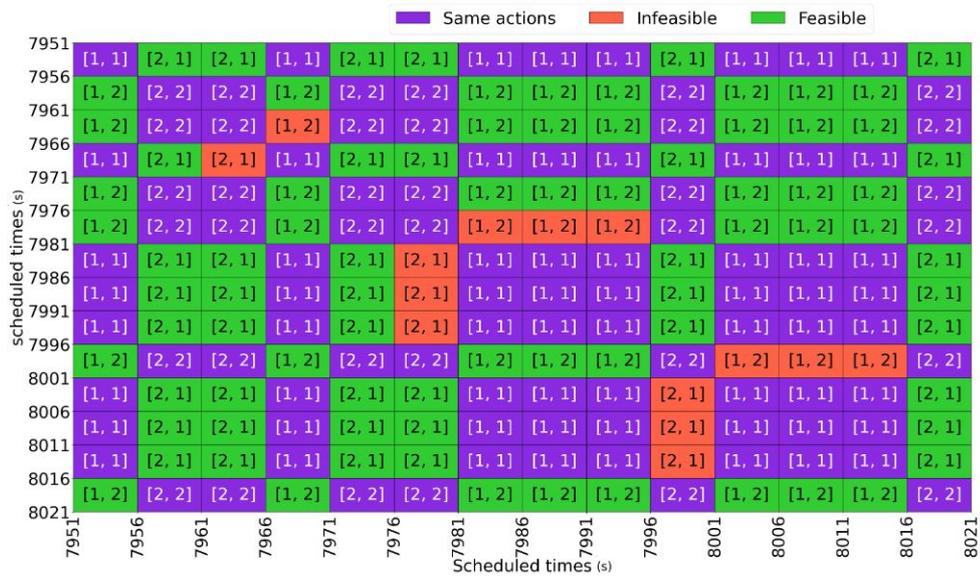


Figure 5.8: An example showing the implications of pairwise exchange occurring between any two actions within a 70 seconds window of a satellite schedule.

red) and 6806 feasible exchange opportunities (shaded green). It further shows the effectiveness and scalability of PEP in evaluating exchange opportunities. Additionally, it should be noted that the computation of these possibilities and graphical display was resource-heavy; therefore, consideration of available resources should be taken into account for replication.

These results were analysed based on the question raised in Chapter 5.3.1, and the findings have revealed the practical applications of PEP would be based on the hardware capabilities of the satellite as well as the level of autonomy allowed by the GSO. This flexibility could enhance and redefine the constraints of operation, especially for dynamic mission scenarios.

SEP and PEP applications provided insights into how alternative actions within a schedule could impact its feasibility and also the scenarios where this exchange might be necessary for changes in critical conditions. Though this study focused on EO satellite with a limited set of actions, the framework is extendable to more complex systems. XAI may be enhanced with the AA approach by:

- Providing a user interface with the generated schedule to the end user, allowing

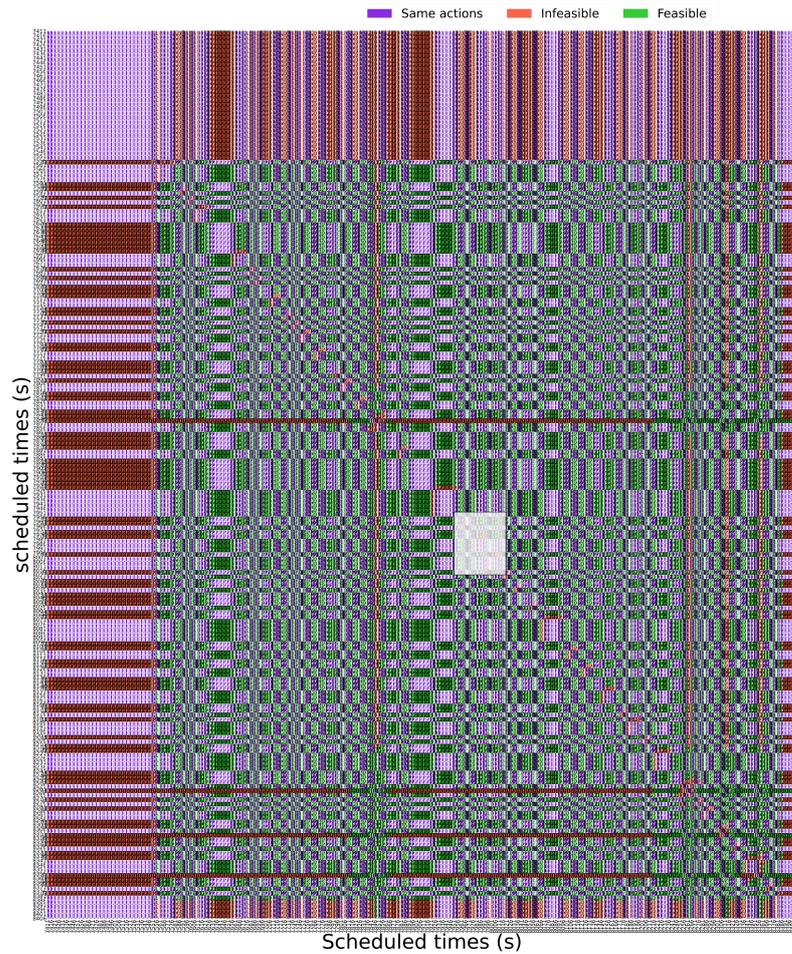


Figure 5.9: An example showing the implications of pairwise exchange occurring across a 16.58-minute window, highlighting where the 70-second window fits within.

them to interact and query the system.

- Providing computational results of the reasoning behind the generated schedule.
- Providing detailed conditions representing conflicts with the environment and conditions on board a satellite, as shown in Fig. 5.4.
- Providing an instantaneous reflection of the exchange properties of SEP and/or PEP, as shown in Fig. 5.2 and Fig. 5.3, based on the users queries.
- Lastly, generating a representation of a decision matrix enables the user to see the decisions that were influenced by the conditions.

Overall, AA serves as an effective layer in explainable scheduling, providing transparency in mission planning without compromising operational feasibility.

The application of AA for an EO satellite scheduling was investigated to achieve feasibility by enabling scheduled actions to be substituted by other actions in the form of relational attacks at different time intervals throughout the day. Each action within the schedule was subject to attacks that needed to satisfy the constraints. Following the attacks, they were immediately assessed and compared for their memory profile when SEP was applied. Meanwhile, when PEP was applied, this enabled an exchange of two actions at two independent time intervals on the condition the onboard memory remained within its limit. In the scenarios where a breach in memory occurs, based on the exchange property, visual illustrations were provided to show the effect of the schedule. In summary, AA has demonstrated the potential to be useful in assisting in improvements of satellite scheduling and thus could be integrated into an explainable layer within an autonomous scheduling system. In addition, the enhanced capabilities of a TAF were later explored with the expanded satellite scheduling problem defined in Chapter 3.5.2.

5.3.3 Tripolar Argumentation for the Expanded satellite schedule

Satellite scheduling requires balancing many varied and sometimes conflicting constraints to maximise mission objectives; in this section, the expanded satellite schedule developed in Chapter 3.5.2 was used to successfully create a more complex schedule, aligned with real-world requirements for specific areas of interest. The schedule was used to employ a TAF to explore the potential for the argumentation framework to fit XAI needs.

Through applying a Tripolar Argumentation (TA) layer to this schedule, as outlined in Chapter 5.3.1, the study explores how argumentation for XAI may provide insight into decision-making trade-offs by facilitating conflict resolution for each action, improving the interpretability of scheduling processes in autonomous mission planning.

5.3.3.1 Methodology

Applying the concept of argumentation to this satellite scheduling problem, all relevant factors must be considered to establish the support ‘+’ and attack ‘-’ conditions for each action. With this scheduling problem, power consumption P was integrated with the existing constraints, including memory availability and satellite position.

The argumentation layer examines the feasibility of every action a for all targets to be scheduled across every time interval. It was formulated based on the weighted relationships between competing actions, as only one action can be executed per time step. This establishes the potential replacements for any scheduled action at time instance i , by considering the conditions that exist during the specified time.

For example, if an image-taking action for a specific target that is visible (VT) was scheduled at i , the argumentation layer would evaluate all attack conditions to the target and the actions related to all other observation targets to discern why the decision maker has assigned the action of observing VT . This involves:

- Attacks from $N - 1$ other image-taking actions (one for each alternative observation target, where N represents the total possible image-taking actions).
- Attacks from all N alternative actions (processing, downlinking, and idle).
- Environmental constraints affecting both the visibility of the target (VT) and competing actions.

Figure 5.10 represents an overview of the argumentation structure, illustrating the actions’ interconnected nature and their respective conditions. Each action is associated with a goal (G) that defines the number of images requested for the observation target (VT_n). Additionally, the probability (PR) of any action occurring influences the decision makers’ output, reflecting the likelihood of the specific actions and targets impacting the scheduled action.

Through the process of evaluating the scheduling decisions, each possible action was assigned a weighted value based on its constraints and feasibility. These weighted values represent the influencing factors such as available onboard memory, power consumption,

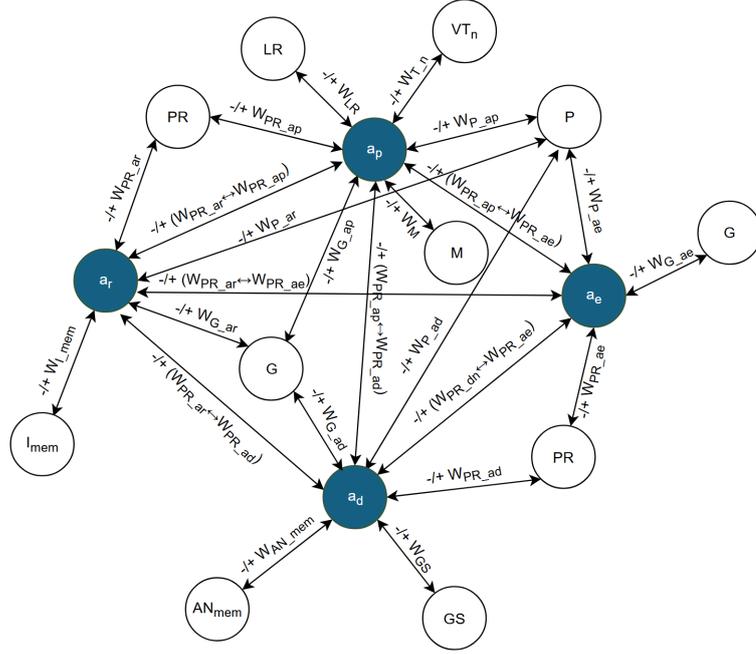


Figure 5.10: An overview of the argumentation structure for supports and attacks for all actions within the schedule as well as the environmental conditions and impact on all actions

Target visibility, and ground station access. The weighted values associated with image taking action a_p were calculated by:

$$W_M = 1 - \left(\frac{M}{Tot_a} \right) \quad (5.6a)$$

$$W_{T_n} \in \{-5, 1\}^i \quad (5.6b)$$

$$W_{LR} \in \{-5, 1\}^i \quad (5.6c)$$

where W_M in Equation 5.6a represent the weight associated with the onboard memory (M), and Tot_a denotes the total number of images currently in memory. In Expressions 5.6b, and 5.6c; W_{VT_n} and W_{LR} respectively are the weights governing whether target (VT_n) is visible or within light range (LR). The -5 negates all possible positive supports when the critical conditions are not met, as the actions are unable to be executed without these conditions. For example, a target may be visible, adequate memory and sufficient power may be available, but light is out of range (not present), and there-

fore the action must not be scheduled to execute. However, when the conditions are met, a 1 is applied to allow for execution.

For image processing a_r , based on the constraints of images in memory, the weight is associated with the unprocessed images in memory I_{mem} . Therefore, the weight is calculated by:

$$W_{I_{mem}} \in \{-1, 1\}^i \quad (5.7)$$

Where 1 in Expression 5.7 stands for the presence of a specified unprocessed image of the target that would support the action to be scheduled, while -1 would imply there are no unprocessed images of the target available for processing, which would hinder the action from being scheduled. For example, if the image of one target is in memory, and the processing of the image related to another target is being considered to be scheduled, this won't be possible. Thus, -1 is applied.

Similarly to a_r , the weight relations for downlinking of images a_d are reliant on processed images of the target in memory as well as ground station access. The expressions are as follows:

$$W_{An_{mem}} \in \{-1, 1\}^i \quad (5.8a)$$

$$W_{GS} \in \{-5, 1\}^i \quad (5.8b)$$

In Expression 5.8a, $W_{An_{mem}}$ relies on whether or not there is a processed image of the target available in memory for download. Likewise, in Expression 5.8b, W_{GS} is solely based on ground station accessibility at i . A -5 value is used to negate all positive supports with no ground station, as it would not be possible to execute.

When no other actions are possible, there is idle, which also has its own weight to be calculated. These are:

$$W_{G_{ae}} = - \left(\frac{1}{Tot_a} \right) \quad (5.9a)$$

$$W_{P_{ae}} = 1 - \left(\frac{P - 1}{100} \right) \quad (5.9b)$$

Equations 5.9a and 5.9b reflects the relationship between idle (a_e) and the mission's goal (G) for $W_{G_{ae}}$; and the relationship between power (P) and a_e for $W_{P_{ae}}$ respectively. Considering the satellite's power level, the likelihood of idling increases with a decrease in power.

In addition to the expressions and equations mentioned, there are two additional weighted calculations used across actions a (a_p , a_r , and a_d). These are:

$$W_{P_a} = P - \left(\frac{P_C * t_{a-dur}}{100} \right) \quad (5.10a)$$

$$W_{G_a} \in \{-1, 1\} \quad (5.10b)$$

W_{P_a} in Equation 5.10a represent the argument weight between the power level P and action execution, where P_C stands for the power consumed with t_{a-dur} action duration. W_{G_a} Expression 5.10b, however, assesses the weight between the goal G and any action based on the specified requirements of the number of images to be retrieved.

An overall weight is calculated for the specific target with the respective scheduled time. The total weight for any action scheduled, including idle time, was calculated by summing all weights. This means the larger weighted value, based on the attacks of all actions for all targets, would determine the direction of the attack, leading to the action being scheduled. The probability (PR) values were created to reduce computational complexity and resources due to the exponential growth in required calculations for the alternative actions.

A representation of the overview of the combined attack variations with actions and their respective environmental conditions is shown in Table 5.2

Table 5.2: An overview of each action and condition attack required for every action type.

Action	Action and Condition Attacks									
a_d	P_a	G_a	PR	AN_{mem}	GS	a_d^{1-N}	a_p^{1-N}	a_r^{1-N}	a_e	
a_r	P_a	G_a	PR	I_{mem}		a_r^{1-N}	a_p^{1-N}	a_d^{1-N}	a_e	
a_p	P_a	G_a	PR	M	VT_n	LR	a_p^{1-N}	a_r^{1-N}	a_d^{1-N}	a_e
a_e	P_i	G_i	PR				a_r^{1-N}	a_p^{1-N}	a_d^{1-N}	

5.3.3.2 Results and Discussion

Every action at every time interval following the methodology was computed and analysed to assess the capabilities of argumentation for the derived schedule. With a schedule of 9000 scheduled actions across the time horizon, a sample of three actions (a_p , a_r , and a_d) taken at different time intervals was evaluated and shown in Table 5.3. The closest alternative action to be selected based on the probability and argument weight (provided as an input to the problem) was also included to provide additional context. The table outlines the action scheduled with its respective time and weight values for each argument condition based on Table 5.2. As there are 10 targets that each action has to be applied to, for simplicity of data display, only the total attack value for reaction type is displayed for clarity, as there were over 3500 calculations per time interval that would otherwise be required.

Action a_{d4} (downlink for target 4) at time 208 was scheduled with a total argument weight of 4.0126, primarily influenced by the high probability score PR returned by the decision maker. In addition, PR also received strong support from the processed images in memory (AN_{mem}) and ground station access (GS) that was similarly provided to a_{d7} , which was the next best option. The variance in this was linked to the PR score and attacks from the other options, leading to a total weight of 2.1247, which was lower than the scheduled action, confirming that the decision-maker made a justifiable selection, prioritising a_{d4} over its alternative. Looking into a_{r1} (processing the image of target 1) that is scheduled at time 480, achieved a total argument weight of 3.9029. This was also justified by the high PR score and the available unprocessed images of that target in memory I_{mem} . The next best option, a_{ar2} (processing of the image taken

Table 5.3: Example calculations of argumentation weights for each action type and the closest probable alternative, except Idle. *Please note, the total of all attacking actions is presented here.

Time (s)	Action	Action and Condition Attacks										Total
208	a_d	P	G	PR	A_{mem}	GS		a_d^*	a_p^*	a_r^*	a_e	4.0126
	a_{d4}	0.7300	1.0000	0.9715	1.0000	1.0000	-	-0.0273	-0.0000	-0.6614	$-6.5974e^{-05}$	2.1247
208	a_{d7}	0.7300	1.0000	0.0276	1.0000	1.0000	-	-0.9715	-0.0000	-0.6614	$-6.5974e^{-05}$	2.1247
480	a_r	P	G	PR	I_{mem}			a_r	a_p^*	a_d^*	a_e	3.9029
	a_{r1}	0.9500	1.0000	0.9764	1.0000	-	-	-0.0235	-0.0000	-0.0000	$-5.2090e^{-08}$	1.9901
480	a_{r2}	0.9500	1.0000	0.0200	1.0000	-	-	-0.9799	-0.0000	-0.0000	$-5.2090e^{-08}$	1.9901
696	a_p	P	G	PR	M	VT	LR	a_p	a_r^*	a_d^*	a_e	5.4893
	a_{p7}	0.7250	1.0000	0.9983	0.7667	1.0000	1.0000	-0.0000	-0.0001	-0.0000	-0.0006	1.4927
696	a_{p10}	0.7250	1.0000	0.0000	0.7667	-1.0000	1.0000	-0.9983	-0.0001	-0.0000	-0.0006	1.4927

of target 2), also had an unprocessed image in memory; however, it received a lower weighted score of 1.9901. The decision maker, therefore, made a reasonable choice as it ensured optimal scheduling based on the available conditions.

During the time interval of 696, a_{p7} (the image taking action for target 7) was scheduled with a score of 5.4893. Again, this was due to the PR score in conjunction with the visibility of target 7 (T_7) and light exposure (LR). In contrast to the next alternate option of a_{p10} with an argument weight of 1.4927 due to the absence of the observation target at that time and a probability of 0.0000, this is an implication that the decision-maker selected the correct action.

A visual was created for this problem, using the Figure 5.10 and applying the scores from Table 5.3 for a_{d4} at time 208, Figure 5.11 was created presenting the complete support (single arrows), attack (double-headed arrow), and neutral (bidirectional arrows) conditions. The figure represents the outcome of the action attacks for the specific targets:

- a_{r10} (processing of the image of target 10), attacking a_{d4} , a_e , and a_{p1} producing a weighted score of 0.4025.
- a_e (idle action) attacking a_{d4} , a_{ar10} , and a_{p1} with a score of -1.4734.
- and a_{p1} (image processing for target 1) attacking a_{d4} , a_e , and a_{r4} with a score of -4.0636.

These actions for the targets were selected to see the conflicts within the system under the scenario of the user wanting to schedule another action for a different target

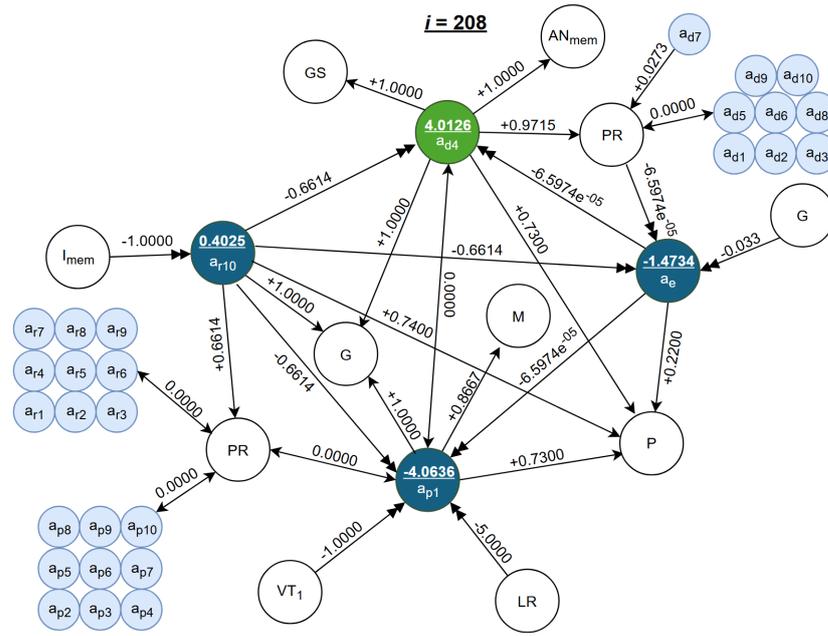


Figure 5.11: A Tripolar argumentation representation of arguments between 4 main actions (a_{r10} , a_{d4} , a_e , a_{p1}) and environmental conditions at time $i = 208s$.

instead of the same target. It was observed that a_{p1} was not possible due to the absence of light and target (T_1) visibility, which had the lowest score despite memory availability, power and goal alignment. For a_{r10} , it was not scheduled due to no available images in memory for target 10, which negatively influenced its feasibility despite the probability, power, and goal. Likewise, a_e was less favourable than a_{r10} but more favourable than a_{p1} ; overall, this concept applied to all other competing actions at this time step and all other times. It can also be observed that all other actions (image taking - a_{p2} , a_{p3} , a_{p4} , a_{p5} , a_{p6} , a_{p7} , a_{p8} , a_{p9} , and a_{p10} ; image processing - a_{r1} , a_{r2} , a_{r3} , a_{r4} , a_{r5} , a_{r6} , a_{r7} , a_{r8} , and a_{r9} ; and downlinking - a_{d1} , a_{d2} , a_{d3} , a_{d5} , a_{d6} , a_{d9} , a_{d10}) all carried a weight of 0; where a_{d7} had a weight of $+0.0273$. This means all environmental factors were attacking each action as well as all actions attacking each other except themselves at this time; however, the 3rd-dimensional visualisation is not represented here, but is calculated.

The principles of TA were applied to a satellite schedule derived via a RL approach. The results demonstrated how argumentation can calculate, visualise, explain and validate the decisions made by autonomous systems through the assessments of

all contributing conditions. The satellite scheduling problem contained numerous variables that needed to be evaluated to justify the decision-making system. This approach captured the relevant information and successfully executed the calculations while also considering the values for each condition, clearly illustrating why the scheduled actions were chosen over alternate options.

This provides transparency to users on influencing factors for the decisions and enables informed interventions to modify system behaviour in validating the system's effectiveness in generating the desired outcome. Careful consideration must be given when designing and implementing AFs due to the potentially large datasets that would contain a vast number of variables with their respective conditions and constraints that may require performing complex calculations. If the framework developed is computationally heavy, it will utilise the time and computing resources needed to perform these calculations.

5.4 Conclusion

This chapter initially introduced a SCA framework for evaluating satellite schedules based on three key criteria: total images captured, idle time, and downlink efficiency. By applying mathematical formulations, the analysis provides a transparent methodology for selecting and investigating optimal schedules based on mission-specific objectives, which aids operators in making informed decisions on which schedules are more effective than others. Future work in this area can explore additional metrics, such as energy consumption and dynamic reconfiguration under uncertainty. Additionally, the application of Multiple Comparative Analysis (MCA) may be used to compare different schedules at once or multiple comparisons across several instances at once.

Following the SCA study, different approaches of argumentation explored in these studies demonstrated their potential uses for explainability of automated systems, and can provide significant detail and insight to the factors that support or challenge the allocation of tasks and resources in a schedule. The AA approach showed promise in evaluating and improving satellite scheduling when combined with SEP and PEP techniques, where fine-tuning schedule improvements could be identified. The informa-

tion gathered on the impact of switching the positions of one or more actions across the schedule would greatly support system operators in confirming or modifying automated scheduling decisions and provide transparency on system functionality and logic, enabling informed decisions and aiding in the building of trust.

The TA methodology used with the expanded satellite schedule was able to add a more granular and quantifiable response to evaluating the scheduling decisions. Through the ability to calculate and present weighting scores for each of the possible actions a system operator would be able to clearly identify the strongest and most effective choice for each time interval, as well as being able to review how the weighting was calculated resulting leading to informed knowledge and decision-making capabilities. Both approaches allow for enhanced interaction and transparency to an automated systems decision-making which can assist with validation of systems to allow for adoption in stringent industries, such as space operations, with the opportunity to develop evidence-based trust in system decisions. Consideration, however, must be given to the resourcing requirements necessary to complete argumentation calculations on large datasets as this may be time or cost prohibitive for certain missions or projects. Future research should explore the most cost-effective and resource efficient methods for performing calculations, as well as applying the principles outlined in these studies to larger or more complex systems to observe performance and accuracy when used at scale. Argumentation can also be combined with other techniques, such as NLP, to form part of a holistic solution to variable datasets and automated systems.

Chapter 6

Large Language Model application to Argumentation

6.1 Chapter Overview

This chapter looks into the application of utilising LLMs with SEP and AA techniques for an EO satellite schedule, followed by assessments of the results as explained in Chapter 2.5. The type of satellite scheduling applied here is that described in Chapter 3 with the extension of the SEP argumentation concept from Chapter 5.3.2 where NLP techniques defined in Chapters 8.2, and 2.5 are applied. This chapter was created from “*Natural language processing for explainable satellite scheduling*”, Paper presented at SPACEOPS 2023, by *C. Powell, A. Riccardi, and A. Berquand* [40], “*Abstract argumentation for explainable satellite scheduling*”, Paper presented at DSAA 2022 [39] [Peer-reviewed] as well as “*On-board re-planning of an earth observation satellite for maximisation of observation campaign goals*”, Paper presented at 73rd International Astronautical Congress (IAC) 2022, both papers authored by *C. Powell, A. Riccardi* [38].

The chapter is broken into four sections:

- The Literature Review (6.2) contains a background on LLMs use with different XAI techniques, as well as their application with argumentation techniques explained in Chapter 5.3.

- The Methodology (6.3) outlining the application of an LLM with the concept of SEP, defined in Chapter. 5.3.2, and the metrics used to evaluate performance.
- The Results (6.4) are examined to assess and compare the performance of the different approaches outlined in the Methodology (6.3).
- The Conclusion (6.5) discusses the findings from the results and observed capability, as well as the future opportunities that exist for further research.

6.2 Literature Review

As identified in Section 2.4 LLMs are evolving in sync with intentions to incorporate XAI techniques, aimed at improving the interpretability and transparency of these models. By embedding XAI methods, researchers strive to gain insights into the decision-making processes of autonomous systems and enable users to understand and trust their outputs [357, 358].

There are a multitude of different approaches in development for XAI, as shown in recent studies [359, 360], 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 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 quantitative-based explanations or are built to be model-specific and do not offer easily understandable, human-language responses. Additionally, the explanations generated are single executions 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 optimise 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 [361]. 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 [362]. 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 [363].

Techniques like LRP as introduced in Chapter 2.4.4, Table reftab: Model-S Techs 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 [194]. 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 [364].

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 [365]. 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 [366]. 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 [367]. 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. 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 RNN models [368].

For XAI in recommender systems, research was conducted with the aim to provide users with insights into product recommendations [369]. Their work emphasised 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/visualisation), using a conjoint survey. Results showed a preference for local explanations and visualisations 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.

LLMs adoption for space applications has demonstrated their capabilities in several research; one such experiment considered the capabilities of a Vision-Language Model (VLM) through the comparison of different pertaining methodologies of a fine-tuned Large Language and Vision Assistant (LLaVA) model against that of a zero-shot ap-

proach through the generalist LLM GPT-4 and the original LLaVA model. It was observed the LLaVA model outperformed GPT-4 and the base LLaVA model consistently in visual problem-solving tasks, such as navigation pathfinding, highlighting the advantage of tuning generalist models for domain-specific applications [370].

Another study implemented an LLM-based solution to the Kerbal Space Program Differential Games (KSPDG), which is a competition for manoeuvring satellites autonomously within a game engine. The KSPDG comprises three challenge scenarios: Pursuer-evader, Target guarding, and Sun-blocking, requiring commands to be given to a craft to achieve the objectives to minimise the distance to an evader craft or to position the craft between the Sun and the evader. Using prompt engineering, few-shot prompting, and fine-tuning techniques to maximise the performance of the LLM using a GPT-3.5 approach at completing the challenge, achieving second place overall. Through the experience of utilising the combined optimisation techniques, the study proposes that generalist Large Language Models (LLMs) can outperform RL approaches as they can thrive on limited data and simulations, which could be well suited to applications within the space industry [371].

The potential impact space debris can have on future operations has been a growing concern and has resulted in the creation of inventories of objects in space, where an investigation has undergone the use of LLMs in answering queries of these space object catalogues. A recent study has evaluated six LLMs, including both GPT-4 and ChatGPT-3.5, and investigated their ability to answer expert-derived queries on space debris and objects utilising a bespoke dataset combining KQA Pro Dataset and ESA's DISCOS Database [372].

Additionally, data retrieval from the ESA Astronomy Science Archives was explored by the ESA using ESA Virtual Assistant (EVA), an in-house developed LLM and through the adoption of the Retrieval-Augmented Generation (RAG) architecture; their search performance improved to near 100% accuracy as compared to zero and few-shot techniques. As EVA is not open-source, its applicability must be validated using generalist LLMs to determine the broader feasibility of RAG-based architectures in space applications [373].

LLMs have also been proposed for use in space medicine by combining healthcare and space exploration interests. An article discussed the use of ChatGPT’s potential role in emergencies to provide medical advice in distant space environments where expert consultation is unavailable. Through this experiment, the limitations were acknowledged for being a generalist model in domain-specific expertise and, therefore, would need to be fine-tuned to become a viable solution [374]. This notion is not limited to healthcare but extends beyond to include other areas such as scheduling updates, automating tasks, and operational decision support.

Despite the extensive research into the potential applications of AI systems in satellite operations and task scheduling [307, 375–379], explaining and validating AI decisions within spacecraft operations remains a significant gap. Therefore, XAI is crucial in addressing this issue, making it more transparent, interpretable, and trustworthy.

Bridging the gap between AI automation and human interpretability for satellite scheduling, Computation Argumentation, as stated in Chapter 5, focuses on logical reasoning through analysing and structured decision-making processes and identifying relationships between influencing variables, such as time, available resources, and environmental conditions [380]. The frameworks developed to formalise argumentation such as AAF, BAF, TAF [344, 348, 380], iAF [347], SAF [381] provide enhanced capabilities for handling real-time scheduling constraints. However, for NLP based methods, enabling semantic understanding of scheduling decisions and enhancing explainability [382], complementing argumentation techniques is an underexplored area, leading to the research within this chapter.

6.3 Methodology

Utilising the principles and constraints outlined in Chapters 3 and 5.3.2, a day was randomly selected, excluding the first day when all variables were initialised. An argumentation layer SEP was used to assess the capabilities of initially integrating LLMs with this concept.

An overview of the process is shown in Figure 6.1, illustrating the role of NLP in establishing communication between the end user and the system. This enables the

user to submit a query related to the schedule, where the argumentation layer extracts the relevant information from the scheduler. To provide a response to the user, the data is initially structured in a list of arguments and passed to the LLM combined with the query to generate an explanation.

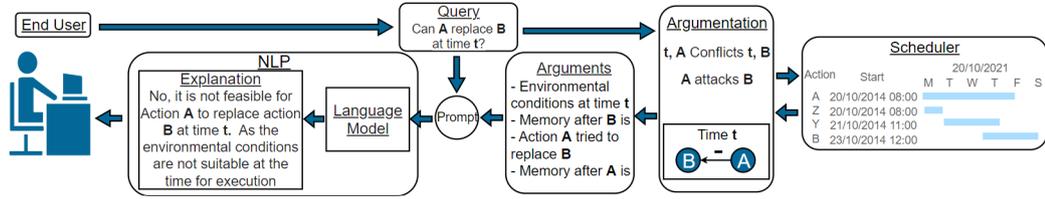


Figure 6.1: The use of NLP in the system.

6.3.1 SEP application with the schedule

To provide a visual of the SEP concept with satellite scheduling for each time instance, Figure 6.2 represents an overview of the memory profile across three different time intervals (m_{in} , m_{in+1} , m_{in+2}) and how the change works within an existing schedule. In this figure, green denotes the scheduled action, while grey represents the actions trying to take its place (attack). Additionally, the attacking actions are also under the influence of constraints and external factors, shaded in blue for attacks and yellow for supports.

At the time m_{in} , if a query was asked if an action could replace the scheduled action a_r , the potential attacks would be a_p , and a_d based on whether there is:

1. enough memory onboard, enough visibility of land, and exposure to sunlight for a_p .
2. Access to the ground station and enough processed images in memory for a_d .

As mentioned in Chapter 5.3.2, the replacement is only feasible on the condition the memory profile hasn't been exceeded at any point, and in this case, along branch a , memories (m_{ian+1} , and m_{ian+2}) respectively. This is similarly applied to m_{ian+1} , where action a_e is challenged by a_d , a_r , and a_p once all the constraints and future memory states are met for branch b for m_{ibn+2} , and m_{ibn+3} . Likewise for m_{in+2} for branch c .

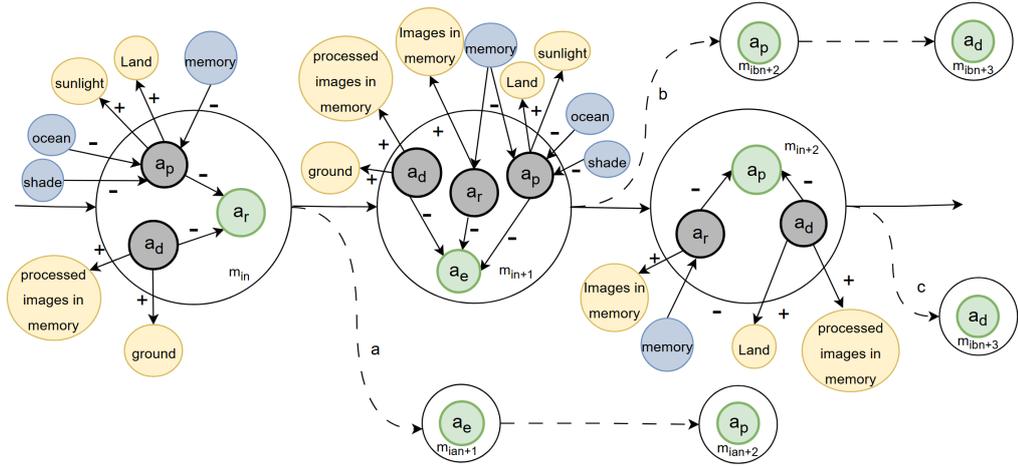


Figure 6.2: The functionality of SEP applied in 3 scenarios resulting in 3 different memory branches with the initially scheduled actions (a , b and c) on the conditions that the attacks are successful.

In the scenario where a_p was to be replaced, the satellite coordinates and field of view would be retrieved, using the methodology from Chapter 3.3.2, to determine the next opportunity to recapture the image, enabling the system to inform the user when this will be.

6.3.2 LM selection and hyper-parameters

At the time of this experiment, several LMs were investigated, such as Pathways Language Model (PaLM), BLOOM, and GPT-3. PaLM was found to be cost prohibitive and BLOOM was primarily focused on multi-linguistic capability, as such GPT-3 provided the most suitable results as a generalised LLM at the time of this experiment, based on the literature and was therefore chosen with the following parameters:

Table 6.1: Hyperparameters used for GPT-3

hyper-parameters	Values
model	<i>text-davinci-003</i>
temperature	0.7
max_tokens	256
top_p	1
frequency_penalty	0
presence_penalty	0

The parameters selected represent the configuration settings to alter the output behaviour of the LLM. These parameters were chosen to achieve the best results through iterative testing for interfacing with the SEP layer created.

6.3.3 Prompting approach

Implementing the prompting approach (the input text to the model), every prompt consists of a query such as: *Can action A replace action B at time T?* Here, the list of arguments based on the constraints and the query is prompted to the LLM. Shown in Figure 6.3 is a representation of the decision flow, where the query is processed by the argumentation layer, where the conditions are verified, and where the conditions are met, a revised memory profile is checked for the rest of the scheduled actions.

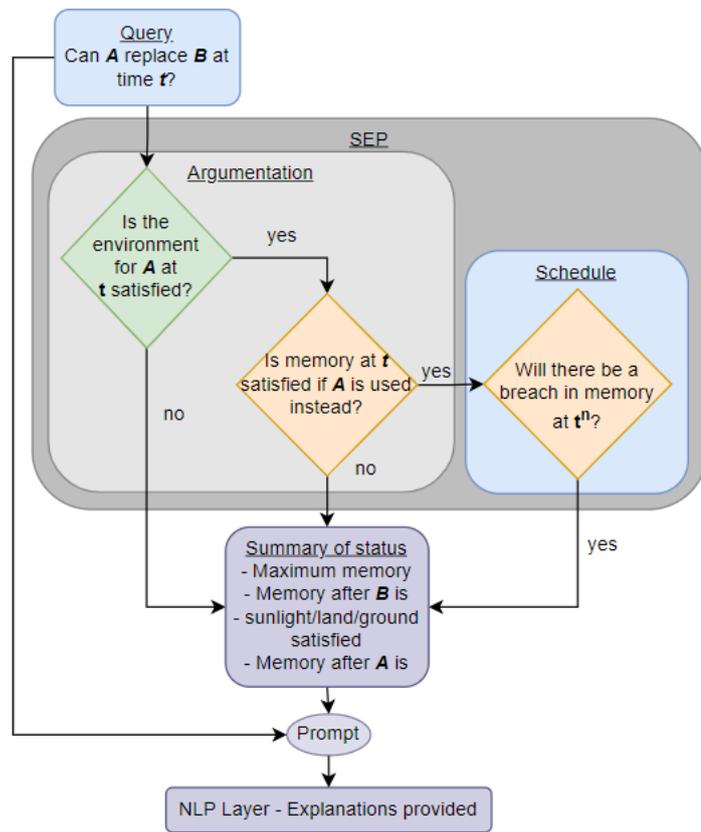


Figure 6.3: Decision flow based on the constraints and environmental conditions for generating prompts from the AA layer to establish an explanation based on the query from the user

Within Table 6.2, a few examples include user queries with the corresponding list of example arguments provided, which is the baseline prompt used in this Chapter, forming a Zero-Shot Learning (ZSL) approach (introduced in Chapter 2.5, where no examples or guidance is provided). The LLM analyses the arguments with the query, determines if an exchange is possible, and provides a justifiable explanation. Following this, Few-Shot Learning (FSL) (also introduced in Chapter 2.5, where a small number of examples are provided), was investigated with three and ten examples created by the authors in the format of *query*, *arguments*, *examples* to provide a reference for the model to improve explanation accuracy.

Table 6.2: Argumentation Prompt Examples

Query	Arguments
Can the Image taking action replace the Processing action at time 74701?	<ul style="list-style-type: none"> - The Processing action was initially scheduled - Processing image number 2928.29 - Memory for the scheduled action is 1804098 MB - Image taking action tried to replace Processing action - Land is satisfied - Sunlight is satisfied - The memory after Image taking at time 74701 would be 1806536 MB - The memory would have exceeded at time 79106 with a value of 1920094 MB - Maximum memory is 1920000 MB
Can the Down-linking action replace the Processing action at time 81752?	<ul style="list-style-type: none"> - The Processing action was initially scheduled - Processing image number 3012.56 - Memory for the scheduled action is 1391638 MB - Down-linking action tried to replace Processing action - No access to the ground station - Maximum memory is 1920000 MB
Can the Image taking action replace the Processing action at time 27646?	<ul style="list-style-type: none"> - The Processing action was initially scheduled - Processing image number 2216.8 - Memory for the scheduled action is 1232350 MB - Image taking action tried to replace Processing action - Land is satisfied - Sunlight is satisfied - The memory after Image taking at time 27646 would be 1234788 MB - The memory would have exceeded at time 79106 with a value of 1920094 MB - Maximum memory is 1920000 MB

6.3.4 Dataset and Evaluation Metrics

Fifty queries were created, each with its own set of arguments and answers created by the authors as a ground truth response to evaluate the LLMs output. For FSL, ten additional queries with their arguments and answers were generated.

To evaluate the response from the LLM, five metrics were applied:

- BLEU: A precision-oriented score that measures n-gram overlap between generated and reference text [383].
- ROUGE: A recall-oriented metric assessing n-gram and sequence overlaps [384].
- BERTScore: A token-level similarity measure using contextual embeddings and summing their cosine similarities [260].
- Sentence-BERT (SBERT) Score: A sentence-level similarity metric using cosine similarity between embeddings [385].
- Human Evaluation: Three annotators assessed generated explanations for correctness and completeness using a rating scale adapted from best practices in [386].

Where BLEU and ROUGE both focus on syntactic matches, where two transformer-based measures were applied based on token-level and sentence-level embedding [384]. Human evaluation, however, analyses the data more in depth for the quality of the response. In [387], four annotators assessed the factual accuracy of 30 generated facts, where the mean of these scores was correlated to the ROUGE and internal evaluation metric. These metrics have a similar score where 1 represents the highest and best results, while 0 represents the lowest and worst results.

The data evaluated by humans were under the following criteria shown in Table 6.3 according to best practices mentioned in [386], which were modified for this chapter. The quality of the text was based on the accuracy and the quality of the model deduction.

Table 6.3: Scoring scale (end result: normalise average score)

Points	Justification
3	The model's deduction is correct, and its explanation leverages all relevant environmental and memory parameters.
2	The model's deduction is correct, but its justification is partial and lacks reference to all the relevant parameters.
1	The model's deduction is correct but its justification is wrong, it either lacks to reference the relevant parameters or makes a faulty usage of them.
0	The model's deduction is incorrect.

6.4 Results

Scores were obtained for the three different prompting approaches: ZSL, few-shot with 3 (FSL-3) and 10 examples (FSL-10) and shown in Table 6.4. For the different prompting approaches, a trend was observed that as the number of provided examples increases for FSL, so does the performance across all evaluation metrics.

- Human evaluation assessment demonstrated an increase in explanation quality between the ZSL and FSL approach, with the highest mean score of 0.84 for FSL-10 as opposed to ZSL with a score of 0.53.
- BERTScore, and SBERT (Semantic Metrics) showed that the semantic richness was better captured as more examples were provided, even though the increase is marginal, suggesting the semantic similarity stabilises within a certain range of examples.
- BLEU, and ROUGE (Syntactic Metrics) produced the lowest scores across all settings (except SBERT for FSL-10 compared to BLEU). These performance gaps indicate that the metric was less reliable for evaluating explanation quality. This is due to the text generated by the LLM often using synonyms or different words to the ones found in the target text.

Table 6.5 displays the explanations automatically generated with FSL-10, and their respective scores, for the prompts presented in Table 6.2. As shown in this table, the syntactic metrics reflect poorly on the quality as their scores depend on the tokens contained in the target text, and not on the meaning of the target explanation. Overall, the BERTScore and SBERT align more closely with the human evaluation metric, showing a more trustworthy assessment of informativeness. The human evaluation remained the most reliable, though it may be time-consuming as opposed to automated metrics.

While GPT-3 demonstrated factual accuracy of 90% of the queries with ZSL by correctly stating if the exchange of actions was feasible, it showed an accuracy rise to 96% with the FSL method. The model exhibited an impressive understanding of the

Table 6.4: Average scores and standard deviation (subscript) obtained for each evaluation metric and prompting approach.

Approach	BLEU	ROUGE	BERTScore	SBERT	Human
ZSL	0.56 _{0.14}	0.37 _{0.11}	0.83 _{0.03}	0.73 _{0.1}	0.53 _{0.08}
FSL-3	0.68 _{0.15}	0.41 _{0.12}	0.85 _{0.03}	0.72 _{0.12}	0.83 _{0.12}
FSL-10	0.74 _{0.12}	0.43 _{0.09}	0.86 _{0.02}	0.73 _{0.1}	0.84 _{0.09}

Table 6.5: Examples of explanations generated by the GPT-3 model with the few-shot learning 10 examples approach

Query	Target Text	Generated Text	Scores
Can the Image taking action replace the Processing action at time 74701?	The processing action was attempted to be replaced by image taking, resulting in a violation of the memory constraint with a saturation at time 79106, thus meaning processing was critical for this time. The two constraints at the time of the attack were initially satisfied, but the exchange was infeasible.	No, the image taking action cannot replace processing at time 74701 because the memory after the action would exceed the maximum memory at a later time. Therefore, the processing action is not feasible to be replaced.	BLEU: 0.65 ROUGE: 0.34 BERTScore: 0.83 SBERT: 0.5 Human: 0.78
Can the Down-linking action replace the Processing action at time 81752?	No, it is not feasible to replace the Processing action at time 81752 with a Down-linking action. The Processing action was initially scheduled for Processing image number 3012.56 and with a scheduled memory of 1391638 MB. With no access to the ground station and no change to the scheduled memory, it would not be possible to replace the Processing action with a Down-linking action.	No, the down-linking action cannot replace the processing action at time 81752 as there is no access to the ground station. Therefore the processing action will continue to process image number 3012.56.	BLEU: 0.40 ROUGE: 0.51 BERTScore: 0.88 SBERT: 0.90 Human: 1
Can the Image taking action replace the Processing action at time 27646?	It is not feasible for image taking to replace processing at time 27646. It is initially possible with both sunlight and land satisfied, however, due to the memory being exceeded later at time 79106, it is no longer possible for this action to take place	At time 27646, the memory after the image aking action would be 1234788 which is below the maximum memory. However, later at time 79106, the memory would be exceeded resulting in an infeasible result which means, image taking cannot replace the processing action at time 27646.	BLEU: 0.81 ROUGE: 0.42 BERTScore: 0.84 SBERT: 0.74 Human: 0.89

domain-specific terminology despite the variations in the quality of the explanations, as shown in Tables 6.6 and 6.7.

The main observations from the three human evaluators are:

- **Temporal Reasoning:** Due to the FSL approach demonstrating a greater understanding of the concept of time, including whether memory constraints will be violated and image retaking opportunities.
- **Linguistic Complexity:** ZSL and FSL-3, they revealed an inclination to reuse the

Chapter 6. Large Language Model application to Argumentation

Table 6.6: Examples of text generated by the GPT-3 model with different learning approaches .

Query	Arguments	Approach	Outputs	Score
Can the Downlinking action replace Image taking action at time 56661?	<ul style="list-style-type: none"> The image-taking action was initially scheduled. Land satisfied for the scheduled action Sunlight satisfied for the scheduled action Memory for the scheduled action is 729698 MB Down-linking action tried to replace Image taking action Ground station access The memory after Down-linking at time 56661 would be 724210 MB The image number that would be down-linked is 2540.63 The satellite coordinates are: latitude 67.027671, longitude -60.361479, and image retake times are:('day:6', 'time: 57251'), ('day:6', 'time: 57256'). Maximum memory is 1920000 MB 	Target:	The down-linking action can replace image taking at time 56661. The ground station is accessible, thus enabling the downlinking of image 254063 to take place. If down-linking was to take place instead, the image retake times would be on day 6 at 57251 and 57256 to retrieve the image that was replaced.	
		ZSL:	Yes, the Down-linking action can replace the Image taking action at time 56661. The memory after downlinking at time 56661 would be 724210 MB and the image number that would be downlinked is 2540.63. The satellite coordinates are latitude 67.027671, longitude -60.361479 and the image retake times are day:6 time: 57251 and day:6 time: 57256. The maximum memory available is 1920000 MB.	BLEU: 0.58 ROUGE: 0.33 BERTScore: 0.82 SBERT: 0.74 Human: 1
		FSL 3:	Yes, the down-linking action can replace the image-taking action at time 56661 as the memory would not be saturated, and ground station access is available. The image number that would be downlinked is 2540.63 and the next opportunity for image retake at latitude 67.027671, longitude -60.361479 is on day 6 at time 57251.	BLEU: 0.72 ROUGE: 0.39 BERTScore: 0.87 SBERT: 0.64 Human: 0.89
		FSL 10:	It is feasible for down-linking to replace image taking, as the memory would remain below the maximum memory. The next opportunity for this image to be retaken is on day 6 at time 57251.	BLEU: 0.78 ROUGE: 0.41 BERTScore: 0.92 SBERT: 0.72 Human: 0.78
Can the Image taking action replace the Processing action at time 33816?	<ul style="list-style-type: none"> The Processing action was initially scheduled Processing image number 2297.53 Memory for the scheduled action is 1198358 MB Image taking action tried to replace Processing action Land is satisfied Sunlight is satisfied The memory after Image taking at time 33816 would be 1200796 MB The memory would have exceeded at time 79106 with a value of 1920094 MB Maximum memory is 1920000 MB 	Target:	The exchange was initially possible at time 33816 but the memory would later be exceeded at time 79106 therefore making it infeasible to do the exchange.	
		ZSL:	No, the Image taking action cannot replace the Processing action at time 33816. The memory after Image taking at time 33816 would exceed 1920000 MB after 79106, which exceeds the maximum memory allowed.	BLEU: 0.53 ROUGE: 0.34 BERTScore: 0.84 SBERT: 0.71 Human: 0.33
		FSL 3:	No, the image taking action cannot replace processing at time 33816 because the memory would exceed the maximum allowed at time 79106. Therefore, it is not feasible for the image taking action to replace the processing action.	BLEU: 0.69 ROUGE: 0.40 BERTScore: 0.80 SBERT: 0.67 Human: 0.67
		FSL 10:	At time 33816, the memory after the image taking action would be 1200796 which is below the maximum memory. However, later at time 79106, the memory would be exceeded resulting in an infeasible result which means, image taking cannot replace the processing action at time 33816.	BLEU: 0.75 ROUGE: 0.44 BERTScore: 0.88 SBERT: 0.73 Human: 0.67

arguments in their explanations as opposed to FSL-10, where the approach effectively captured and reformulated the arguments with more complex vocabulary and varied phrases.

Table 6.7: Continuation of examples of text generated by the GPT-3 model with different learning approaches.

Query	Arguments	Approach	Outputs	Score
Can the Image taking action replace the Processing action at time 7416?	<ul style="list-style-type: none"> The Processing action was initially scheduled Processing image number 2000.28 Memory for the scheduled action is 1674702 MB Image taking action tried to replace Processing action Land is satisfied Sunlight is not satisfied Maximum memory is 1920000 MB 	Target:	Image taking cannot replace processing at time 7416. This is due to sunlight not being satisfied, as a result, leaves the memory is the same and no changes made to the scheduled action.	
		ZSL:	No, the Image taking action cannot replace the Processing action at time 7416. The Processing action was initially scheduled and the memory for the scheduled action is 1674702 MB, which is greater than the maximum memory of 1920000 MB. Additionally, although the land is satisfied, the sunlight is not, which may prevent the Image taking action from replacing the Processing action.	BLEU: 0.59 ROUGE: 0.39 BERTScore: 0.84 SBERT: 0.75 Human: 0.33
		FSL 3:	No, the image taking action cannot replace processing at time 7416 because sunlight is not satisfied which is a requirement for the action to be executed. Therefore memory is the same.	BLEU: 0.65 ROUGE: 0.37 BERTScore: 0.83 SBERT: 0.66 Human: 0.78
		FSL 10:	No, the image taking action cannot replace the processing action at time 7416, because even though land is satisfied, sunlight is not which is a requirement for the action to be executed. Therefore memory will be unaffected.	BLEU: 0.66 ROUGE: 0.38 BERTScore: 0.85 SBERT: 0.68 Human: 1
Can the Down-linking action replace the Processing action at time 7271?	<ul style="list-style-type: none"> The Processing action was initially scheduled Processing image number 1997.58 Memory for the scheduled action is 1667452 MB Down-linking action tried to replace Processing action No access to the ground station Maximum memory is 1920000 MB 	Target:	It is not feasible to replace the processing action at time 7271 with the down-linking action, as there is no access to the ground station. Therefore, the memory remains the same, and the initial action will proceed.	
		ZSL:	No, the Down-linking action cannot replace the Processing action at time 7271. The Processing action requires 1667452 MB of memory, which is greater than the maximum memory available (1920000 MB). Additionally, the downlinking action cannot be performed without access to the ground station.	BLEU: 0.52 ROUGE: 0.36 BERTScore: 0.85 SBERT: 0.76 Human: 0.33
		FSL 3:	No, the Down-linking action cannot replace the Processing action at time 7271 because there is no access to the ground station and the memory is already saturated. This means that the image will not be able to be downlinked and the memory will remain the same.	BLEU: 0.71 ROUGE: 0.43 BERTScore: 0.86 SBERT: 0.74 Human: 0.33
		FSL 10:	No, the down-linking action cannot replace the processing action at time 7271 as there is no access to the ground station. Therefore, image number 1997.58 will proceed to be processed.	BLEU: 0.75 ROUGE: 0.46 BERTScore: 0.82 SBERT: 0.70 Human: 1

- **Conceptual understanding:** The FSL-10 approach effectively captured subtleties of constraints such as resource availability and revisit times as compared to ZSL and FSL-3.
- **Ambiguities:** The ZSL approach often generated confused statements regarding

the processing action (a_r), with the allocated memory.

Three tables, 6.8, 6.9, and 6.10, further represent the average scores obtained for each learning method per type of query. The queries involved replacing Processing (a_r) with Downlinking (a_d) generated the lowest scores for ZSL and FSL-3 as shown in the fourth example of Table 6.7 due to the memory triggered in the model by the word processing. It was observed that the overall average score for ZSL was 1.64, 2.68 for FSL-3, and 2.52 for FSL-10. This demonstrates that providing even a small number of examples does improve the overall performance of a model, but increasing the number of examples may not continue to improve performance. Furthermore, ZSL narrowly out performed FSL-10 in 3 of the 9 query categories (replacing imaging with down-linking or processing, and replacing idle with processing), which highlights the inherent capabilities of a model to interpret data and queries without guiding examples, albeit unreliably.

Table 6.8: Average score per user query with the Zero-Shot approach

Replace \ With	Imaging	Down-linking	Processing
Imaging	-	2.2	2.3
Down-linking	1	-	1.9
Processing	1.6	0.9	-
Idle	1	1	2.9

Table 6.9: Average score per user query with the Few-Shot (3 examples) approach

Replace \ With	Imaging	Down-linking	Processing
Imaging	-	2.8	2.7
Down-linking	2.9	-	2.9
Processing	2.5	1.9	-
Idle	2.7	2.7	3

Table 6.10: Average score per user query with the Few-Shot (10 examples) approach

Replace \ With	Imaging	Down-linking	Processing
Imaging	-	2.1	2.1
Down-linking	2.6	-	2.1
Processing	2.7	2.6	-
Idle	2.7	3	2.8

6.5 Conclusion

Using AA and NLP for XAI for implementation on an EO Satellite Scheduling system has demonstrated significant potential for explaining scheduling decisions. Initially, when a query is retrieved by the system from the end user, a list of arguments is gener-

ated by the AA from SEP. A list summarising the schedule constraints and the status of the schedule for the query, along with the query itself, was prompted to the LLM (GPT-3 at the time of this study) within the NLP layer with no examples (ZSL). The conditions of the AA layer were also prompted with the examples using a FSL approach with three examples (FSL-3) and ten examples (FSL-10). The model responded by providing both an answer to the query and an explanation, demonstrating its ability to process space systems-related queries despite being developed on the general corpus.

The results for FSL-3 and FSL-10 produced contextually rich and more accurate explanations than ZSL. Overall, the approaches achieved high accuracy in answering the questions about the feasibility of an exchange of actions. Therefore, confirming the capabilities of NLP driven methods in enhancing decision transparency, especially when combined with information from an AA for decision context. This research has opened future opportunities in exploring more AFs to expand the diversity and complexity of queries for different datasets, enhancing explanation depths and creating a more detailed analysis of explanation quality for XAI for satellite schedules.

Chapter 7

Large Language Model application to Knowledge Graphs

7.1 Chapter Overview

This chapter is focussed on the use of KGs with LLMs for satellite scheduling derived in Chapter 3, it evaluates LLMs compatibility with KGs for generating executable code through understanding queries of different categories and data extraction from tabular results returned from the KG to then generating an appropriate response to the query. This chapters contents were adapted from “Question answering over knowledge graphs for explainable satellite scheduling” from the proceedings proceedings of the International Astronautical Congress (IAC) 2023 [41], and the extended version published in the Journal of Aerospace Information Systems of the American Institute of Aeronautics and Astronautics (AIAA) in 2025 [43] [Peer-reviewed].

The chapter consists of four sections:

- The Literature Review (7.2) contains a brief review of the research on KGs used with LLMs and the broad range of applications.
- The Methodology (7.3) outlines the process for the experiment, including the configuration of the LLM and KG along with the design of the prompts used to generate results. The metrics used to evaluate performance are also explained.

- The Results (7.4) are examined to assess for capabilities and patterns in performance, in addition to considering the difference when including examples within the query prompt.
- The Conclusion (7.5) discusses the findings from the results and observed capabilities, as well as the future opportunities that exist for further research.

7.2 Literature Review

As described in Chapters 2 and 3, satellite scheduling is a crucial aspect of designing and implementing a successful mission for any satellite. As the operation and scheduling for satellites becomes more complicated, the potential benefit of implementing AI solutions to supplement or replace manual processes becomes greater as well. Therefore, as identified in Chapter 2.3.1, research has begun to explore the capabilities of AI for satellite missions, investigating where more data or faster solutions can be achieved; however, the use of LLMs to assist with interface and explanation has seen little attention. Other disciplines have investigated the potential capabilities and enhancements to LLM reasoning by integrating KGs into the approach, namely open-domain question answering [388], link prediction [389], and recommendation generation [390].

Liu et al. investigated a novel framework called Dual-Reasoning [388], combining the explicit reasoning of KGs and the implicit reasoning of LLMs. This was achieved through an integration process linking a dedicated Graph Neural Network (GNN) with an LLM to enhance question answering capabilities for any query domain. The study tested the performance with 3 different LLMs (Llama2, ChatGPT, and GPT-4) and found that performance improved unilaterally across the 5 tested datasets, enabling much more accurate and consistent responses when compared to stand-alone LLMs.

A study by Shu et al. explored how KGs influence link prediction by directly combining KGs and LLMs into a single model, introduced as KG-LLM [389]. The approach was tested with 3 different LLMs (Flan-T5, Llama2, and Gemma) and demonstrated that a KG-LLM would outperform other LLMs in F1 and AUC metrics, emphasising the potential benefits of combining KGs with an LLM.

Another study, from Rasheed et al. [390], took a simply yet effective approach of supplementing GPT-4 prompts with contextual and structural information from a KG. Even without technical integration between the the KG system and the LLM, the contextualised prompts produced better scores for F1, Precision, and Recall than the LLM without the support from the KG. This highlights that sophisticated or deep integration between systems may not be necessary to improve response performance and minimise technical overhead.

Various techniques have been employed in research to combine KGs with LLMs, including Knowledge-Enhanced Prompting (as outlined in the studies by Rasheed et al. [390] and Li et al. [391]); dynamic adaptation multi-hop link predictions, which was applied by Wang et al. by initially using Markovian searching to identify temporal logical rules from KG data that was then supplied to the LMM to generate a ruleset. The dynamic adaptation was achieved by prompting the LLM to challenge the ruleset and update any low-quality rules followed by the model producing two candidate responses - one based on the ruleset and the other based on the KG data, which are then aggregated and the LLM selects the best response to the query [392]; dual-reasoning, introduced in the study by Liu et al as explained above, combining the strongest reasoning capabilities of both KGs and LLMs to produce more accurate responses [388]; and beam search process as examined in the study by Cao et al, which explains their process as the LLM determining the most appropriate path for reasoning logic based on information provided by KGs. The LLM will iteratively prune and progress through the available KG data in answer a query, conducting multiple searches whenever additional information is required [393]. As one of the fundamental challenges with AI solution adoption for space mission applications is developing trust and interpretability, the creation of an XAI solution, through the utilisation of an LLM, was necessary to facilitate interaction between AI systems and human operators, which can be enhanced through the introduction of KG data.

Using the problem introduced in Chapter 3, a study was conducted to measure the performance of a generalist LLM in answering questions on satellite schedule data and logic, through self-extracting tabular data from a KG. The structure, schema, and

function of the KG are fully explained in Chapter 4. Many studies have investigated the benefits of pre-training, prompt optimisation, and model fine-tuning; therefore, this experiment intends to document the performance of a non-specialist and non-optimised LLM to provide insight into what can be achieved with minimal development and preparation. Additionally, the approach was designed to maximise the number of automated steps that could be implemented to establish a self-contained processing capability with minimal requirements for human intervention for effectiveness. The LLM in this chapter was used for the generation of queries, the extraction of data from a KG, and the answering of queries on a satellite schedule. The results and performance were then assessed for correctness, similarity, validity, and language quality. This provided a broad analysis of how generalist LLMs can perform on domain-specific topics, without conducting optimisation or fine-tuning techniques.

7.3 Methodology

A methodology was designed to derive textual explanations on schedule data instances from user questions in natural language. Figure 7.1 depicts a graphical overview of the proposed approach, broken into three main modules (A, B, and C), followed by an assessment of the generated results.

LLMs have three applications in the method proposed here: for creating a benchmark of natural language queries to use as prompts for the generation of the explanations, to generate code for each query for the creation of the knowledge graph and last, to derive the requested explanations using the output from the queries created, and the generated KG incorporating the constraints of the problem (that have been introduced in Chapter 3.3). OpenAI's GPT-4 was selected as the LLM model due to its ability to generate a sufficient number of tokens. The few-shot learning approach was chosen as the prompting strategy, alongside an established API connector to facilitate the streamlined execution of the experiment. All prompts used through the experiment are located in Tables 7.1, 7.2, 7.3.

Algorithm 4 outlines the automated Python-based process, aligned with Figure 7.1, that combines GPT-4, Google Sheets API, and a local TypeDB server to process the

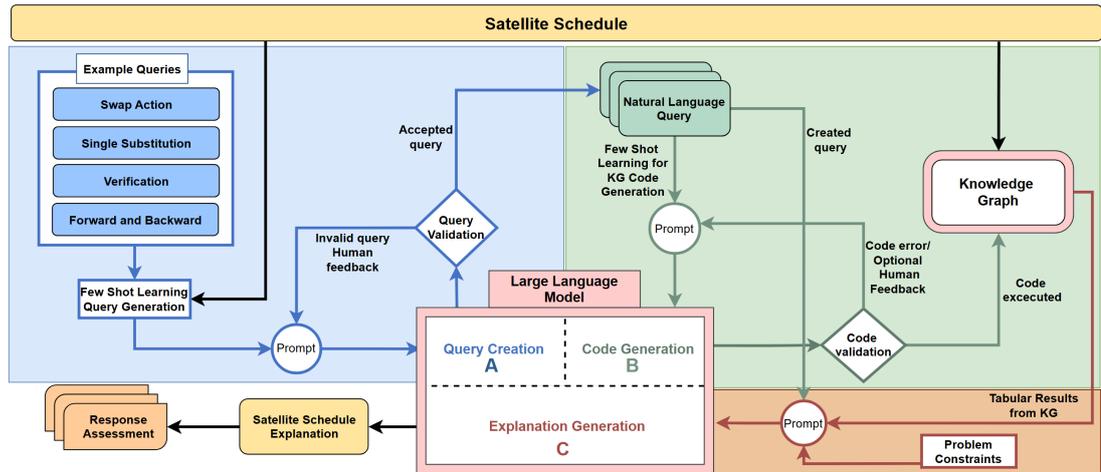


Figure 7.1: Overview of the experiment approach showing LLM use in KG query creation (A), code generation for data extraction (B), and explanation assessment (C), with colour-coded shading.

graph generation of all results. The results were documented in individual sheets of the Google Sheets file, each specific to every query category, code iteration, prompting approach, and extracted tabular data.

Algorithm 4 Process for generating questions and answers using KGs with LLMs for satellite scheduling

- 1: Prompt the LLM with satellite schedule and example queries for query creation (outlined in Section 7.3.1).
 - 2: **while** Query is invalid ¹ **do**
 - 3: Provide human feedback to update the prompt
 - 4: **end while**
 - 5: Prompt the LLM with coded examples, and the generated Query for code generation (outlined in Section 7.3.2).
 - 6: **while** Code inexecutable **do**
 - 7: Provide error to update the prompt with an optional human input
 - 8: **end while**
 - 9: Generate the KG and tabular results
 - 10: Prompt the LLM with the combined tabular results, problem constraints and query (outlined in Section 7.3.3).
 - 11: Generate and assess the response (outlined in Section 7.3.4)
-

¹Incorrect data that isn't reflected in the schedule. For example, the LLM selects actions for specific times that have not been scheduled for those times or suggested times that aren't correct in the schedule.

Initially, human interaction, also known as human in the loop, was necessary to

assist in creating relevant and suitable queries for the scheduling requirements. Once the schedule conditions were achieved, these queries were translated into executable code. This ensured the queries were accurate and aligned with the scheduling constraints and would produce data to allow for explanations.

Following the code creation by the LLM for the generation of the KG, in the event the code was identified as non-executable, an automated feedback loop was triggered, prompting further refinement and adjustments to ensure executability. This loop had the option to return the output error to the LLM for code regeneration or for the human to intervene and customise feedback in order to regenerate the code. Furthermore, the authors manually validated the LLM generated executable code to ensure the necessary data was available to answer the query. This iterative code generation and feedback process was required to develop a robust and reliable process.

The results are organised and populated in tabular form once the KG has been created. This data was then interpreted by the LLM, including the initial query and problem constraints. Two different prompt approaches were used to generate explanations for the query, one of which included an example human-written answer, and the other did not include any example answers. The LLM’s explanations and interpretations, categorised as example-driven local post-hoc natural language explanations [394, 395]; this was determined by the example-based prompting approach, applied to this specific model. The responses underwent an assessment process to measure the correctness, validity, BERTscore, and cosine similarity, which is explained in detail in Section 7.3.4; the approach in assessment combines both human and machine-derived analysis, advancing on previous natural language explanation assessments [396]. An example of the overview of this process is presented in Figure 7.2

7.3.1 Query Creation

A benchmark of queries in natural language needed to be created to effectively validate the method proposed. This was to create a simulation of real-world queries that GSOs could ask but was chosen to be generated through the LLM to remove unconscious bias the authors could introduce in manually written queries, and therefore, the accurate

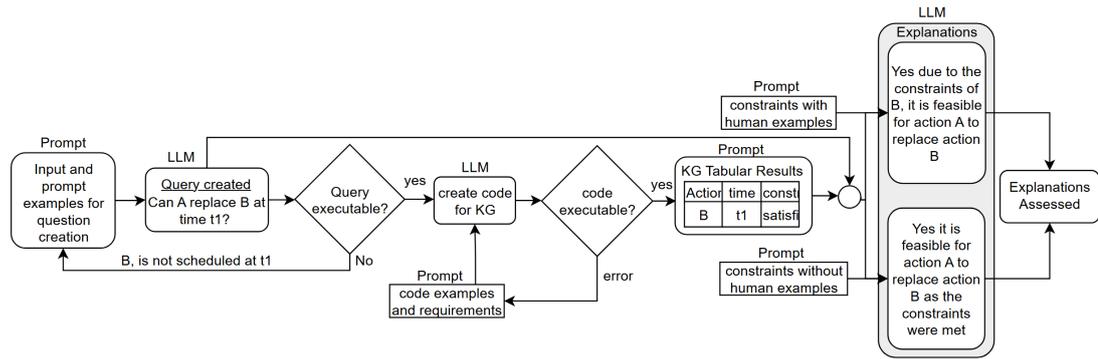


Figure 7.2: An example process for prompting an LLM to generate executable code for retrieving data to facilitate an explanatory response to a decision query.

interpretation of data by the LLM is essential for generating questions pertinent to the satellite schedule. The tabulated scheduled data encompassed over 17,000 distinct time intervals across a 24 hr time period, each associated with corresponding actions, and was provided to the LLM to ensure the types of questions created correlated to the schedule. A categorisation strategy was employed, where four distinct categories of questions were defined for the scheduling problem at hand. For each category, explicit prompts, along with two illustrative query examples, as shown in Table 7.1, were provided as few-shot learning mechanisms to learn the task of paraphrasing alternative queries from the same family.

The process was conducted under supervision to generate 20 questions for each category, creating a total of 80 across all categories. In instances where the authors expressed dissatisfaction in the generation, where a misunderstanding of the scheduled data occurred, constructive feedback was provided. This feedback mechanism enabled the LLM to refine or recreate the questions, aligning them more closely with the schedule. This iterative feedback loop was aimed at enhancing the precision and relevance of the generated questions in accordance with the satellite schedule. An overview of this process is shown in Figure 7.1, module A.

The four query categories are:

- Swap actions - For any replacement of a scheduled action with one of the other possible actions.

Table 7.1: Table reporting example queries used in the prompt as a few-shot learning strategy.

Prompt used across all categories	Query Categories	Few-shot Learning Queries
<ul style="list-style-type: none"> • There are 3 actions: Taking of images, Processing of images, Downlinking of images, and ‘idle’ is created when no other actions can occur. • Any of these actions can replace another within a satellite schedule apart from itself; the start times are shown in the file provided <i>schedule_data</i>. • Note that idle cannot replace another action, but can be replaced. • The column titled “Scheduled action” represents the actions scheduled with the respective time shown in the column labelled start time. • The numbers for the Scheduled actions represent: 1 for processing, 2 for downlinking, 0 for taking of images, and -1 represents idle. • Here are 2 examples of types of questions <i>See Query Categories</i> • Can you make 20 questions similar to the example questions using actual times and actions? 	Swap actions	<ul style="list-style-type: none"> • Can <i>A</i> replace <i>B</i> at time <i>t</i>? • What would happen if <i>C</i> was scheduled instead of <i>B</i> at time <i>t</i>? <p>Create questions for the exchange of actions in a single satellite schedule</p>
	Single Substitution	<ul style="list-style-type: none"> • Can action <i>A</i> be added at time <i>t</i> after action <i>B</i>? • Is it possible for action <i>C</i> to be removed at time <i>t</i>? <p>Create questions for the addition or removal of any action to the schedule.</p>
	Verification	<ul style="list-style-type: none"> • Is it possible for action <i>A</i> to occur at time <i>t</i>? • How many times is action <i>B</i> scheduled up to time <i>t</i>? <p>Create questions to verify actions in a schedule for a satellite.</p>
	Forward and Backward analysis	<ul style="list-style-type: none"> • Will the memory capacity be exceeded at time <i>t</i>? • What happened after action <i>C</i> at time <i>t-1</i> with the memory? <p>Create questions for analysing historical and predictive data in a schedule for a satellite.</p>

- Single Substitution - For the addition or removal of any action in the schedule.
- Verification - For validating the possibility of an action being executed.
- Forward and Backward analysis - For analysis of past or future conditions.

The prompt statement for generating the queries was shared across all query categories, with the exception of the inclusion of the few-shot learning examples that are tailored for each category. These query categories were selected to represent real-world queries an operational system may be asked by a human operator to help understand why particular decisions were made or could not be made.

7.3.2 Code Generation

To ensure the relevant information is available for extraction from the KG by the LLM, a universal prompt was applied across all query categories to assist the LLM in creating executable code to generate the KG. This prompt is detailed in Table 7.2, where sample queries were provided with the respective code and syntax necessary to generate the representative KG that includes the data required to answer the query (note: symbols such as ‘\$’ are used as variables, a requirement for TypeDB as stated in Chapter 4.4). The type of KG used in this thesis is TypeDB, as this KG format was found to provide a balance between graphical clarity and data provision [321,397].

The returned code was automatically used to generate the KG, and the data was extracted in tabular form and provided to the LLM to complete the query response. However, where the generated code was not executable, an error message was collected and fed back to the LLM as an additional prompt with an optional human intervention to provide feedback, and an updated code was requested; this process (highlighted green in Figure 7.1, module B) would continue until an executable code was returned, with each iteration step recorded. The code generation for all questions across all categories was completed three times to observe the pattern and consistency in the performance of the LLM.

7.3.3 Explanation Generation

After retrieving the tabulated data from the KGs, an extensive process was undertaken to incorporate the various constraints of the scheduling problem into the question-answering framework, enabling the LLM to effectively address the original queries. These constraints encompassed conditions such as light exposure, available onboard memory, access to ground stations, and land visibility. Additionally, precise definitions of all acronyms and the standardised format of time were included in the prompts to ensure consistency and clarity.

The alignment process involved synchronising the KG outputs with the generated queries, ensuring that the prompts accurately reflected the necessary constraints and conditions. This step was required for guiding the LLM to understand not only the

Table 7.2: Few-shot data sample used for GPT-4 for code generation

Prompt used across all categories	Sample Queries	Few-shot Learning Sample Answers
<ul style="list-style-type: none"> This is a template code using TypeDB to extract and get all data from a dataset created for a schedule to create a knowledge graph. If there is a question asking for an explanation, only create a code for the Knowledge Graph. This template is fixed and only the variable definition and the 'get' statement can be amended but must maintain the same variable labels. All variables in the questions should be defined at the beginning, after the "match" clause: For example. What would happen if time was 26 seconds? Time would be written like this 2020-12-03T00:00:26; Using the same year-month-day and time format. Note: \$st should only be used if the question specifies it, otherwise it should be removed. Additionally, if there are any questions with a range of variables (you should use the start time (\$st)), the same variable should be used and written like this: \$st >= a; \$st <= b; Also if there is a question stating how many times it is possible for 'a', the "count" statement should be written after the "get" statement for example, how many times is the ground station accessible? \$g isa ground_station, has access \$ac; get \$ac; count; Additionally, when asked about MB, multiply by 1000000. All times requested are in seconds for the fixed day of 2020-12-03, so they must be converted to the correct format for example time 2251 is written as 2020 12-03T00:37:32 If there is a question about swapping actions, at a specific time, only consider the conditions for that specific time by using the exact example below. Create a knowledge graph based on the question and the code template provided. If you need to check other times, feel free to do so and include it in the range. 	<p>What would happen if taking of images was scheduled instead of processing of images at the time of 26 seconds?</p>	<pre> match \$st = 2020-12-03T00:00:26; \$a isa action, has name \$n, has a_timestamp \$at; \$r isa \$ret-type; {\$n contains 'processed'; \$x isa processed, has \$ret; \$r-type type pr_in_memory;} or {\$n contains 'image'; \$x isa image, has \$ret; \$ret- type type im_in_memory;} or {\$n contains 'idle'; \$x isa idle, has \$ret;} or {\$n contains 'downlink'; \$x isa downlinked, has \$ret; \$r-type type total_sent;}; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$ssat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$st(\$a,\$ssat) isa schedule; \$sl(\$mem,\$ssat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$ssat,\$env) isa localisation, has start \$st, has end \$en; get \$n, \$at, \$ret, \$pt, \$ac, \$lv, \$d, \$lat, \$lon, \$l, \$sa, \$st, \$loc, \$id, \$mem, \$g, \$cc, \$max, \$st, \$en; </pre>
	<p>Can taking of images be added at 38 seconds after processing at 26 seconds?</p>	<pre> match \$st >= 2020-12-03T00:00:26; \$st <= 2020-12-03T00:00:38; \$a isa action, has name \$n, has a_timestamp \$at; \$r isa \$ret-type; {\$n contains 'processed'; \$x isa processed, has \$ret; \$r-type type pr_in_memory;} or {\$n contains 'image'; \$x isa image, has \$ret; \$ret- type type im_in_memory;} or {\$n contains 'idle'; \$x isa idle, has \$ret;} or {\$n contains 'downlink'; \$x isa downlinked, has \$ret; \$r-type type total_sent;}; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$ssat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$st(\$a,\$ssat) isa schedule; \$sl(\$mem,\$ssat) isa installation; \$sa(\$g,\$env) isa station_access; \$loc(\$ssat,\$env) isa localisation, has start \$st, has end \$en; get \$n, \$at, \$ret, \$pt, \$ac, \$lv, \$d, \$lat, \$lon, \$l, \$sa, \$st, \$loc, \$id, \$mem, \$g, \$cc, \$max, \$st, \$en; </pre>

nature of the query correctly but also to consider the context and limitations within which the scheduling operates. The comprehensive set of constraints provided a systematic framework for the LLM to analyse the data and generate informed responses. The detailed list of constraints is located in Table 7.3.

To further evaluate the influence of example-based prompting, the LLM was prompted to answer each query twice - once with a human-written example answer and once without any example answer given, generating two responses for every query. The human-written examples were specific to each query category and are listed in Table 7.3. This was performed to measure the influence that providing example answers can have on the performance and comprehension of the LLM. The queries and categories remain the same for both approaches, with an overview reflected in Figure 7.1, module C. The prompt was designed by providing an outline of the problem and purpose of the query with the necessary constraints of the schedule, with the addition of an instruction to be brief to avoid unnecessarily long or repetitive responses.

7.3.4 Assessment Metric

The assessment approach combined expert human analysis with automated quality scoring to calculate a single overall score for each query performance. Once the explanation response from the LLM was produced, it was assessed on the following criteria:

- Correctness: evaluating if the overall affirmative or negative answer to the query is correct.
- Validity: the response is valid based on the schedule constraints.
- BERTscore: structure of the explanation evaluating the semantic completeness and complexity of the generated explanation.

Each was evaluated individually and then averaged to calculate a single final score for overall performance. In addition to analysing the LLM's responses, the generated queries were assessed for similarity by calculating the cosine similarity between each query. This was calculated using the queries within their own category, as well as those from the other categories, providing a score between -1 (completely unidentical) and 1 (completely identical). These metrics were chosen to assess all aspects of a response to allow for holistic and targeted analysis. The results will offer insight into the LLM's ability to generate unique yet thematically identifiable queries and reveal any challenges with word or phrase repetition.

Table 7.3: Prompts containing constraints for answering questions with tabular results retrieved from the Knowledge Graph with human responses for additional responses

Prompts containing constraints of the problem	Query Categories	Few-shot Learning Queries with Answers
<p>This is a problem for a satellite schedule. That has several constraints for executing actions on a satellite.</p> <p>There are 3 actions:</p> <ol style="list-style-type: none"> 1. Image taking, can only occur when there is enough available memory onboard, when land is visible, and when there is sunlight exposure. 2. Processing, can only happen when there is enough available memory and when there are unprocessed images in memory. 3. Downlinking can only happen when there is access to a ground station and there is at least 1 processed image in memory. <p>Idle, only occurs when no other action can take place.</p> <p>pr_in_memory means processed instances in memory, the number of images processed = $pr_in_memory/10.57$.</p> <p>Likewise for total_processed, means the total processed images up to that point (some may have been downloaded already).</p> <p>im_in_mem means "images in memory" This means that some of these images may be processed, so to determine the unprocessed images, it would be "$im_in_mem - (pr_in_memory/10.75)$".</p> <p>current_capacity represents the current memory used, im_total_taken is the total number of images that have been taken up to that point (some may have been processed and downloaded already).</p> <p>Maximum memory (max_capacity) cannot be exceeded at any point in time which is 1920000. "daylight", "land_visibility", and "access" are represented as only True or False.</p> <p>Taking of images is independent of images in memory as images in memory is only required for downlinking. Only the current_capacity and maximum memory, along with land visibility and sunlight can impact when images can be taken.</p> <p>Only 1 action can be executed at each start time. Each action only lasts for 1 time interval.</p> <p>The time is in hh:mm:ss, so it is the same as the question just a different format.</p> <p>Please provide a direct and short answer to the question based on these conditions and using the support file.</p>	Swap actions	<p>Question: What would happen if the taking of images was scheduled instead of the processing of images at the time of 26 seconds?</p> <p>Answer: The criteria for the taking of images action to be scheduled are for land to be visible and in sunlight, as well as enough available memory to complete the action. At time 26, land is not visible, and therefore, the taking of images action cannot be scheduled in place of the processing of images.</p>
	Single Substitution	<p>Question: Can taking of images be added at 38 seconds after processing at 26 seconds?</p> <p>Answer: Referring to the schedule data, the 'idle' action is currently scheduled at time 38 seconds. In order for the taking of images action to be scheduled memory must be available, in addition to land being visible and in sunlight. As all conditions are met at time 38, the taking of images action can be scheduled in place of idle; however, consideration of the future impact on the schedule should be taken into account before adding the action, as memory utilisation will increase.</p>
	Verification	<p>Question: How many times is action downlinking of images scheduled up to time 6079 seconds?</p> <p>Answer: The downlinking action is scheduled to occur 187 times up until time 4411 seconds, according to the provided data.</p>
	Forward and Backward analysis	<p>Question: Will the memory capacity be exceeded at time 9276?</p> <p>Answer: Yes, based on the provided schedule data and the calculation of consumed memory up to time 9276, which is 1,965 Gb, the memory capacity is exceeded at this time.</p>

The *correctness* of the answers was assessed and scored simply as either 0 (*Incorrect*) or 1 (*Correct*), based on the responses judgement of the submitted query. This was conducted by human assessors with an expert understanding of the satellite scheduling problem, ensuring accuracy in establishing if the response aligned with the query and

available data.

Every explanation response was also assessed for the *validity* in respect to the schedule constraints. The assessment needed manual verification by human assessors, due to the automated extraction of pertinent schedule data elements and the association with the constraint function being considered outside of the scope of this study. This was scored using a Likert 5 scale, as discussed in Section 2, with the following grades: Completely invalid (Score - 0), Mostly invalid (Score - 0.25), Partially valid (Score - 0.5), Mostly valid (Score - 0.75), Completely valid (Score - 1).

An answer scored as ‘Completely invalid’ will be entirely impossible for the schedule to tolerate, whether the result of misunderstanding the constraints, returning an invalid time interval, or any other breach of the schedule’s rules and structure. A ‘Mostly invalid’, ‘Partially valid’, or ‘Mostly valid’ response will consist of both valid and invalid statements, where responses with the fewest errors will score higher. A response rated as ‘Completely valid’ would contain only valid judgements and calculations.

The quality of language utilised in each explanation response was also assessed through the *BERTscore* process [398], which returns a score between 0 and 1, where 0 is considered completely unrelated and useless as a response, and 1 would be perfectly aligned with reference material and, in effect, human-like. To calculate the *BERTscore*, each response is compared against a dataset of human-written phrases and statements, with several generalist datasets available for completing scoring exercises. However, as no benchmark dataset for satellite scheduling or operations exists for evaluation, the explanation generated by the model was compared to one manually produced by an expert human annotator. This human answer dataset was comprised of a human-written answer to each question in every category, totalling 80 answers.

7.3.4.1 Final Score

The Final Score (*FS*) was calculated through taking the mean of the *BERTscore*, correctness (*C*), and validity (*V*) score as below:

$$FS = \frac{BERT + C + V}{3} \quad (7.1)$$

The *FS* will return a result between 0 and 1, with 0 being a completely ineffective response and 1 being a perfect response, which will provide insight to the overall level of understanding and accuracy the LLM demonstrated in the responses.

All assessment scores were analysed for comprehension and any patterns that were observed from the responses and discussed in Section 7.4.

7.4 Results

The hyperparameters used for GPT-4 are shown in Table 7.4. These parameters were used for all prompts and applications as described in the previous section. The code used to complete the experiment is located within the GitHub Repository,² and the full record of results is available in the supplemental materials, and the University’s Knowledge Research Portal.³

Table 7.4: Hyper-parameters used for GPT-4

hyper-parameters	Values
temperature	1
max_tokens	1400
top	1
frequency_penalty	0
presence_penalty	0

7.4.1 Query Creation

The 80 queries were generated utilising the tabulated schedule data for each category through an API connection to OpenAI’s GPT-4. Reviewing the responses showed the LLM correctly retrieved the respective time intervals from the tabulated schedule in every case and accurately generated relevant paraphrased questions from the examples provided in the prompt. Table 7.5 represents two examples of the queries automatically generated for each category. Additionally, it was noted that the example questions provided in the prompt gave no indication of which actions were to be queried and served only as a guideline for the generation of queries.

²https://github.com/strath-ace/smart-xai/tree/main/Earth_Observation_Satellite_Case_Study/Knowledge_Graphs

³<https://doi.org/10.15129/a1ec351d-70bc-4cde-9994-dddcfe9323d5>

Table 7.5: Table reporting for each query’s category two examples of automatically generated questions

Family Query Category	Queries generated
Swap actions	-Is it feasible for image processing to take over the time slot allotted for downlinking at 1406? -What kind of impact would result from scheduling image downlinking over image capturing at 1436?
Single Substitution	-Given that image processing is currently scheduled to start at time 20656 seconds, can the downlinking of images be added at time 20661 seconds without disrupting the existing schedule? -Considering there is processing at 3171 followed by taking of images at 3191, could downlinking be scheduled at 3191?
Verification	-Up to time 3201 seconds, how many intervals of idleness are observed in the satellite’s schedule? -What is the earliest time after 4096 seconds at which the satellite can commence the processing of images, given that each action takes 5 seconds to execute and the schedule follows the same pattern observed in the excerpt?
Forward and Backward	-Based on the predictive scheduling, how many more image-processing actions can be accommodated before reaching the memory capacity limit, starting from time 4221 seconds? -After the taking of images at time 3511, how many consecutive processing actions occurred before the next different action was scheduled, and what was that action?

Table 7.6: Cosine Similarity between each question for each category and across each category (Standard deviation in subscript)

Query Categories	Swap actions	Single Substitution	Verification	Forward and Backward
Swap actions	0.619 _{0.06}	0.554 _{0.017}	0.522 _{0.015}	0.472 _{0.009}
Single Substitution	0.564 _{0.036}	0.573 _{0.029}	0.532 _{0.030}	0.450 _{0.022}
Verification	0.471 _{0.064}	0.476 _{0.051}	0.537 _{0.043}	0.503 _{0.013}
Forward and Backward	0.447 _{0.015}	0.406 _{0.021}	0.484 _{0.029}	0.535 _{0.019}

After all queries were generated, each query category was assessed for language similarity with the cosine similarity metric, as mentioned in Section 7.3.4. Table 7.6 shows the cosine similarity scores ranging from 0.406 to 0.619, with each category achieving the highest similarity score for queries within the same category. Little variation was seen in the scores, as a very small standard deviation was recorded, which means the degree of similarity was consistent and predictable. It was expected that the queries within the same category would be more similar, though the middling scores are encouraging for the LLMs ability to generate queries that are not too similar as to become repetitive, while still similar enough to align with a specific theme. The Forward-Backward category presented the lowest overall similarity across all categories, which was understandable as this category has the most varied and complex queries.

Table 7.7: The average number of iterations for each run for each query category

Query Category	Code Generation Run		
	1	2	3
Swap actions	1.70	1.50	1.25
Single Substitution	1.80	1.95	1.70
Verification	1.85	1.90	1.90
Forward and Backward	3.95	3.80	4.20

7.4.2 Query Translation and Code generation

As a result of the closed-loop cycle of code regeneration, the few-shot learning approach, which included only two coded examples, led to multiple iterations of code being required for some of the prompts. The average iterations required to generate successfully executable code, presented by query category, are shown in Table 7.7, where the process was performed in three independent runs to observe consistency. The runs were limited to three to avoid excessive resource consumption; 240 sets of code were generated in total to ensure sufficient iterations were assessed.

All categories required less than two iterations of code to be generated, as an overall average, with the exception of the Forward - Backward category, which averaged between 3.80 and 4.20 across the three runs. This was expected, as the complex nature of the Forward-Backward queries serves as a greater challenge for the LLM to produce correct code and retrieve the appropriate data.

Although the number of necessary iterations was low, potential improvements may be achieved by altering the hyperparameters, introducing additional prompts or prompt optimisation techniques, and including more examples combined with their respective code snippets. Taking one query from each category of queries, Table 7.8 presents examples of the results generated by the LLM for automatic code generation.

To illustrate an example, the code produced for query 1 from the Swap actions category retrieved data featuring the action’s name ‘ $\$n$ ’, the action execution timestamp ‘ $\$at$ ’, the scheduled memory retrieval action ‘ $\$ret$ ’, the contents relation ‘ $\$pt$ ’, ground station access ‘ $\$ac$ ’, land visibility ‘ $\$lv$ ’, daylight conditions ‘ $\$d$ ’, the satellite coordinates: latitude ‘ $\$lat$ ’, and longitude ‘ $\$lon$ ’, installation relation ‘ $\$l$ ’, station access re-

Table 7.8: Queries with their respective code generated

Queries	Generated code
<p>Swap actions</p> <p>Is it feasible for image processing to take over the time slot allotted for downlinking at 1406?</p>	<pre> match \$st = 2020-12-03T00:23:26; # Convert 1406 seconds to the correct time format \$a isa action, has name \$n, has a_timestamp \$at; # Action entity \$ret isa \$ret-type; # Type of result \$n contains 'processed'; \$x isa processed, has \$ret; or \$n contains 'image'; \$x isa image, has \$ret; or \$n contains 'downlink'; \$x isa downlinked, has \$ret;; # Action types \$pt(\$x,\$mem) isa contents; # Contents related to memory \$g isa ground_station, has access \$ac, has \$id; # Ground station \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; # Environmental conditions \$ssat isa satellite; # Satellite \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; # Memory unit \$t(\$a,\$ssat) isa schedule; # Schedule relation \$i(\$mem,\$ssat) isa installation; # Installation relation \$sa(\$g,\$env) isa station_access; # Station access relation \$lloc(\$ssat,\$env) isa localisation, has start \$st, has end \$en; # Localisation with start and end times get \$n,\$at,\$ret,\$pt,\$ac,\$lv,\$d,\$lat,\$lon,\$i,\$sa,\$t,\$lloc,\$id,\$mem,\$g,\$cc,\$max,\$st,\$en; </pre>
<p>Single Substitution</p> <p>Given that image processing is currently scheduled to start at time 20656 seconds, can the downlinking of images be added at time 20661 seconds without disrupting the existing schedule?</p>	<pre> match \$st >= 2020-12-03T05:44:16; # Start time queried \$st <= 2020-12-03T05:44:21; # End time queried \$a isa action, has name \$n, has a_timestamp \$at; \$ret isa \$ret-type; \$n contains 'processed'; \$x isa processed, has \$ret; or \$n contains 'image'; \$x isa image, has \$ret; or \$n contains 'downlink'; \$x isa downlinked, has \$ret;; \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$ssat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; \$t(\$a,\$ssat) isa schedule; \$i(\$mem,\$ssat) isa installation; \$sa(\$g,\$env) isa station_access; \$lloc(\$ssat,\$env) isa localisation, has start \$st, has end \$en; get \$n,\$at,\$ret,\$pt,\$ac,\$lv,\$d,\$lat,\$lon,\$i,\$sa,\$t,\$lloc,\$id,\$mem,\$g,\$cc,\$st,\$en; </pre>
<p>Verification</p> <p>Up to time 3201 seconds, how many intervals of idleness are observed in the satellite's schedule?</p>	<pre> match \$at <= 2020-12-03T00:53:21; # Time queried \$a isa action, has name \$n, has a_timestamp \$at; \$n contains 'idle'; \$x isa idle, has \$ret;; get \$n,\$at; count; </pre>
<p>Forward and Backward</p> <p>Based on the predictive scheduling, how many more image-processing actions can be accommodated before reaching the memory capacity limit, starting from time 4221 seconds?</p>	<pre> match \$st = 2020-12-03T01:10:21; # Start time queried is 4221 seconds, converted to the correct format. \$a isa action, has name \$n, has a_timestamp \$at; \$ret isa \$ret-type; \$n contains 'processed'; \$x isa processed, has \$ret;; # Checking for processing actions \$pt(\$x,\$mem) isa contents; \$g isa ground_station, has access \$ac, has \$id; \$env isa environment, has land_visibility \$lv, has daylight \$d, has latitude \$lat, has longitude \$lon; \$ssat isa satellite; \$mem isa memory_unit, has current_capacity \$cc, has \$id, has max_capacity \$max; # Memory unit to check the capacity limit \$t(\$a,\$ssat) isa schedule; \$i(\$mem,\$ssat) isa installation; \$sa(\$g,\$env) isa station_access; \$lloc(\$ssat,\$env) isa localisation, has start \$st, has end \$en; get \$n,\$at,\$ret,\$pt,\$ac,\$lv,\$d,\$lat,\$lon,\$i,\$sa,\$t,\$lloc,\$id,\$mem,\$g,\$cc,\$st,\$en; </pre>

lation' $\$sa$ ', schedule relation ' $\$t$ ' (facilitating visualisation of connected components in the graph), localisation ' $\$loc$ ', id ' $\$id$ ', memory ' $\$mem$ ', current memory capacity ' $\$cc$ ', maximum memory ' $\$max$ ', followed by start ' $\$st$ ' and end ' $\$en$ ' times respectively. This signified the LLM successfully inferred the requisite attributes and relations necessary

for generating a graph suitable to answer the query and be represented in TypeDB Studio. The LLM would also occasionally add explanatory comments to the code to convey the purpose of the next line of code.

Likewise, looking at the generated code for the second and fourth questions in Table 7.8, the LLM again understood the time range needed, suggesting the prompts were interpreted correctly. This included the request to retrieve the names of the actions, with their respective relations and entities. For the third question, the generated code was formatted differently from the other examples, whereby only the names of the actions were retrieved, combined with the execution time, followed by the ‘count’ function to satisfy the requirements of the query.

Once the code was executable, the results from the generated KG were stored for the next stage of generating explanations. Examples of the tabular data extracted from the KG are shown in Table 7.9, presenting all the data extracted by the code for the queries shown in Table 7.8, which was made available to the LLM. The full extraction of a specific time interval is recorded over three rows of the table, as seen with the data for Swap actions, Single Substitution, and Forward and Backward categories. Each retrieval provided a unique combination of available data, ensuring the relevant data was available for the LLM to answer the query, with a clear separation with the Verification category as only a single count was required to answer the query in the example. These results represent the capability the LLM had to produce executable code for curated data extraction, demonstrating comprehension of the original query and the necessary attributes to answer the question successfully.

7.4.3 Question and Answering over Tabular Data

Following the code execution, the results from TypeDB (such as the examples included in Table 7.9) were used to prompt the LLM in combination with the corresponding query to create a response and an explanation from the LLM; once with a human written example explanation, and once without. The results from the four example questions, for both approaches, are reported in Tables 7.10, 7.11, 7.12 and 7.13.

The response to the Swap Actions query, without an explanation example, in Table

Table 7.9: Tabular results for the query categories

Swap actions						
name	a_timestamp	total_sent	contents	access	land_visibility	daylight
downlinked	2020-12-03 00:23:26	1670.833333	1670.833333	True	False	True
latitude	longitude	id	current_capacity	max_capacity	start	end
58.696137	-42.600072	281	1485346.0	1920000	2020-12-03 00:23:26	2020-12-03 00:23:31
pr_in_memory	total_processed	im_total_taken	dl_size	im_in_memory		
2488.2	1902.250744	1992	560	323.83		
Single Substitution						
name	a_timestamp	pr_size	access	land_visibility	daylight	latitude
processed	2020-12-03 05:44:16	250	False	True	True	52.638976
longitude	id	current_capacity	start	end	total_sent	pr_in_memory
82.768199	4131	1291900.0	2020-12-03 05:44:16	2020-12-03 05:44:21	1898.958333	2468.4
total_processed	im_total_taken	im_in_memory				
2128.534226	2150	254.07				
name	a_timestamp	total_sent	access	land_visibility	daylight	latitude
image	2020-12-03 05:44:21	1898.958333	False	True	True	52.349975
longitude	id	current_capacity	start	end	im_in_memory	im_total_taken
82.628301	4132	1294588.0	2020-12-03 05:44:21	2020-12-03 05:44:26	255.07	2151
total_processed	pr_in_memory	im_size				
2128.534226	2468.4	2688				
Verification						
Total						
0						
Forward and Backward						
name	a_timestamp	total_sent	access	land_visibility	daylight	latitude
processed	2020-12-03 01:10:21	1715.625	False	False	True	-46.979413
longitude	id	memory_unit	current_capacity	start	end	pr_in_memory
130.69365	844	memory_unit	1483258.0	2020-12-03 01:10:21	2020-12-03 01:10:26	2434.6
total_processed	pr_size	im_total_taken	im_in_memory			
1942.057292	250	2041	328.11			

7.10 initially mentions the feasibility of the action exchange and assesses the available memory required for the action to be executed, as well as after the downlinking action has been executed. However, the LLM did not account for whether the downlinking action was replaced with image processing; it should have considered the memory before the downlinking action and not after, as it would not have executed. Furthermore, as there was enough memory available onboard, it would have been feasible for the *processing* action to proceed; therefore, the LLM answered correctly.

For the response when the human example answer is included, also shown in Table 7.10, the LLM again deduced that the replacement of the downlinking action with image processing is feasible, with the response containing similar details on all the satisfied constraints that would allow the swap to take place.

The explanation provided for the Single Substitution category query, without the example explanation, in Table 7.11 only made mention of 1 action when 2 were in fact scheduled. The LLM accurately assessed that the processing action would start at time 5:44:16. However, an image-taking action was scheduled to begin next at 5:44:21, after the processing action finished. This data was overlooked by the LLM, although it

Table 7.10: Explanation created by the LLM for Question 1 (Swap Actions) with and without an explanation example

Example provided	Output to user
<p>Without an explanation example <u>Is it feasible for image processing to take over the time slot allotted for downlinking at 1406?</u></p>	<p>Based on the provided information, the feasibility of image processing taking over the time slot allotted for downlinking at 1406 can be determined by checking if there is enough available memory onboard for image processing. According to the data, the current memory capacity (<code>current_capacity</code>) is 1,485,346.0, and the maximum memory capacity allowed (<code>max_capacity</code>) is 1,920,000. To perform image processing, we need to consider the memory requirements. The processing action uses 250 units of memory. Therefore, to determine if image processing is feasible at 1406, we need to check if the available memory after downlinking is enough for processing. The available memory after downlinking at the specific time will be computed as the current capacity minus the memory used for downlinking (<code>current_capacity - 2800</code>). If the available memory after downlinking is greater than or equal to the memory required for image processing (250), then it is feasible for image processing to take over the time slot allotted for downlinking at 1406. Let's perform the calculation: $\begin{aligned} \text{Available memory after downlinking} &= \text{current_capacity} - 2800 \\ &= 1,485,346.0 - 2800 \\ &= 1,482,546.0 \end{aligned}$ Since the available memory after downlinking (1,482,546.0) is greater than the memory required for image processing (250), it is feasible for image processing to take over the time slot allotted for downlinking at 1406. Therefore, the answer is yes, it is feasible for image processing to take over the time slot allotted for downlinking at 1406.</p>
<p>With an explanation example <u>Is it feasible for image processing to take over the time slot allotted for downlinking at 1406?</u></p>	<p>The scheduling of image processing instead of downlinking at the time of 1406 seconds depends on certain conditions being met. Image processing can only happen if there is enough available memory and there are unprocessed images in memory that need to be processed. Let's examine the conditions at 1406 seconds. According to the data, at this time, "<code>pr_in_memory</code>" is 2488.2, which means there are approximately 232 processed images in the memory ($2488.2 / 10.75$). The "<code>im_in_memory</code>" value is 323.83, meaning there are about 323 total images in the memory, both processed and unprocessed. Therefore, the number of unprocessed images can be calculated as approximately 92 ($323.83 - 232$). This ensures that there are unprocessed images in memory, which is a necessary condition for processing an image. The other condition that needs to be met is that there is enough available memory for processing. The current capacity in memory, according to the data, is 1485346.0 and the maximum capacity is 1920000. Processing an image will only take 250 of memory, which is less than the available memory at this time ($1920000 - 1485346.0$). So, looking at these conditions, it is feasible to replace the downlinking with image processing at 1406 seconds, as there are unprocessed images in memory and there's enough memory space to process an image.</p>

appropriately applied the constraints for the downlinking action required for scheduling and identified that the ground station was not accessible. Therefore, the LLM correctly concluded that the action for downlinking at the requested time was not possible.

The answer given when including the human example answer in the prompt, also shown in Table 7.11, provided a similar response and, likewise correctly asserting the constraints could not support the scheduling of the downlinking action and failing to acknowledge the taking of images was already scheduled at the queried time.

Table 7.11: Explanation created by the LLM for Question 2 (Single Substitution) with and without an explanation example

Example Provided	Output to user
<p>Without an explanation example Given that image processing is currently scheduled to start at time 20656 seconds, can the downlinking of images be added at time 20661 seconds without disrupting the existing schedule?</p>	<p>Starting with image processing at 20656 seconds (5:44:16), it is scheduled to end at 20661 seconds (5:44:21). According to the generated results, if we insert the downlinking at 20661 seconds, there would be no interference with the existing schedule. However, for downlinking to occur, certain conditions need to be met: - There should be access to a ground station at that time, and - There should be at least one processed image in the memory. From the provided results, we see that: 1. The access to the ground station, represented by the value "access", is false at time 20661, meaning there is no access to the ground station. Therefore, downlinking cannot occur. 2. There are 2468.4 units of memory allocated to processed images in memory ("pr_in_memory") at the start of the proposed downlinking period. Since each image uses 10.57 units of memory, there are about 233 processed images in memory. Thus, the condition of having at least one processed image ready for downlinking is satisfied. Summarising, although there are processed images ready for downlinking, the lack of access to a ground station at the proposed time (20661 seconds or 5:44:21) means that downlinking cannot be scheduled at that time without disrupting the existing schedule.</p>
<p>With an explanation example Given that image processing is currently scheduled to start at time 20656 seconds, can the downlinking of images be added at time 20661 seconds without disrupting the existing schedule?</p>	<p>The downlinking of images scheduled at time 20661 seconds after image processing at 20656 seconds cannot be accommodated without disrupting the existing schedule. This is because the action of downlinking can only occur when there is access to a ground station and at least one processed image in memory. The satellite currently has an action scheduled for every second, so adding an additional downlinking action would disrupt this schedule. To accommodate the downlinking action, the schedule would need to be altered to ensure there is sufficient time and memory available on the satellite. Additionally, any changes to the schedule would also need to account for the memory usage of downlinking, which is significant. The downlinking action reduces the memory by 2800, so sufficient memory needs to be available to allow for this reduction without exceeding the maximum capacity of 1920000. Also considering the impact of the change to other planned actions or operations should also take into consideration. Any disruption might lead to delays or cancellation of other planned actions or operations. Hence this action of scheduling downlinking cannot be added immediately after processing without disrupting the current schedule.</p>

Table 7.12 reports the explanations in response to the third question. These results, for both approaches, provided a direct and succinct explanation reiterating the time from the query to further explain that no idle action was scheduled before 00:53:21. The LLM continued to explain it was not scheduled due to the conditions met for the other actions, despite not having any information from the results, as it only produced an integer ('0') in this case. This meant the LLM was capable of applying the conditions stated in the prompt to deduce this conclusion.

Analysing the responses presented in Table 7.13, the query was on predictive scheduling, specifically regarding the potential for accommodating additional image processing actions. The LLM's response without an explanation example showcased a methodical approach, beginning with an assessment of the memory, which involved identifying both the maximum capacity and the remaining available memory. Subsequently, the LLM calculated the unprocessed images in memory and correctly selected the lower

Table 7.12: Explanation created by the LLM for Question 3 (Verification) with and without an explanation example

Example provided	Output to user
<p>Without an explanation example Up to time 3201 seconds, how many intervals of idleness are observed in the satellite’s schedule?</p>	<p>The satellite’s schedule did not have any period of idleness up to time 3201 seconds. This means that during the given duration, the satellite was always occupied with one of the three actions - Image taking, Processing, or Downlinking. There was always sufficient memory, appropriate conditions, and tasks (such as images to be processed or downlinked) that kept the satellite busy. Hence, no intervals of idleness were observed.</p>
<p>With an explanation example Up to time 3201 seconds, how many intervals of idleness are observed in the satellite’s schedule?</p>	<p>The satellite schedule does not have any idle intervals up until time 3201 seconds. The term “idle” refers to periods where the satellite is not performing any operation. In this case, it means that the satellite is either capturing images, processing them, or downlinking them to a ground station throughout this duration. Hence, the number of idle intervals is 0.</p>

value between the remaining memory and the number of unprocessed images, factoring in the assumption of no additional images being taken or downlinked after the time 4221 (01:10:21) as stated in the query. From this analysis, the LLM accurately derived the maximum number of feasible actions and correctly asserted that, based on predictive scheduling from time 4221 seconds, approximately 97 more image-processing actions could be accommodated before running out of unprocessed images in memory, with no risk of hitting the memory capacity limit. This proficient response underscores the LLM’s effectiveness in handling intricate queries related to predictive scheduling and resource constraints.

The answer provided from the prompt including the human-written example, did not achieve the same comprehension of the scenario constraints, claiming that the image processing action could be executed 1746 times before reaching the memory limit. This demonstrates that prompt engineering, by including human-derived answer examples, is not infallible and further pre-training or prompt optimisation may be required to improve accuracy overall.

7.4.4 Question and Answering Assessment

Each response from all query categories was assessed individually and independently by two human assessors, who completed the manual assessments of correctness and validity simultaneously, with separate and independent scores. The average correctness (C) and

Table 7.13: Explanation created by the LLM for Question 4 (Forward and Backward) with and without an explanation example

Example provided	Output to user
<p>Without an explanation example</p> <p>Based on the predictive scheduling, how many more image-processing actions can be accommodated before reaching the memory capacity limit, starting from time 4221 seconds?</p>	<p>Based on the given information, at the start time of 4221 seconds, the current memory capacity is 1483258.0 and the maximum memory capacity is 1920000. To determine how many more image-processing actions can be accommodated before reaching the memory capacity limit, we need to consider the following constraints:</p> <ol style="list-style-type: none"> 1. Available Memory: Each image processing action uses 250 units of memory. Therefore, the remaining available memory can be calculated as: Remaining Memory = Maximum Memory - Current Capacity = $1920000 - 1483258.0 = 436742.0$ 2. Unprocessed Images: To determine the number of unprocessed images in memory, we need to subtract the processed images from the total images in memory. The number of processed images can be calculated as: Number of Processed Images = $pr_in_memory / 10.57 = 2434.6 / 10.57 \approx 230.39$ Therefore, the number of unprocessed images can be calculated as: Number of Unprocessed Images = $im_in_mem - (\text{number of processed images}) = 328.11 - 230.39 \approx 97.72$ 3. Image-Processing Actions: Each image-processing action can process one image. Therefore, the maximum number of image-processing actions that can be accommodated is limited by either the remaining memory or the number of unprocessed images, whichever is smaller. The maximum number of image-processing actions can be calculated as: Maximum Number of Image-Processing Actions = $\min(\text{Remaining Memory} / \text{Size of Each Image Processing Action}, \text{Number of Unprocessed Images}) = \min(436742.0 / 250, 97.72) \approx \min(1746.97, 97.72) \approx 97.72$ <p>Therefore, based on the predictive scheduling, starting from time 4221 seconds, approximately 97 more image-processing actions can be accommodated before reaching the memory capacity limit.</p>
<p>With an explanation example</p> <p>Based on the predictive scheduling, how many more image-processing actions can be accommodated before reaching the memory capacity limit, starting from time 4221 seconds?</p>	<p>The current memory capacity at the starting time of 4221 seconds (or 01:10:21) is 1483258.0Gb. Each image processing action takes up 250 Gb of memory. The maximum memory capacity is 1920000 Gb. Therefore, the remaining available memory at time 4221 seconds is $1920000 - 1483258.0 = 436742.0$ Gb. Now, we need to calculate how many more image-processing actions can be accommodated without exceeding the memory capacity limit. To do this, we divide the remaining memory by the amount of memory each action takes, which gives us $436742.0 / 250 = 1746.97$ roughly. Thus, approximately 1747 image-processing actions could be accommodated before reaching the memory capacity, starting from time 4221 seconds, however only whole actions can be performed so the answer is 1746 actions can be performed.</p>

validity (V) scores were calculated from the combined assessor scores, with the addition of the BERTscore (F1) (BrF1) and the subsequent aggregated Final Score (FS), which were computed for each query and their average reported per query category in Table 7.14.

Table 7.14: The mean values and standard deviation scores for all question families with and without example explanation

Question Family	Without Explanation Example				With Explanation Example			
	C	V	BrF1	FS	C	V	BrF1	FS
Swap Actions	0.425 _{0.482}	0.781 _{0.259}	0.450	0.552	0.700 _{0.458}	0.813 _{0.242}	0.520	0.677
Single Substitution	0.925 _{0.238}	0.856 _{0.263}	0.527	0.769	0.900 _{0.300}	0.775 _{0.295}	0.560	0.745
Verification	0.650 _{0.443}	0.719 _{0.369}	0.542	0.637	0.650 _{0.477}	0.694 _{0.338}	0.578	0.641
Forward and Backward	0.300 _{0.430}	0.450 _{0.317}	0.545	0.432	0.400 _{0.490}	0.450 _{0.327}	0.589	0.480
Overall Average				0.598				0.636

Only one of the explanation responses failed to provide a correct answer, with all others demonstrating a high level of comprehension with ‘Mostly correct/valid’ or ‘Completely correct/valid’ scores across all queries, with the exception of the human example

prompt answer in Table 7.13. Overall, the results outline the potential for a generalist LLM to communicate in detail on domain-specific topics without dedicated specialist pre-training or prompt optimisation. Notably, the response to the Forward and Backward example query, from the prompt without the example explanation, demonstrated the LLM's ability to understand the context of the schedule and data by recognising that in addition to calculating the available memory, the question also required calculating the remaining unprocessed images in memory to provide the most accurate and factual answer.

Although a high standard deviation was seen for all correctness and validity scores, the 'Single Substitution' query category achieved a significantly higher average scoring of 0.925 and 0.900 for correctness, in addition to the highest (0.856) and second-highest (0.775) in validity, for the responses without and with the human example answer, respectively. The LLM outperformed the correctness results (referred to as accuracy in the study) in previous similar assessments, albeit outside of the domain of satellite scheduling, where the best-observed scores (without pre-training) were 0.816 and 0.794, respectively [399, 400].

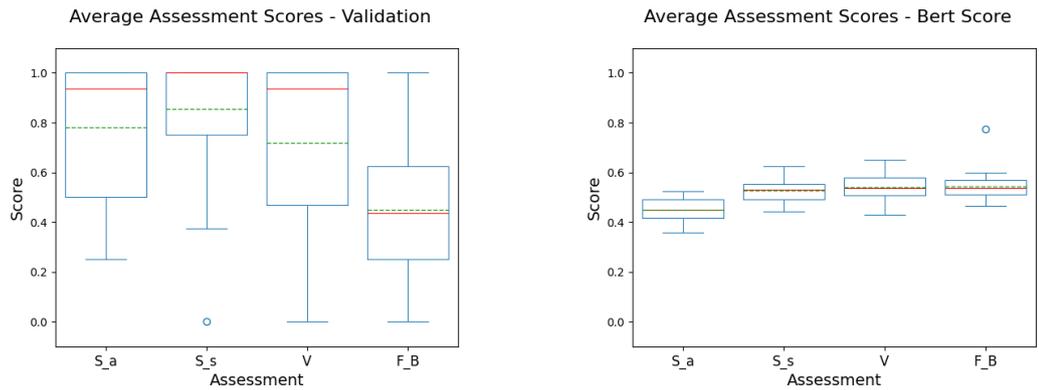
The variance in assessment scoring between the prompt including the human-written example answer and the prompt without was not found to be significant, with the overall final score average improving to 0.636 from 0.598 (an increase of 6.4%) when including the example explanation. However, noteworthy increases in the performance of responses to the Swap Actions and Forward and Backward query categories were observed. The Swap Actions category had a significant increase in average accuracy (from 0.425 to 0.700 (64.7% increase)), while the Forward and Backward category showed a more modest increase (from 0.300 to 0.400 (33.3% increase)). One possible reason for the increase may be the outcome of inferred context to queries and schedule data by the LLM from analysing the provided example answer. Furthermore, the BertScore (F1) increased for all query categories by an average of roughly 8%, which, while not a substantial increase, does indicate the LLM interpreted the example explanation and returned a response more closely aligned to human-written responses.

The standard deviation for correctness, accounting for the binary scoring, resulted

in the highest observed standard deviation. As the generalist LLM was not pre-trained on satellite scheduling concepts, it demonstrated gaps in comprehension that can occur through a few-shot prompting process, even when examples are included. Overall, the Single Substitution and Verification query categories scored highly in both correctness and validity on average, confirming that the LLM was capable of interpreting complex data and concepts with frequent accuracy, as shown by comparing the average assessment whisker plots for each query category in Figures 7.3a, 7.3b, 7.3c, and 7.3d where the mean is represented with broken green lines and the median as a solid red line. In each plot, S_a is the Swap Actions query category, S_s is Single Substitution, V is Validation, and F_B is Forward and Backward.

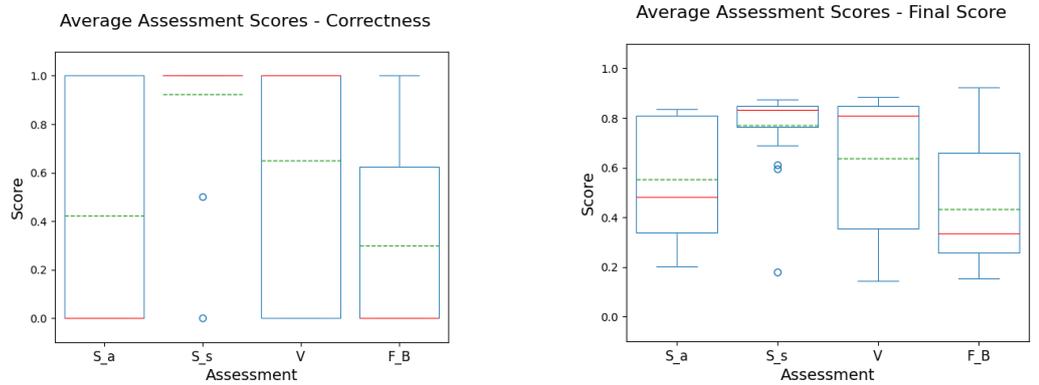
Conversely, as stated previously, the ‘Single Substitution’ query category scored the highest for Correctness and Validity, shown in Figures 7.3a and 7.3c, with 90% being correct with only two scores returning an average of 0.5 and 0 (as there were two assessors) and 70% scoring full validity. However, the ‘Forward and Backward’ query category recorded the lowest scores for both correctness and validity, with averages of 0.300 and 0.450, respectively. This was significantly lower than other query categories, demonstrated through the ‘bottom-heavy’ nature of the distribution in Figures 7.3a, 7.3c, and 7.3d. There appears to be a relation with the context of the ‘Forward and Backward’ queries, which were often centred around broad temporal considerations. This could have caused difficulty for the LLM in generating an effective code to extract the information necessary to construct a fully informed answer and explanation. Even in instances where the full and correct information was successfully extracted from the KG for these queries, the LLM still had challenges comprehending the correct explanation.

The patterns in assessment scoring for the responses, including the example explanation, were very similar to the response without, as shown in Figures 7.4a, 7.4b, 7.4c, and 7.4d. Verification and Single Substitution query categories both scored highly again (it can be noted that Single Substitution also achieved a percentage of 90% being correct, with the remaining 10% scoring 0), with the addition of Swap Actions now scoring 0.700 on average for correctness as mentioned previously. Forward and Backward also



(a) Average Assessment scores for Validation across all categories.

(b) Average Assessment scores for Bert Score across all categories.



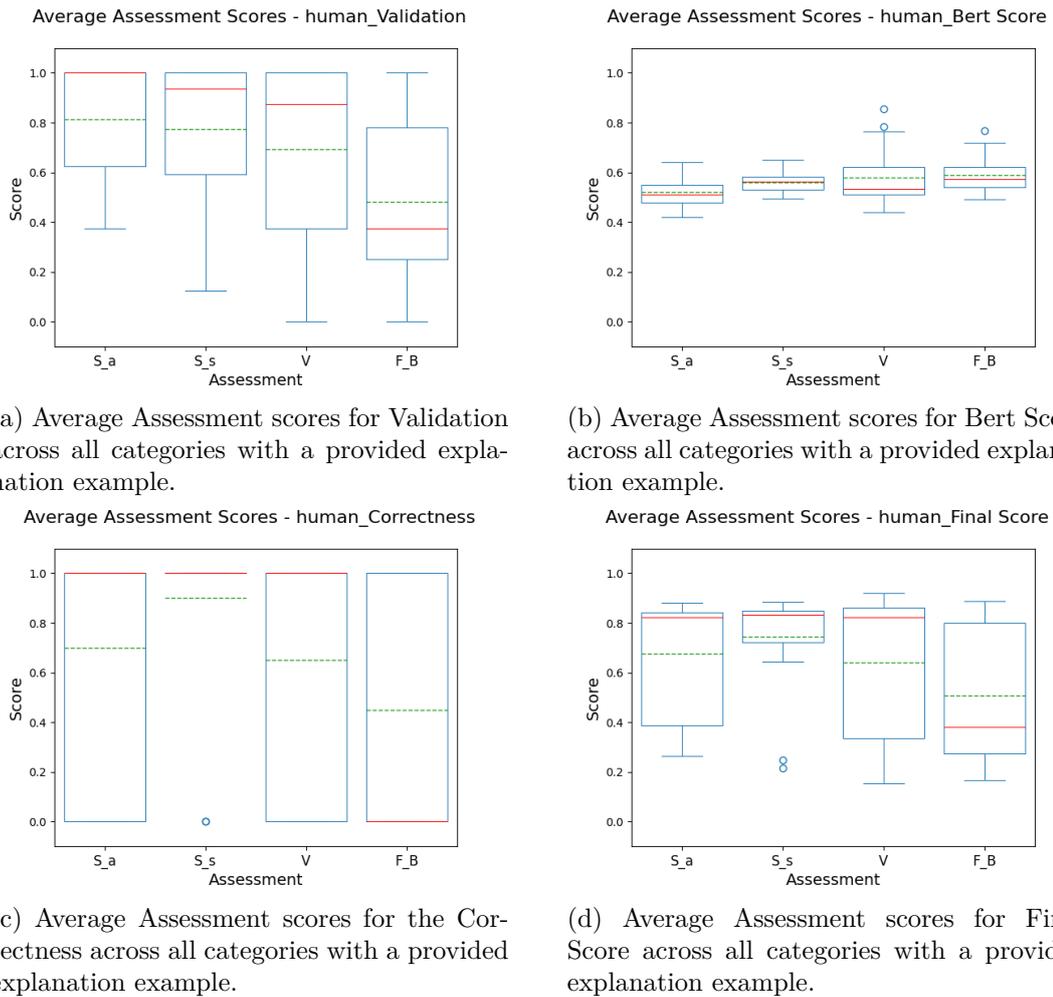
(c) Average Assessment scores for the Correctness across all categories.

(d) Average Assessment scores for Final Score across all categories.

Figure 7.3: Average Assessment scores for Validation, Bert, Correctness for each query category, including final score without explanation examples provided. S_a is the Swap Actions query category, S_s is Single Substitution, V is Validation, and F_B is Forward and Backward.

improved, by a smaller degree, suggesting that including an example explanation can improve correctness, though validity scores were largely unaffected.

Despite the variance observed in the quality of explanations, the LLM demonstrated consistent accuracy and capability with interpreting the provided data. The responses demonstrated both correctness and validity, with scores of 0.58 and 0.7 without the example provided, and 0.66 and 0.68 with the example provided, respectively, particularly with the more direct and structured questions, matching, if not exceeding, the performance of open-domain studies as highlighted earlier in this Section. There was also no compelling separation between the scores of correctness or schedule validation



(a) Average Assessment scores for Validation across all categories with a provided explanation example.

(b) Average Assessment scores for Bert Score across all categories with a provided explanation example.

(c) Average Assessment scores for the Correctness across all categories with a provided explanation example.

(d) Average Assessment scores for Final Score across all categories with a provided explanation example.

Figure 7.4: Average Assessment scores for Validation, Bert, Correctness for each query category, including final score using the provided explanation example with explanation examples provided. S_a is the Swap Actions query category, S_s is Single Substitution, V is Validation, and F_B is Forward and Backward.

from either approach, revealing that the LLM was able to, in most cases, comprehend the context of the question, the rules and constraints required by the schedule and the available data extracted from the KG.

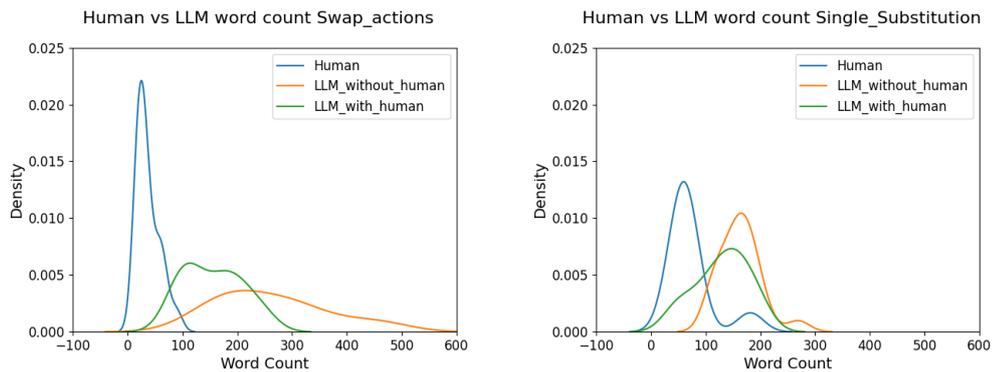
Though the LLM responses had broad distribution in correctness and validity scores, the BERTscore (F1) assessment returned a largely consistent result, with the majority of scores between 0.4 and 0.6, regardless of prompting approach; as the overall average scores were 0.562 and 0.516 for the prompts with and without the example explanation respectively, visible from the plots in Figures 7.3b and 7.4b. This outlines that, on

average, the LLM was generating explanation responses containing around 50% of the language and phrases used in the human explanations reference material, which is positive given no pre-training was performed.

The responses from the LLM were often exhaustive and verbose in their use of language, however, even where the question required minimal exploratory analysis. As the human responses were more concise, this was likely the main cause for the scores for BERTscore that the responses achieved. Several responses, however, did score significantly higher than the overall average for BERTscore (0.539 combined for both prompting approaches), with 8 answers achieving scores between 0.667 and 0.855. Notably, seven of these answers were responses from the prompt, which included the example explanation; four from the Verification query category and three from ‘Forward and Backward’. This aligns with the observation that all query categories produced a small improvement in BERTscore for the prompt including the example explanation, with the average increasing from 0.516 to 0.562, suggesting that providing an example in a prompt can assist the LLM in generating more humanlike responses.

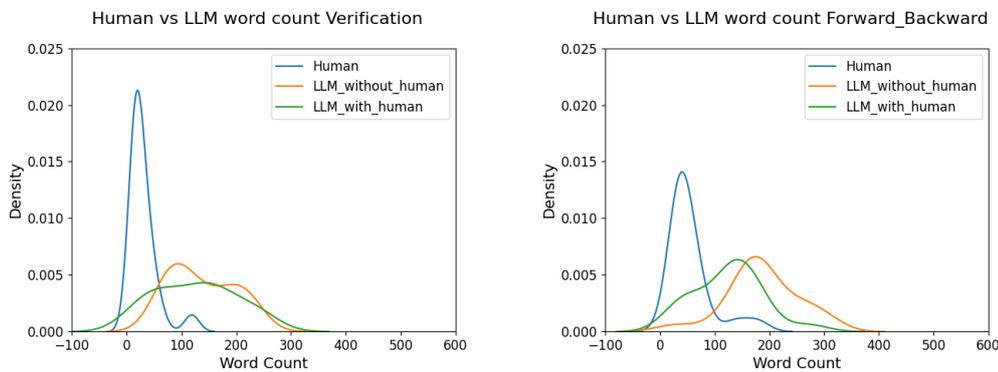
The density of the word count plots for both prompt approaches and each query category, shown in Figure 7.5, conveys that the average human-written explanations were consistently shorter than the LLM explanations, with the average word count for the human explanations being 48 across all categories. Whereas the average word count for the LLM was 188 and 134 for the prompt approach without and with the example explanation, respectively. The most closely aligned LLM responses were those in the Single Substitution and Forward and Backward query categories. However, the separation from human-written answers demonstrates the overuse of language and irrelevant detail in responses from the LLM, even if they are correct and valid. Ultimately, the length of a response from the LLM appeared to be uninfluential on the accuracy and quality of an explanation.

In Table 7.15, is a presentation of the scored results for the responses in Tables 7.10, 7.11, 7.12, and 7.13 for the four respective queries. These scores contain a reflection on the LLM’s use of language for the explanations of these questions. Regardless of the prompt approach, they were fairly even across the query categories, except for the



(a) Word count comparison between the Human and GPT-4 for the Swap-actions category.

(b) Word count comparison between the Human and GPT-4 for the Single Substitution category.



(c) Word count comparison between the Human and GPT-4 for the Verification category.

(d) Word count comparison between the Human and GPT-4 for the Forward and Backward category.

Figure 7.5: Word count of Human vs GPT-4, with and without explanation prompt examples of the four categories

Swap Action questions, with both prompting approaches, which scored considerably lower than the other areas. In these specific cases, this was the result of a significant mismatch between the human reference explanation and the LLM generated response. Though the differences between the BERTscores are very minor across the question categories, the more complex questions of the Verification and Forward and Backward categories lent themselves better to the lengthy automated responses from the LLM, as the human explanations were also longer, so scored slightly better on average, as shown in Table 7.14.

Given the reliance on reference material unique to this study, which did not include the human answers to the queries used in the experiment, for the calculation of

Table 7.15: Assessment of the four exemplary questions for responses in Tables 7.10, 7.11, 7.12, and 7.13

Queries	Without Explanations				With Explanations							
	C	SV	BrfF1	FS	C	SV	BrfF1	FS				
Swap actions												
1	Is it feasible for image processing to take over the time slot allotted for downlinking at 1406?				1.000	0.875	0.419	0.765	1.000	1.000	0.464	0.821
Single Substitution												
2	Given that image processing is currently scheduled to start attime 20656 seconds, can the downlinking of images be added at time 20661 seconds without disrupting the existing schedule?				1.000	0.750	0.583	0.778	1.000	0.375	0.567	0.647
Verification												
3	Up to time 3201 seconds, how many intervals of idleness are observed in the satellite’s schedule?				1.000	1.000	0.547	0.849	1.000	1.000	0.568	0.856
Forward and Backward												
4	Based on the predictive scheduling, how many more image-processing actions can be accommodated before reaching the memory capacity limit, starting from time 4221 seconds?				1.000	1.000	0.599	0.866	0.000	0.375	0.667	0.347

BERTscore and the lack of any domain-specific reference text, there were no reasonable means of comparing these results with the performance observed in previous studies. The results described here highlight the capability of a generalist LLM to understand and produce domain-specific responses effectively and consistently, although there is room for improvement to better align with the author-derived reference material.

The responses in the Single Substitution category achieved the highest average final score for both prompting approaches (0.769 with the example explanation and 0.745 with, as shown in Table 7.14), primarily due to the strong human assessment scoring. Conversely, the Forward and Backward query category recorded the lowest average final scores (0.432 and 0.480, respectively), due to the low human assessment performance. The structure and wording of queries could be investigated in future research

to determine the impact on performance and examine opportunities for improvement.

The final score results overall, shown in the plots within Figures 7.3d and 7.4d, establish that the LLM, for most query categories, is capable of interpreting complex and technical data and concepts, with the ‘top-heavy’ distribution of scores, regardless of prompting approach. Except for the predictive nature of questions within the Forward and Backward category, the majority of responses from the LLM were accurate, well-written, and technically valid, demonstrating a useful foundation on which further development and research can build and challenge on broader domain-specific requirements.

7.5 Conclusion

The purpose of this experiment was to examine how effective the combination of KGs and LLMs was at facilitating XAI for a satellite scheduling problem. Initially, the generation of suitable queries was achieved by using LLMs to create queries assigned to different categories under the supervision of the authors, utilising GPT-4.

The critical aspects of the satellite scheduling data were detailed utilising the framework of a KG, proceeded by the adoption of the few-shot prompting method, which allowed the model to respond with executable code to produce the KG containing the necessary data to answer the query.

The results from the generated KG were supplied as tabular data, enabling the LLM to analyse the information in conjunction with additional prompts entailing constraints for the satellite scheduling problem and the question to create explanation responses. Two prompts were used to request the explanation, one including a human-written example explanation and one without any examples.

When analysing the results, the LLM was found to be proficient in generating appropriate questions related to each category, though this did require repeated iterations to produce completely accurate responses, which was identified as an area for future improvement. Reviewing the code produced and executed, the KGs displayed and generated relevant information required for answering the questions, even where multiple codes were necessary for a single question. The LLM proved to have capability, as well

as further potential, for providing explanations through understanding the extracted information, the context of the question, and the constraints of the schedule. There were variances in correctness, validity, and use of language that introduced inconsistency into the results, but given that the LLM is not domain-specific, this was expected. Additionally, including a single example explanation did not significantly improve performance, although it encourages further investigation into more expansive use of examples or alternative prompt optimisation approaches to determine if a greater improvement can be achieved. Allowing for corrective prompting and multiple explanation iterations would possibly mitigate this limitation, even without dedicated pre-training, which should be considered for future work.

The overall findings have demonstrated significant potential and current ability in generalist LLMs and KGs to generate both basic and complicated questions, and graphical interfaces for satellite schedule data, with the added capability to answer and justify responses accurately. Considerable flexibility and scalability were observed by the KGs, which were able to produce graphs from single-field extractions to several hundred connected data points without impacting interpretability. Assessors were able to narrow or increase query sizes easily, driving enhanced context that enables more informed choices when considering an update to the schedule. When aligned with the LLM generated explanations, substantial knowledge is accessible to users, providing a deep comprehension of why scheduling decisions were made and the consequences of altering, which allows both proactive and reactive engagement. These results enable examination in real-world applications of AI systems for satellite scheduling (and overall operations) situations and will provide developers and quality assurance officers a direct means of understanding the system's decisions, which facilitates the application in real-world satellite missions.

Chapter 8

Generalisation of Large Language Model Applications to Scheduling

8.1 Chapter Overview

This chapter provides a detailed study and analysis of using LLMs with benchmark schedules in tabular form and assessing the capabilities in answering different categories of queries to determine their capabilities for use in XAI applications with different schedules. The benchmark scheduling datasets, LLMs choices and applications were taken from *C. Powell and A. Riccardi, “Generating Textual Explanations for Scheduling Systems Leveraging the Reasoning Capabilities of Large Language Models”* [44] published in the Journal of Intelligent Information Systems [peer-reviewed].

The chapter is made up of five sections:

- The Literature Review (8.2) contains a review of the research on XAI techniques for the application to scheduling problems.
- The Schedule Benchmark Set (8.3) outlines the details of the selected benchmark scheduling problems.
- The Methodology (8.4) sets out the approach for the experiment, including the configuration of the LLM, query creation, prompting technique, and the metrics used to assess performance.

- The Results (8.5) explore how the differing sizes and complexities of scheduling problems impact the performance of the LLM in answering queries, and consider any potential connection between observed metrics.
- The Conclusion (8.6) summarises the findings from the results and observed patterns in behaviour, determining the effectiveness of the approach for real-world applications.

8.2 Literature Review

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) [401]. 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.

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 [380, 402, 403].

In the context of 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 [404]. 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 prioritises tasks or resources in alignment with individual choices, allowing users to comprehend the rationale behind the prioritisation of specific elements in the schedule [405].

A comprehensive survey by Gajane et al. discusses practical applications of 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, Gajane et al. 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) [406]. 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 [407].

The study by Puiutta et al. 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 [407]. 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 Counterfactual (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 [408]. To apply CF explanations to scheduling, further research can explore XAI techniques that prioritise 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 optimised using AI technologies, with a particular focus on the application of 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 [409].

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 chapter. 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 [410], 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 [411]. 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 Figure 8.1, the LLM can extract significant values from the table using self-augmented prompting, which aids in generating improved answers for downstream tasks.

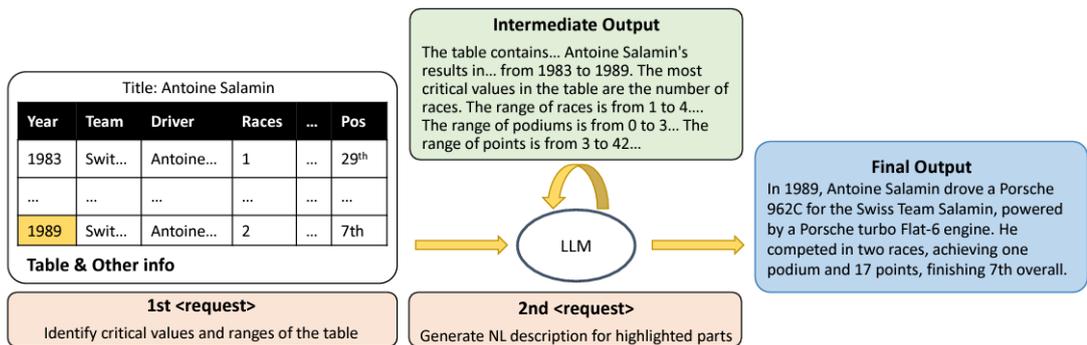


Figure 8.1: Illustration of self-augmented prompting [411].

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 analysed tabulated data from a scheduler based on fixed and structured queries. These queries were targeted at specific scheduled tasks and assessed the feasibility of replacing them with alternative tasks without impacting the schedule, known as a SEP concept. The generated prompts were fed to the LM along with the initial query to aid with generating an appropriate response [412].

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.

The task-based nature of satellite activities as part of achieving mission objectives, the scheduling technique would need to facilitate this inherent requirement; therefore, Job Shop Scheduling was found to fit the requirements best. As such, the history of Job Shop scheduling was examined, which identified the extensive research by E. Taillard on job shop scheduling, in which 260 benchmark scheduling problems of varying sizes were defined. These benchmark schedules have been utilised in research for over 30 years with over 3,000 citations [80], building a well-established baseline that can be applied to any task-based, job-shop-aligned scheduling problem. Because of this, a selection of differently sized schedules was selected from the original research, where the schedule data was available, to form the basis of an XAI experiment to demonstrate the capabilities in a neutral, non-domain-specific setting for a baseline of results to compare against future research experiments.

8.3 Schedule Benchmark Set

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 uninterruptible duration (or processing time), randomly determined between 1 and 99 [84, 413], that must be completed for the job to finish, and machines can only process one job at a time.

The objective in solving the scheduling problem is to find the arrangement of jobs that returns the shortest makespan, which is the time required for all jobs to complete. In E. Taillard's original job shop benchmark set of scheduling problems, the varied combination of jobs and machines was between 15 and 100 and between 15 and 20, respectively (ensuring there are never more machines than jobs). 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 [414]; 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 [415]. From the presented information, at the time of writing, of the 80 Job-Shop Scheduling benchmark problems, 21 remain with non-optimal upper bounds [414].

As previous research in this thesis had explored schedules with non-optimised feasible solutions, the decision was made to use a mixture of optimal and non-optimal feasible solutions to experiment with the benchmark schedules. Ten schedules were selected from E. Taillard benchmarks, which were solved by different schedulers and researchers, where 80% of the schedules had optimal solutions, and the remaining 20% were feasible solutions.

The optimal solutions selected were TA03 through TA09, derived by [416], with makespans between 1175 and 1274, where each schedule consisted of 15 jobs and 15 machines. An optimal solution with 20 jobs and 15 machines, and a makespan of 1342, from benchmark TA13 was also selected, solved by [417]. The same source was used for the first feasible solution from benchmark TA26, formed of 20 jobs and 20 machines and a makespan of 1647. Finally, a feasible schedule for benchmark TA48, containing 30 jobs and 20 machines, was selected from the solution found by [414]. The selections are summarised in Table 8.1 and as a visual representation of one of the schedules, the Gantt chart for schedule TA13 is displayed in Figure 8.2.

While these solved schedules were presented as the best available solutions to the benchmarks, even for the non-optimal solutions, additional research was conducted to validate the selections. Experiments by [418] matched the optimal results for benchmarks TA01 through TA10, however, was unable to achieve optimality for the benchmarks in the range TA21-TA30. Similarly, [419] was able to match much of the smaller benchmarks optimal solutions with their Hybrid Particle Swarm Optimisation approach, and although they completed solutions for all 80 Jop-Shop benchmarks by E. Taillard, the results did not consistently achieve optimality. Tests were also conducted to complete the set of benchmarks [420], yet their solver was unable to match any of the known optimal or best solutions. A Genetic Algorithm approach was applied to the solver [421], but achieved an optimal solution with only 2 of the 30 (TA01-TA30) benchmarks used in their experiments. As the available research could not demonstrate improved solutions as an alternative, the selected benchmark solutions were used.

Table 8.1: Schedules used with their respective makespans [414, 416, 417, 422]

Schedule Label	Schedule Type jobs j by machines m	Makespan	Optimised Solution
TA03	$15j \times 15m$	1218	Yes
TA04	$15j \times 15m$	1175	Yes
TA05	$15j \times 15m$	1224	Yes
TA06	$15j \times 15m$	1238	Yes
TA07	$15j \times 15m$	1227	Yes
TA08	$15j \times 15m$	1217	Yes
TA09	$15j \times 15m$	1274	Yes
TA13	$20j \times 15m$	1342	Yes
TA26	$20j \times 20m$	1647	No
TA48	$30j \times 20m$	1956	No

8.4 Methodology

Once the benchmark schedules were selected as the baseline data for the experiment in testing LLM explainability, a series of assessments were designed to measure the comprehension, reasoning, consistency, completeness, and conciseness of responses.

The LLM-based approach was designed to answer questions on 3 separate and in-

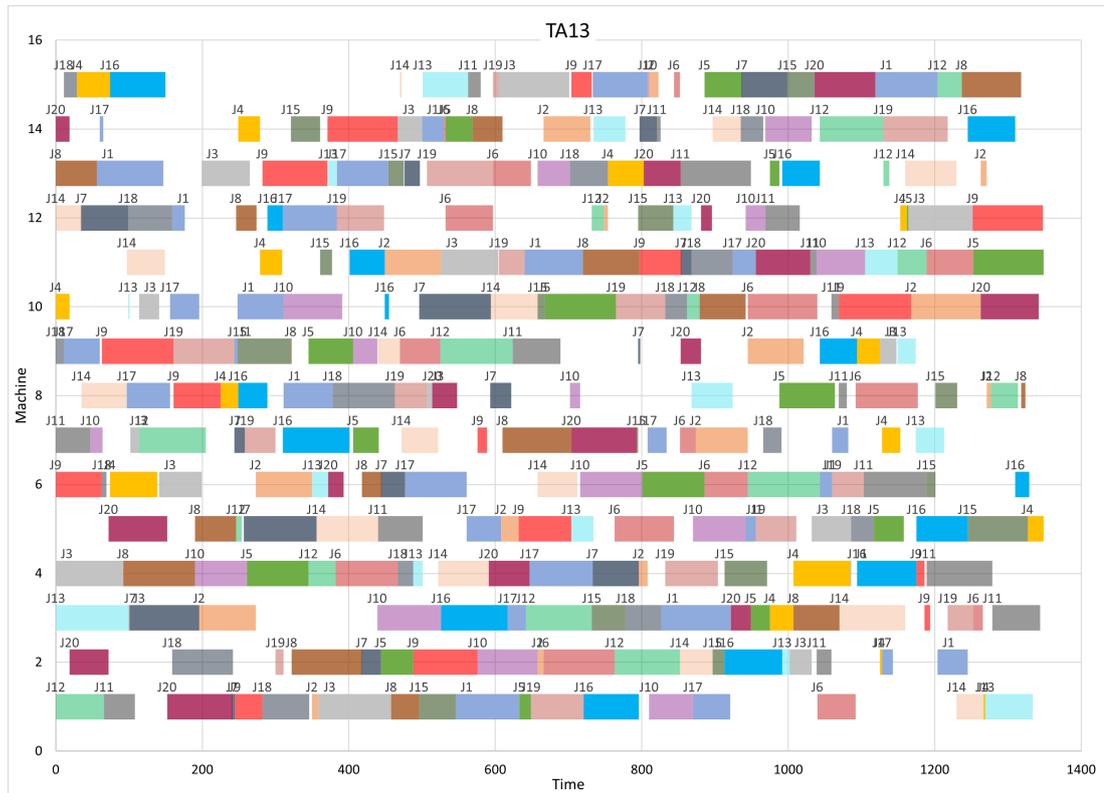


Figure 8.2: A representation of schedule TA13 derived by Henning et al. [417] proposed by E. Taillard [422].

dependent categories of questions, to understand the capacity for analysing differing tabular data from the solved benchmark job shop schedules. The answers to the questions returned by the LLM must adhere to the constraints by which the schedule data is derived; as such, no job sequencing rules were enforced, and jobs were free to be positioned in any order on the machine. However, the same job cannot occur on different machines at the same time, and only one instance of any job will be processed at any one time.

The supplied scheduled data was split into two sets of categorisation: Optimised and Non-Optimised, and Smaller and Larger. The Optimised schedules are those that have an available optimal solution, whereas the Non-Optimised schedules only have a feasible solution available. The Smaller schedules are variants of the $15j \times 15m$ sized benchmarks of which there are 7 in total, with the Larger schedules being 1 instance of each of the following benchmark sizes: $20j \times 15m$, $20j \times 20m$, and $30j \times 20m$; where j

stands for jobs and m represents machines.

OpenAI’s GPT-4 and Meta’s LLaMA were introduced as generalist LLMs, to assess the feasibility of amending the jobs within the schedule, where all unique questions are derived from the questions presented in Table 8.2, along with the question prompt in Table 8.4. Generalist LLMs, as opposed to specialised or pre-trained models, were chosen to explore the capabilities of these emergent tools in domain-specific and technical problems. In Figure 8.3, the overview of the approach for the experiment to provide data to the LLMs and how the returned answers were evaluated is shown. The approach makes use of the schedule data from which the questions from each category were created, as explained in Section 8.4.1, which are then combined with the prompt text and schedule data, as shown in Section 8.4.2 before being issued to the LLMs. With every answer, from all question categories, returned by the LLMs, five assessments were conducted as outlined in Section 8.4.3, including comparative analysis of any relations.

8.4.1 Query Creation

Three query categories were decided upon, each including variants of a fixed number of unique questions, listed in Table 8.2. To ensure that a suitably broad range of questions was created within each category, while not becoming repetitive, every element determining the position and timescales of a job was considered. All jobs within the schedule data have a start time, processing time (duration), finish time, job number, and machine number. Queries were created to alter at least one of these elements, with the freedom given to the LLM to make further alterations in order to satisfy the request within the question, such as changing the position/start time of a job. Furthermore, each question is designed to facilitate a *yes* or *no* answer, which facilitates the unambiguous record of the correctness of the answer. The categories are defined as:

- Swap - Considers the exchange of the processing times, start times, or identical job numbers across different machines. There are seven unique benchmark questions in this category.

- Increase - Involves any queries on postponing the start of a job, delaying the finish of a job, extending the processing duration of a job on any machine, or extending overall machine run time. There are four unique benchmark questions in this category.
- Decrease - The inverse of increase, entails the reduction of job start, finish, or processing duration, processing time of a job or overall machine run time. It also has four unique benchmark questions.

Table 8.2: Table displaying the unique questions for each of the query categories

Query Category	Queries
Swap	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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	<ol style="list-style-type: none"> 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?

More queries were yielded from the Swap category as the number of unique conditions was greater due to the opportunity to exchange the same job numbers across machines while ensuring all jobs remained scheduled across every machine. Additionally, different job numbers can be position swapped on the same machine, along with

exchanging the start or processing times.

For the Increase and Decrease categories, four unique questions were created for each, as these questions focused on altering the time-based elements, and the moving of jobs within or across machines was not a possible line of query. Within each question, the job and machine numbers are varied and then sent to the LLM for a response.

Every query in each category was repeated a set number of n times, with each repetition containing different job and machine numbers, and time elements for the Increase and Decrease category questions. The purpose of generating variants to the unique questions is to assess the LLM’s consistency in response, while not asking an identical question repeatedly. All numbers for jobs, machines and timings were generated randomly within the constraints of the schedule, ensuring that all numbers were in range for the number of jobs and machines in that given schedule.

The questions were pre-generated and recorded to ensure all questions met the parameters. The recorded questions were then included with the schedule data and prompt for the LLMs to respond to, where both LLMs were sent identical question variants to ensure consistency for comparison of performance.

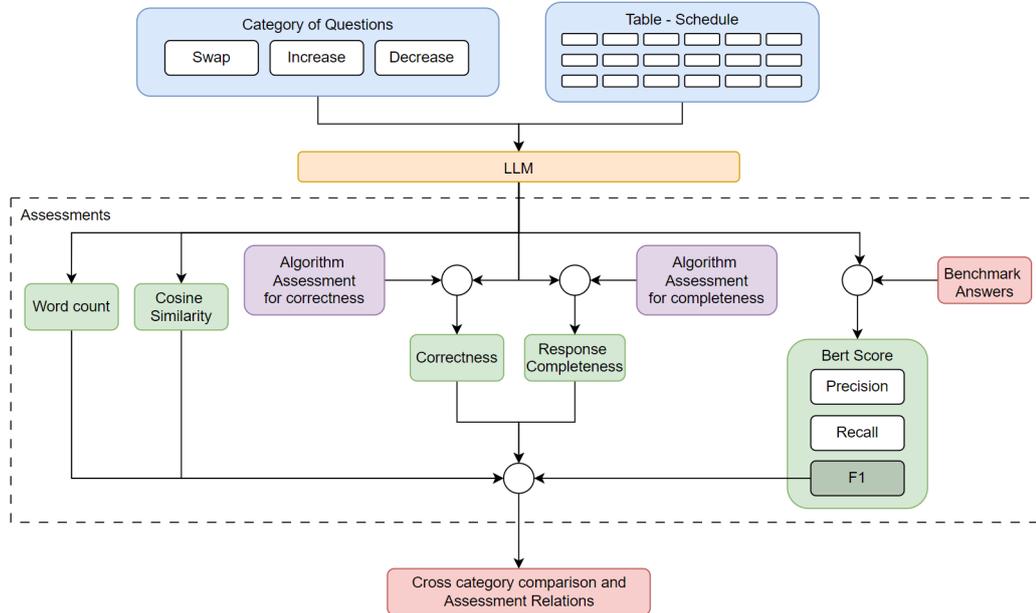


Figure 8.3: Overview of LLM analysis on scheduled data and user queries

8.4.2 Answering benchmark schedule queries

The scheduled data required preparation before being supplied to the LLM with the question prompt; therefore, every schedule had 2 text files created, each containing tabular data. The first file consisted of a table of the processing times for all jobs across each machine, in j rows by m columns. The second file contained a table of the solved schedule start times for each job on every machine, in j rows by m columns. These text files were pre-generated and necessary for providing the scheduled data in tabular format to the LLM and supplemented the prompt to generate the answer responses.

The template prompt, as shown in Table 8.4, was combined with the schedule data files and the questions created in Section 8.4.1 to provide the LLM with all the necessary information to generate an answer to the questions asked. The complete experiment procedure adopted an example-based few-shot approach with the supplied data and information, utilising the API connector to both GPT-4 and LLaMA-3.1. Although the prompt includes a single example question and answer pair from each of the question categories, the approach is recognised as few-shot learning (rather than one-shot learning) as the LLMs do not know the question categories and will therefore apply all the examples when generating an answer. This is an important consideration as it should allow the LLMs more freedom in how to respond and generate more rounded answers, noticeably independent of the directional examples provided.

An additional consideration was to not implement any known prompt optimisation techniques, such as chain-of-thought [362] or ReACT [423], which could be applied to influence performance, as the intention of this novel experiment aimed to establish a baseline of performance with a public, generalist LLMs which could serve as a marker for future research and optimisation techniques.

The results, which included assessments on all question variants in each category, were analysed for their performance in correctness, cosine similarity, completeness, response length, and use of language to evaluate the quality of the LLM responses in answering the questions. Each area of performance measurement was calculated as described in the following Sub-Sections.

8.4.3 Performance measure

For the creation of a baseline performance measure of this novel approach, the performance of the LLM responses must be measured and evaluated. The results from the proposed methodology were assessed for response accuracy, consistency, and interpretability across all question categories and question variants. Furthermore, the potential trends and patterns between the assessment measurements and the question category or schedule size were analysed, detailing the effectiveness of the LLMs in handling varying levels of complexity and questioning tactics.

8.4.3.1 Correctness

Arguably, the most important measure of a successful response generated by the LLMs is whether it is correct or not, and this measure has been widely adopted in research for measuring LLM performance. To accurately capture this, the *yes/no* responses returned by the LLMs were assessed, creating a binary value to identify the correctness of the answer. This was achieved algorithmically to independently validate the feasibility of the questioned alterations, which were then compared against the answer provided by the LLM. If the changes queried in the question were feasible and matched the *yes* response, a score of 1 was given; conversely, if the changes queried were not feasible and the answer given was *no*, a score of 1 was also recorded to validate the response as correct; otherwise, in the event of any other result, a score of 0 was recorded.

An additional consideration was required for questions that necessitate answers with two components for correctness, for example, when exchanging processing times between two jobs (from the Swap question category), which can be achieved by either altering the end time or the start time. On the basis that the experiment algorithm produces a result for two decisions, where at least one coincides with the answer from the LLMs, a score of 1 was recorded to consider the response correct.

Given the direction provided in the prompt, and the deliberate wording of the questions, all answers from the LLMs should return a *yes/no* response; therefore, no other keywords or conditions were assessed to evaluate correctness in the LLM responses.

8.4.3.2 Cosine Similarity

The ability for the LLMs to produce contextually consistent responses is also important, as this demonstrates the ability to adhere to an understanding without simply being repetitive. Within the context of LLM responses, cosine similarity is the measure of how identical two textual statements are.

The cosine similarity was calculated by computing, following the method presented by [424], the similarities between the answers created by the LLMs across each repeated unique question, excluding itself. This means for the n answers generated for each question within each category, and each of these answers (from Table 8.2) was computed for the cosine similarity against the others generated for that question. The returned values were averaged across all responses to the question and recorded.

The cosine similarity scores range from -1 to 1 , where -1 represents no similarity whatsoever, and 1 would be an identical response. It is worth noting that, unlike other metrics, achieving a ‘perfect’ score of 1 in cosine similarity is not desirable, as answers should not be identically worded when the question or details are changed between the question variants. Assessing the Cosine similarity allows for a measure of consistency in language, tone, and response structure, which promotes familiarity with users in real-world applications.

8.4.3.3 Response Completeness

The Completeness of a response is a measure of the LLM’s ability to identify and reference key elements of the question and return these in the response. In the context of this experiment, these key elements include the job number(s) and machine(s) specified, as well as the similarity of the response to the question asked, to evaluate how much common language and terminology was used by the LLMs.

To calculate the completeness of the answer from the LLMs, two steps were followed:

1. Check if the job and machine numbers given in the question were mentioned within the response. Score values were assigned, based on the response returned, ranging from 0 to 1 , where a score of 0 means no mention of the jobs or ma-

chines within the explanatory answer and 1 represents 100% of the noted jobs and machines mentioned.

2. Calculate the cosine similarity between the question and answers generated to assess the use of common words and terminology.

The resultant values from these calculations are averaged to generate the 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 holds no similarity with the original question, and a score of 1 perfectly evidences the relevant information from the question and the greater comprehension the LLM exhibits, while also providing an identically worded response to the question (which should not be possible). As cosine similarity is included within the calculation of completeness, as mentioned above, a ‘perfect’ score of 1 is not expected or desirable, but near 1 scores will demonstrate high performance in comprehension from the LLMs.

8.4.3.4 Word count

Response conciseness is measured to understand the LLM’s response to the constraints in the token limit (as defined in Section 8.5), as well as exploring any relationship between the brevity of response and the scores in other performance measurements.

Assessing for any correlation with other measurements and question categories will provide insight for further investigations and research to explore identified relationships further.

8.4.3.5 BERTScore

BERTScore is a commonly used measurement for evaluating the natural use of language by an LLM, by assessing the precision and recall of a response against reference material. Within this experiment a sample of benchmark questions and answers (including both *yes* and *no* responses), as shown in Table 8.3, was created for each question category by the authors of the study. Each sample was mapped to the list of answers generated by the LLM to return the BERTScore (F1), which measured 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 recorded; however, the results focused on analysing the BERTScore (F1) values. The calculation method follows the instructions provided in [425] and is scored between 0 and 1 , where 0 has no resemblance to the reference material and 1 is identical to a statement in the reference material.

8.4.3.6 Comparative Performance Analysis

Following the completion of all other performance measurements for all questions from all categories and for all schedules, the collated results were analysed for correlation in performance between each question category. To assist with the analysis, graphs were plotted for every performance metric to visualise the results and observe any patterns or trends. The comparative performance was discussed, detailing relevant insights and reasoning gained from the experiment.

8.5 Results

The results are presented in two sections: Chapter 8.5.1 for the individual query categories, namely Swap, Increase and Decrease; and Chapter 8.5.2 for the cross-category comparison across all schedules and categories. The average performance metrics for all queries for each category have been discussed, evaluating where the LLMs performed well and where any difficulties were experienced. The LLM's answers to all fifteen unique questions, created through the process detailed in section 8.4.1, were assessed with five variations of each question for ten schedules, totalling 750 questions. The results from the question variants averaged and presented in combined Tables 8.6 through 8.17, collected and represented throughout Chapter 8.5.1, along with Figures 8.4a through 8.13d throughout Chapter 8.5.2, depicting the averaged performance

Table 8.3: Benchmark answers for each query category

Query Category	YES	NO
Swap	<ol style="list-style-type: none"> 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. 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. 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. 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. 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. 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. 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. 	<ol style="list-style-type: none"> 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. 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. 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. 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. 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. 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. 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	<ol style="list-style-type: none"> 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. 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. 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. 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. 	<ol style="list-style-type: none"> 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. 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. No, it's 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. 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	<ol style="list-style-type: none"> 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. 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. 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. 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. 	<ol style="list-style-type: none"> 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. 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. 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. No, the end time of job 15 on machine 6 cannot be reduced by 14 minutes, as this would breach the scheduling rules defined for the problem.

Table 8.4: Table displaying the prompts used to answer each of the query categories

Prompt used across all categories for answering queries
<p>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:</p> <ol style="list-style-type: none"> 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? 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.

scores for each schedule size. The dataset containing the results is available at the University's Knowledge Research Portal.¹

Section 8.5.1.1 discusses the results from the Swap question category, where all ten benchmark schedule size formats were analysed, namely seven schedules of $15j \times 15m$ (represented as schedules 1 - 7), one schedule $20j \times 15m$, one schedule $20j \times 20m$, and finally, schedule $30j \times 20m$. All the performance metrics were assessed for insights into the LLM's capabilities in individual criteria and collectively across all measurements.

Likewise, Section 8.5.1.2 illustrates the performance of the generated answers and results of the four questions asked within the Increase category for the ten benchmark schedules. After which, Section 8.5.1.3 contains the assessment of average performance for the results for the Decrease category, consisting of all variants of the four unique questions.

Finally, section 8.5.2 contains an overview comparing the performance metrics be-

¹<https://doi.org/10.15129/2cff603d-d504-4d9d-8f54-a66a2a72a835>

Table 8.5: Model configuration to generate answers

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

tween each of the question categories with their respective Figures, where each schedule is represented as 15_15_1 to 15_15_7 for all schedules of 15jx15m, 20_15_1 for schedule 20jx15m, 20_20_1 for schedule 20jx20m, and 30_20_1 for schedule 30jx20m. Additionally, the overall performance is compared with the limited relevant results observed in other studies to determine the performance and success of this experiment.

The analysis conducted within each section provided valuable insight into the application and performance of this novel LLM approach for distinct scheduling problems, which can aid further research to explore and advance capabilities.

Using GPT-4 and LLaMA-3.1 required hyperparameters to determine the performative configuration while completing the answer responses. An iterative approach was taken for the model hyperparameters to ensure both LLMs produced timely and eligible responses. Upon finding the most suitable settings, shown in Table 8.5, this remained unchanged throughout the experiment to ensure fair and consistent assessment across all responses.

8.5.1 Individual Query Categories

8.5.1.1 Swap Query Category

For the Swap query category, Tables 8.6 and 8.8 present 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 8.7, and 8.9 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 Figure 8.4a, where the scores were relatively high across all schedule variants. While LLaMA's

results shown in Figure 8.5a 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 Figures 8.6a and 8.7a, where all $15j \times 15m$ schedules represented with solid lines were compared with $20j \times 15m$, $20j \times 20m$, and $30j \times 20m$ 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 Figure 8.8a for GPT-4, each schedule was plotted against their average scores taken from each question, with schedule 5 of $15j \times 15m$, showing the lowest reading 0.8153 taken from query 7 shown in Table 8.6. Schedule $20j \times 15m$, however, shows the second lowest reading of 0.8531, also taken from query 7 in Table 8.8. The plot for LLaMA, in Figure 8.9a, 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 from both LLMs, there were noticeable variances in the length as shown in Figures 8.10a and 8.11a, 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 low enough to suggest they do not align too closely with the reference material. The consistency of these scores was very similar across both LLMs and is shown in Figures 8.12a and 8.13a, representing a density violin plot of these values.

Chapter 8. Generalisation of Large Language Model Applications to Scheduling

Table 8.6: Average results for the Swap category 15jx15m all schedules for GPT-4

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.7: Average results for the Swap category 15jx15m all schedules for LLaMA-3.1

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.8: Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for GPT-4

20jx15m					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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 8.9: Average results for the Swap category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

20jx15m					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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

8.5.1.2 Increase Query Category

The results for GPT-4 from the Increase query category are shown in Tables 8.10, and 8.12, where the average correctness scores were between 0.4 and 0.75, and over 62% of queries scored an average of 0.6 or higher. The responses from LLaMA shown in Tables 8.11, and 8.13, 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 Figures 8.4b and 8.5b for GPT-4 and LLaMA, respectively. Schedule 4 of 15jx15m stands out for having the lowest correctness score from GPT-4, along with schedule 7 from 15jx15m, 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 Figures 8.6b and 8.7b (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 $20j \times 15m$, $20j \times 20m$, and $30j \times 20m$ 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 $15j \times 15m$ and $30j \times 20m$ 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 Figures 8.8b and 8.9b, emphasising 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 that the length of responses was fairly concise, with neither of the LLMs using the full token limit on average. It can be noted, from GPT-4, that queries 1 and 3 from $15j \times 15m$ schedules 5, 6, and 3, respectively, had significantly lower than average word counts, visible in Figure 8.10b (also shown in Table 8.10). 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 Figure 8.11b.

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 Figure 8.12b. LLaMA scored between 0.63 and 0.665, demonstrating very similar performance, as shown in Figure 8.13b.

Chapter 8. Generalisation of Large Language Model Applications to Scheduling

Table 8.10: Average results for the Increase category 15jx15m all schedules for GPT-4

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.11: Average results for the Increase category 15jx15m all schedules for LLaMA-3.1

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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

8.5.1.3 Decrease Query Category

In the Decrease query category results, Tables 8.14, and 8.16 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 8.15, and 8.17 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 that the larger schedules performed below the average score ranges of the 15jx15m schedules, as shown in Figure 8.4c, 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

Chapter 8. Generalisation of Large Language Model Applications to Scheduling

Table 8.12: Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for GPT-4

20jx15m					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.13: Average results for the Increase category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

20jx15m					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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

addition to assessing larger and more complex datasets; the results fluctuated, and 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 Figure 8.5c, 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 15jx15m 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 Figure 8.6c. 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 Figure 8.7c, 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 Figure 8.8c how closely aligned completeness scores are for schedules 1 and 2 of 15jx15m, and schedule 20jx15m, while the others had much larger score ranges. Conversely, outside of schedule 5 of 15jx15m, the completeness scores for LLaMA were evenly and closely distributed, as shown in

Figure 8.9c, highlighting the ability for the LLM to return relevant information, even if the assessment may be incorrect.

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, emphasising the consistency in responses from this LLM. Considering Figure 8.10c 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 LLaMA in Figure 8.11c 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 Figures 8.12c and 8.13c where there was a close similarity of distribution across all schedules, with the exception of schedules 4 and 6 of $15j \times 15m$ 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.

Chapter 8. Generalisation of Large Language Model Applications to Scheduling

Table 8.14: Average results for the Decrease category 15jx15m all schedules for GPT-4

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.15: Average results for the Decrease category 15jx15m all schedules for LLaMA-3.1

15jx15m schedule 1					
Q#	Correct-ness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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

8.5.2 Cross Category Comparison and Performance

To garner deeper insight from the results, the assessment measures for each query category across both LLM responses were further compared against each other to reveal any correlations or significant differences in performance when considering schedule size or query category. The results were also collated for each schedule variant, in Tables 8.18 and 8.19, to assess if the scheduling size or complexity has any influence on LLM performance.

Chapter 8. Generalisation of Large Language Model Applications to Scheduling

Table 8.16: Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for GPT 4

20jx15m					
Q#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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
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 8.17: Average results for the Decrease category for 20jx15m, 20jx20m and 30jx20m for LLaMA-3.1

20jx15m					
Q#	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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

Table 8.18: Average results across all categories for each schedule (GPT-4)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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 8.19: Average results across all categories for each schedule (LLaMA-3.1)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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

8.5.2.1 Optimal Vs Non-Optimal Schedule Solutions

The overall results were assessed for performance between optimal solution schedules and non-optimal schedules, as mentioned and selected in Section 8.3. Reviewing the overall average performance for the optimal and non-optimal schedules, as shown in

Tables 8.20 and 8.21, revealed there was no significant difference between the sets of schedules and therefore found that the optimality of the schedule solution does not influence the LLMs ability to explain decisions or answer questions. This is useful for real-world applications, as LLMs can assist with system explainability even where optimal scheduling solutions may not be available or possible.

Table 8.20: Average results for Optimal and Non-Optimal Schedules (GPT-4)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
Optimal	0.60	0.8199	0.9473	59.5	0.6674
Non-Optimal	0.53	0.8047	0.9475	60.4	0.6750

Table 8.21: Average results for Optimal and Non-Optimal Schedules (LLaMA-3.1)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
Optimal	0.36	0.8113	0.9543	61.5	0.6310
Non-Optimal	0.34	0.8065	0.9580	60.3	0.6317

8.5.2.2 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 8.22, LLaMA scored 0.17, 0.33, and 0.57 respectively, as shown in Table 8.23. It is worth noting that in Figures 8.4a, 8.4b, and 8.4c, 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 Figures 8.5a, 8.5b, and 8.5c.

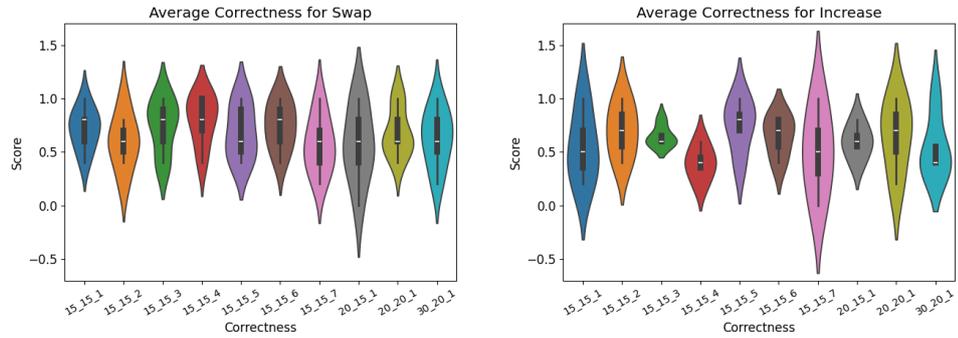
Table 8.22: Average results for all categories across all schedules (GPT-4)

	Correctness	Cosine Similarity	Response Completeness	Word Count	Bert Score fl
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

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

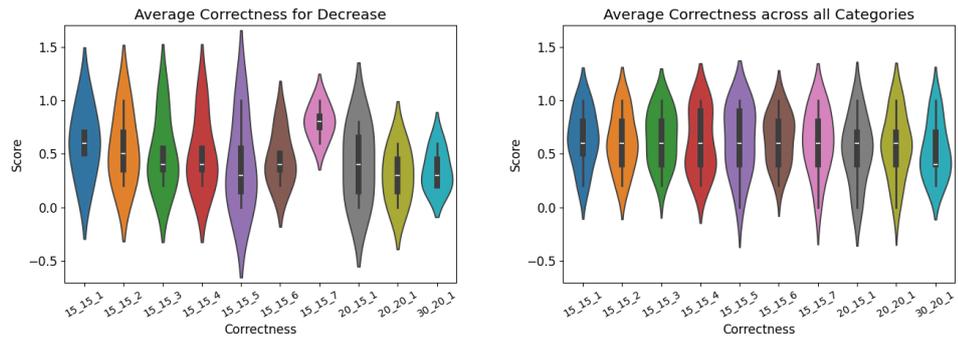
Table 8.23: Average results for all categories across all schedules (LLaMA-3.1)

	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



(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.

Figure 8.4: Average Correctness across all query categories for all schedules for GPT-4

was less consistent and therefore less reliable as responses 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 15jx15m, 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 conditions: firstly, by requiring calculations to

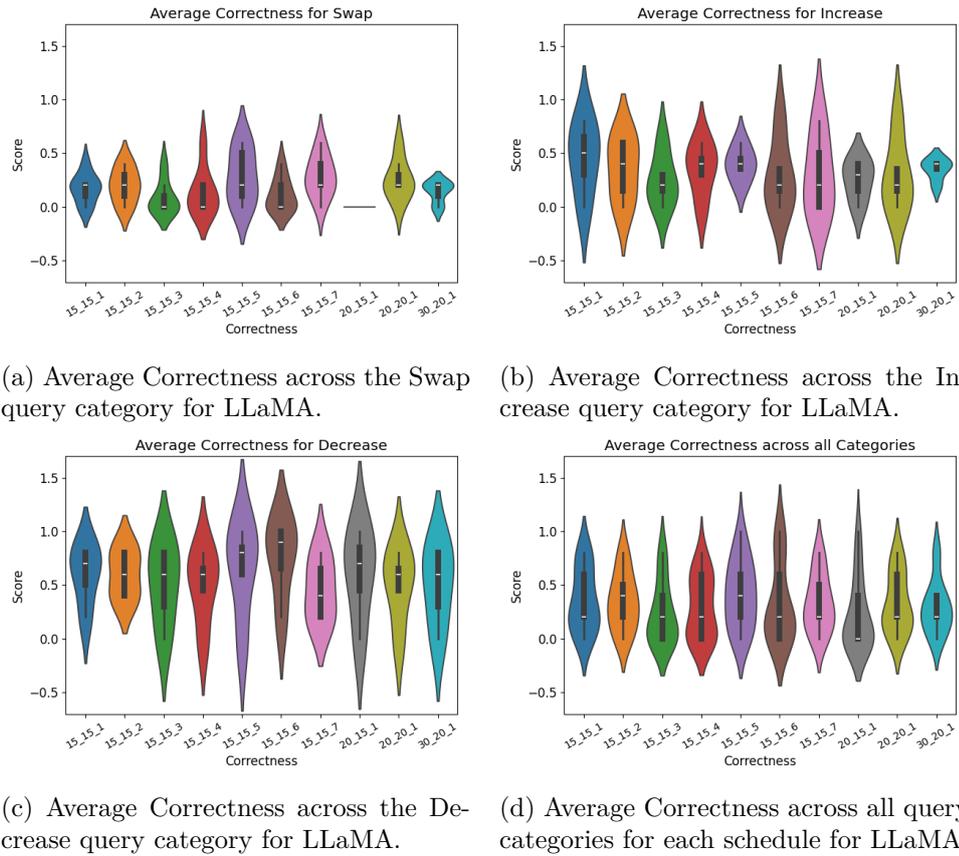


Figure 8.5: Average Correctness across all query categories for all schedules for LLaMA-3.1

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

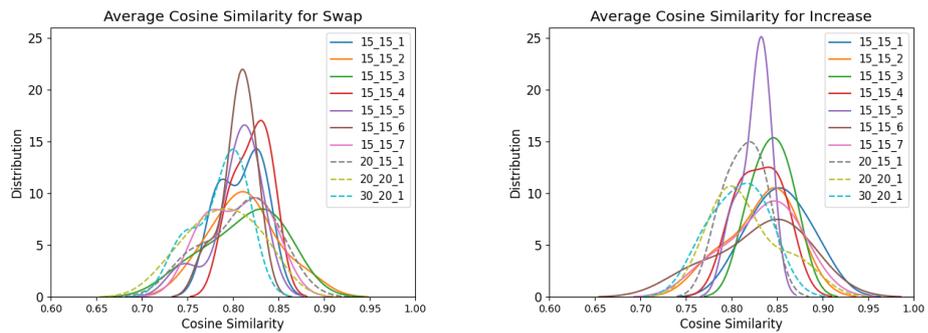
Furthermore, in line with the reduction in overall correctness averages from GPT-4, and increases seen with LLaMA, the increased complexity of the queries also introduced more variability in the correctness scores, which signifies that both LLMs are more inconsistent in comprehending the necessary data 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 Figures 8.4d and 8.5d when considering the collected average scores by each schedule, the density distribution is near-identical across all schedules, with all schedules except *30jx20m* having a median correctness of 0.60 for GPT-4, underlining that the schedule size and complexity have little to no influence on accuracy in assessing scheduling data and queries.

8.5.2.3 Average Cosine Similarity

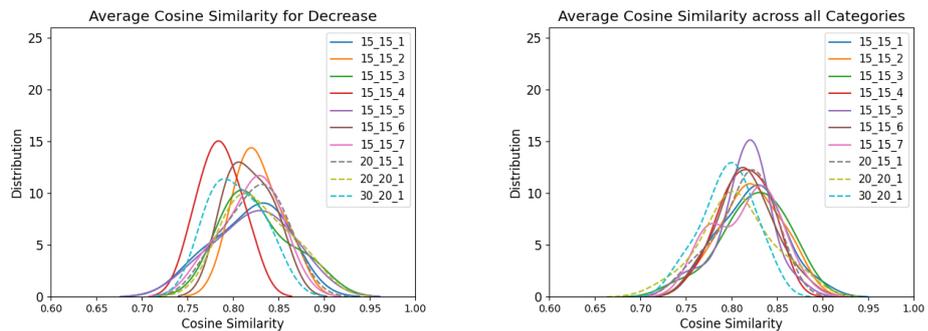
Reviewing the overall average scores of cosine similarity Figures 8.6a, 8.6b, 8.6c, 8.7a, 8.7b, and 8.7c show that the density of scores for each query category was closely aligned for both LLMs, with only one individual stand-out schedule. Schedule 5 of *15jx15m* for GPT-4, within the Increase query 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 Figures 8.6d and 8.7d, demonstrates that both LLMs used very similar language in responses, regardless of the type of query or the size of the dataset.



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

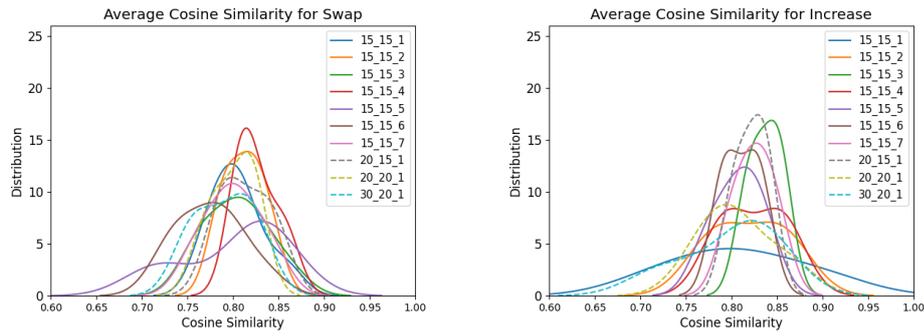
(b) Average Cosine Similarity across the Increase query category for GPT-4.



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

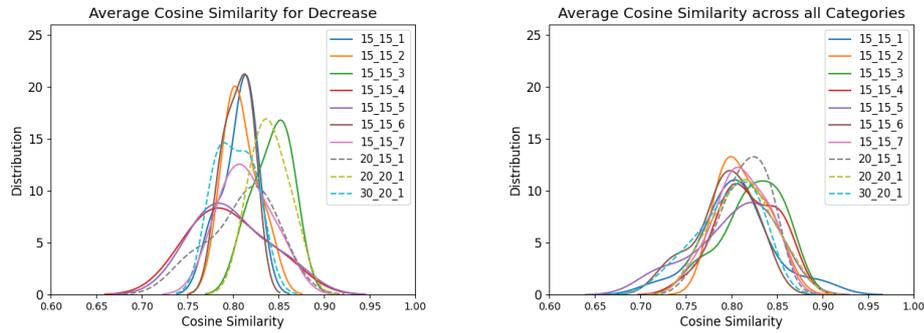
(d) Average Cosine Similarity across all query categories for GPT-4.

Figure 8.6: Average Cosine Similarity across all query categories for GPT-4



(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.

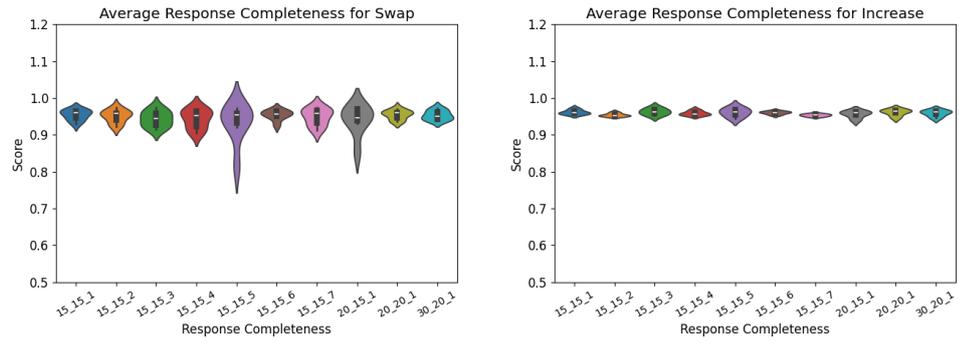
(d) Average Cosine Similarity across all query categories for LLaMA.

Figure 8.7: Average Cosine Similarity across all query categories for LLaMA-3.1

8.5.2.4 Average Response Completeness

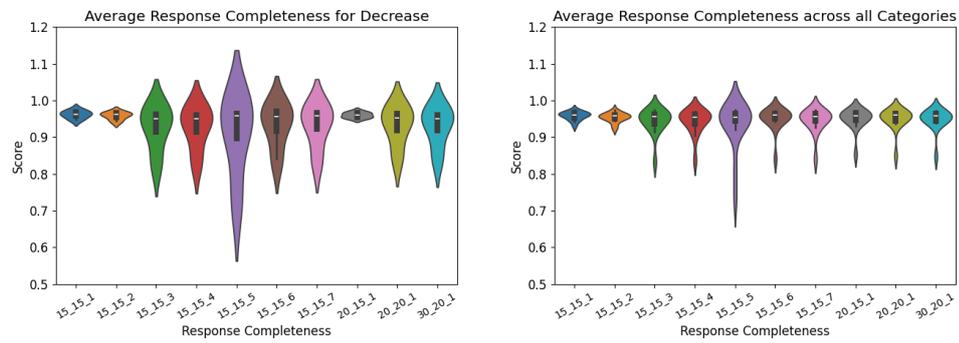
Considering the average scores for completeness, as shown in Figures 8.8a, 8.8b, and 8.8c for GPT-4, the scores achieved in each of the query categories were closely aligned on average, with total averages of 0.9479, 0.9589, and 0.9353 for Swap, Increase, and Decrease respectively, shown in Table 8.22. For LLaMA the total averages were 0.9489, 0.9583, and 0.9580 for Swap, Increase, and Decrease respectively, shown in Table 8.23, with the distribution plotted in Figures 8.9a, 8.9b, and 8.9c.

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 Figure 8.9d. The responses returned in the Decrease query category, for GPT-4, had the most variance and outlying responses, which impacted the overall



(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.

Figure 8.8: Average Response Completeness across all query categories for GPT-4

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 Figure 8.8d for GPT-4, with closely similar 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.

8.5.2.5 Average Word Count

Looking into the average word count scores for GPT-4, depicted in Figures 8.10a, 8.10b, and 8.10c (where the spokes of the plots are the question numbers), the average word count for responses in the Swap query category was 64, with Increase and Decrease both

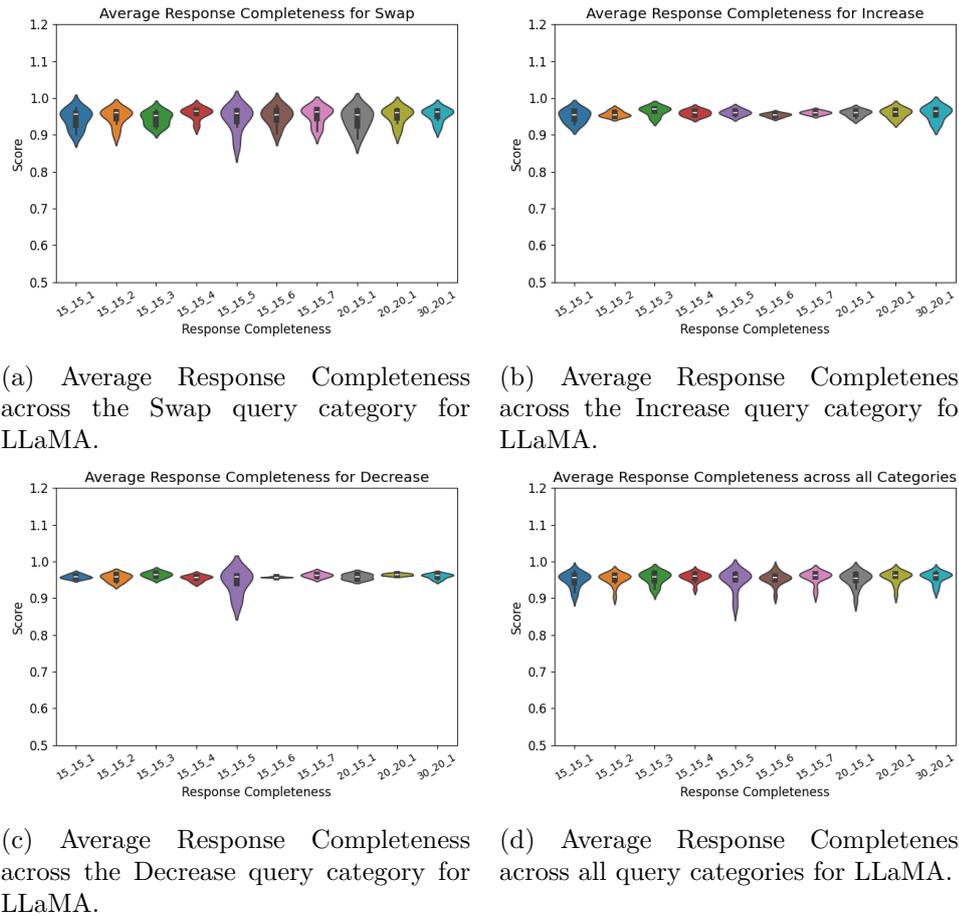


Figure 8.9: Average Response Completeness across all query categories for LLaMA-3.1

returning 58 words on average; rounded up from results in Table 8.22. The responses from LLaMA averaged 60, 59, and 65 for the Swap, Increase, and Decrease query categories, respectively (after rounding), as shown in Table 8.23, with the distributions plotted in Figures 8.11a, 8.11b, and 8.11c.

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 8.18 and 8.19 and Figures 8.10d and 8.11d (where the spokes of the plots are the different schedules), 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

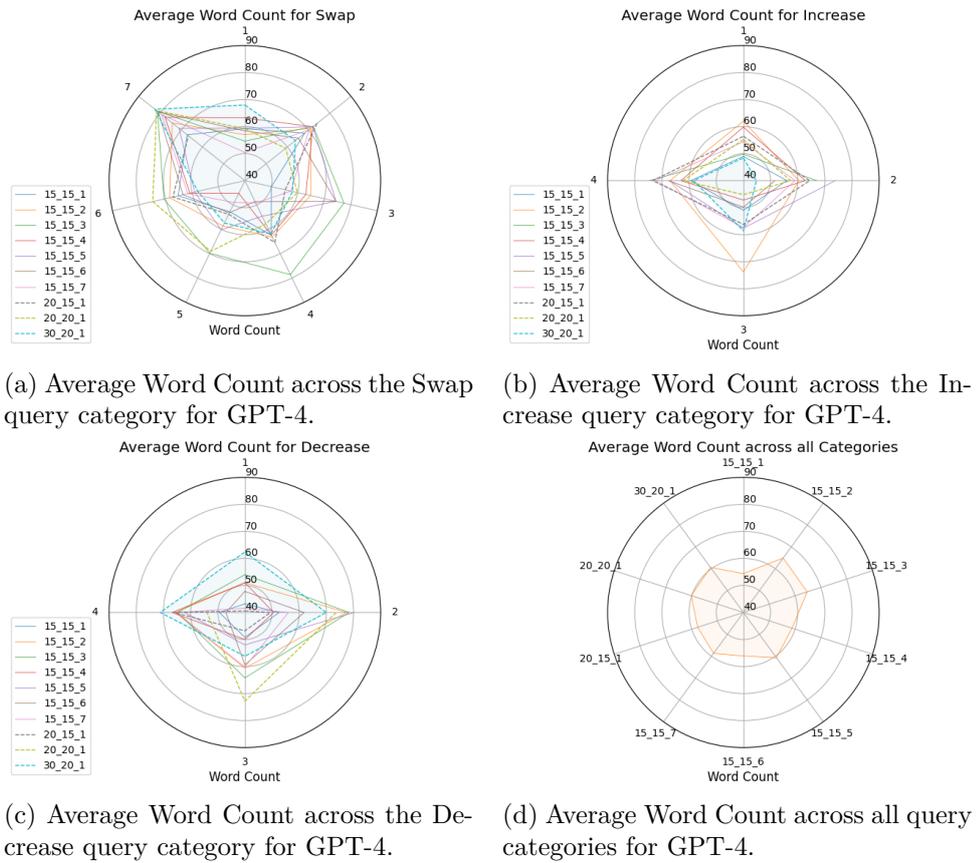


Figure 8.10: Average Word Count across the all query categories for GPT-4

within the schedule.

8.5.2.6 Average BertScore F1

Assessing the average BertScore density plots in Figures 8.12a, 8.12b, 8.12c, 8.13a, 8.13b, and 8.13c, 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 8.22 and 8.23.

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

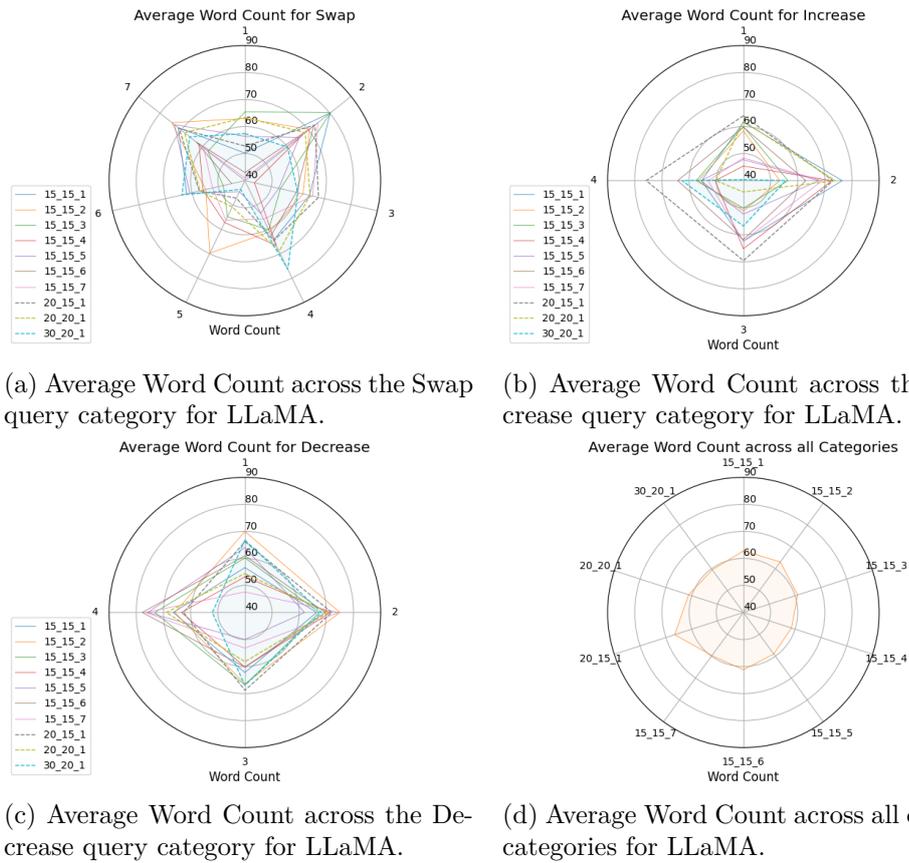


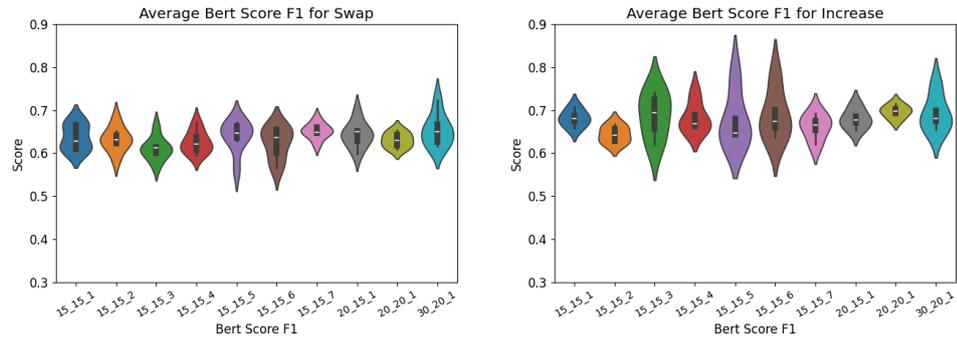
Figure 8.11: Average Word Count across all query categories for LLaMA-3.1

categories, with LLaMA only scoring 5% lower overall. This is also the observation in Figures 8.12d and 8.13d, 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.

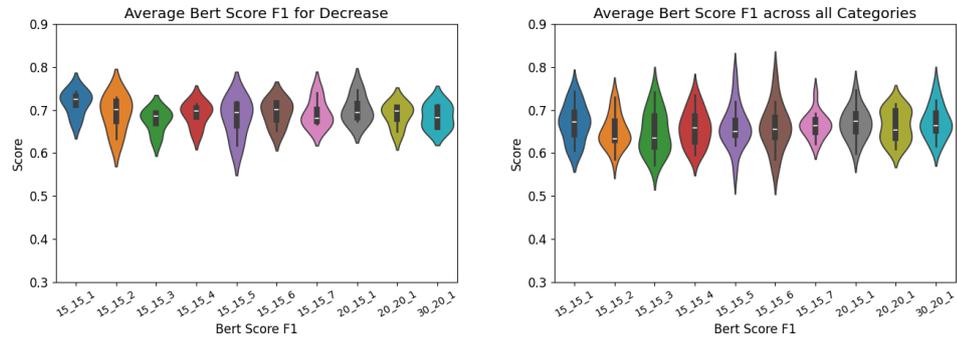
8.5.2.7 Performance Discussion

As mentioned in Section 8.2, 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



(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.

Figure 8.12: Average BertScore F1 across all query categories for GPT-4

offer some insight into how the experimental results of this study 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 study (overall average of 59) [426]. The narrow margin of difference is encouraging as the results generated for this study 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 an-

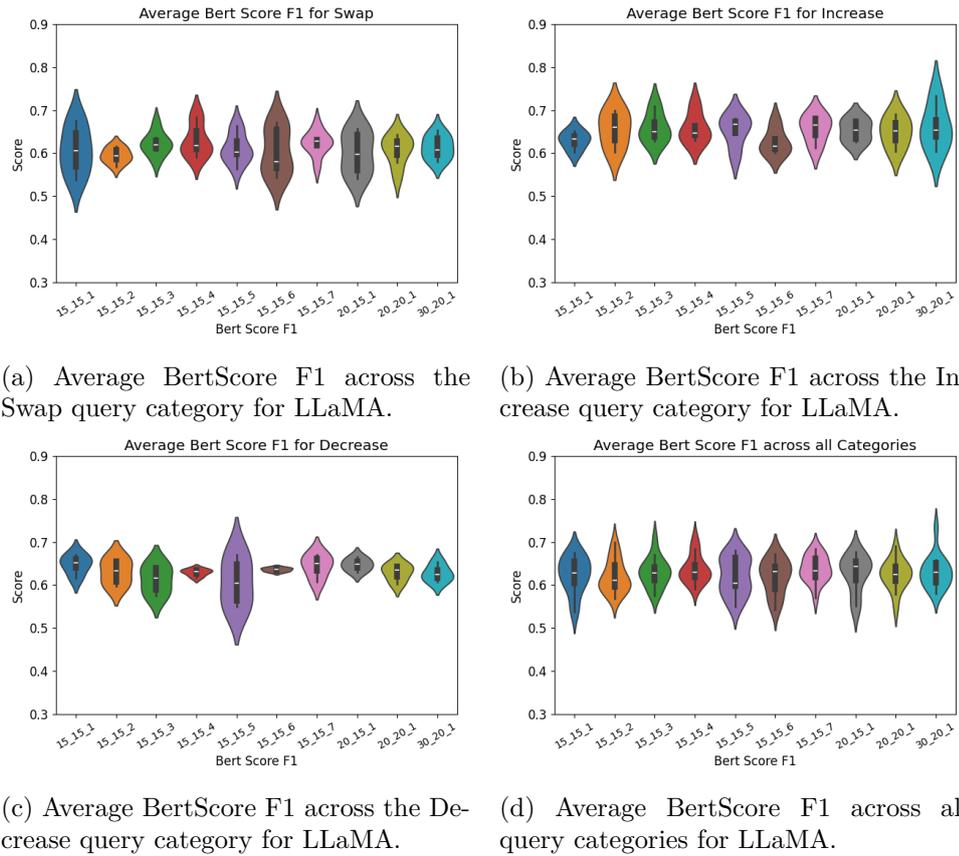


Figure 8.13: Average BertScore F1 across all query categories for LLaMA-3.1

swering 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 experiment (overall average of 0.6689 for GPT-4, and 0.6311 for LLaMA) [427]. The performance demonstrated in this study 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 experiment. 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 suitable for setting a solid performance basis from which more in-depth or targeted 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 [428]), which will facilitate the correction or mitigation of ethical concerns in task assignment and scheduling, avoiding such issues as worker overload or gender-bias.

8.6 Conclusion

This experiment 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. A survey of 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 that 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 experiment.

Chapter 9

Conclusion

9.1 Chapter Overview

This closing chapter summarises the contributions of this thesis and explains the approaches taken in achieving the research aims and objectives. The findings are critiqued, highlighting the limitations and challenges faced throughout the research, where solutions and different approaches may be considered for future work.

9.2 Review of Objectives and Contributions

This research focused on investigating and developing XAI solutions for satellite scheduling through a variation of three different techniques: Computational Argumentation, KGs, and LLMs for onboard and ground segment systems. Many novel and adapted approaches were created during the course of the research to develop solutions and baseline capabilities for XAI utilising emergent technologies. This research's objectives were outlined in Chapter 1.1 and the achievement against each is discussed in this chapter.

Objective 1: Establish a novel formulation of a simplified EO satellite scheduling problem by analysing the constraints of benchmark schedules and satellite scheduling concepts, covering both ground-based and onboard scheduling scenarios.

The Background, in Chapter 2, examined the underlying principles of planning and

scheduling, subsequently identifying several benchmark scheduling problems, which established job shop scheduling as the most prevalent, with one benchmark definition being cited over 3,000 times alone. This benchmark set of job shop schedules consisted of varying sizes of schedule problems, for some of which optimal solutions have been discovered. Job shop schedules are characterised by their management of constraints on available resources and optimisation criteria, for any number of machines or jobs.

Reviewing satellite scheduling specifically revealed a wide number of constraints that can be required for effective satellite schedules, based on the objectives of the mission. Onboard resourcing, visibility windows, memory utilisation, and orbital dynamics were highlighted as some of the common important constraints influencing satellite scheduling. Chapter 3 identified CP and job shop scheduling were found to be effective methods of deriving satellite schedules, which would be utilised for the generation of EO satellite schedules, some of which were designed for onboard generation.

Constraints on land and sunlight visibility, ground station access, and onboard memory utilisation were designed for the schedule to find feasible solutions over a 6-month period, at 5-second action intervals. Four actions were available for selection throughout the schedule, namely ‘Image Taking’, ‘Image Processing’, ‘Image Downlinking’, and ‘Idle’, which was only selected when no other action was possible.

Expanded and more complex schedules were also examined in Chapter 3 to investigate potential real-world use cases of an EO, including the requirement to record a particular area of interest, whether by choice or by fault recovery.

Objective 2: Develop innovative theoretical and computational frameworks that leverage state-of-the-art XAI techniques for evaluating and interpreting benchmark and satellite schedules.

The Background, in Chapter 2, also examined the extensive options for implementing XAI with a mix of model-specific and model-agnostic techniques, which included argumentation-based approaches such as AAF, visualisation techniques such as UMAP, rule-based explanations using fuzzy logic, and natural language explanations through LLMs, to name but a few. The core purposes for XAI were defined as *Justification*, *Control*, and *Improvement* to assist in building confidence and trust in AI models and

systems.

As the main focus of this thesis is to facilitate XAI and trust in automated satellite scheduling, a combination of approaches was selected to explore multiple options for an implementable solution, which is model-agnostic and scalable. AA, KGs, and LLMs were selected for experiments throughout this thesis, often used in a mix of unified approaches, though AA was also used in early singular experiments to develop the approach over time.

AA was implemented, in Chapters 5 and 6, to present information on the selection process within the schedule and provide quantitative analysis of the influencing factors supporting or attacking a decision. KGs were used in Chapters 4 and 7 to visually represent tabular schedule data within the structure of a relational schema, demonstrating the connected relationship of constraints and schedule data. Lastly, LLMs, specifically differing versions of OpenAI’s ChatGPT and Meta’s LLaMA (as investigated in Chapters 6, 7, and 8), were implemented to primarily generate natural language explanations to queries, but were also used for the generation of randomised queries and code for data extraction in building a more automated assessment approach.

Assessment metrics were reviewed with the most appropriate options selected as correctness (also referred to as accuracy), F1 score, cosine similarity, BERTScore, and completeness or validity. Together, these evaluate the model’s performance in returning correct answers to queries, using human-like language (without being repetitive or verbose), and comprehension of the query’s context (utilised in Chapters 4 through 8).

Objective 3: Implement scheduling algorithms to enhance existing satellite schedules, coupled with a novel computational approach to compare and explain the rationale behind scheduling decisions through in-depth conflict analysis.

The selected CP-Solver, OR-Tools, was used to improve the efficiency of the original heuristic schedule, in Chapter 3, which covered a period of 7 days. The solver was able to increase the images taken by 247%, the number of images processed by 673%, the number of downlinked images by 464%, and reduce the amount of idle instances by 94%, significantly improving the balance of activity within the schedule.

This demonstrated how CP can move schedules towards optimal solutions and more effective feasible options.

The CP-Solver was also utilised to perform adaptive replanning in the event of stochastic failure. The solver proved that a simple onboard system would be capable of replanning schedule activities to maintain mission goals without the need for human intervention.

The AA techniques of SEP and PEP were applied to another derived satellite schedule, in Chapter 5, to identify potential improvements through exchanging single or multiple actions, respectively. An objective function based on the mission objective of downlinking processed images from the EO satellite was used as a means of measuring the potential benefit, or loss, from performing the exchange, which is useful in both reactive and proactive schedule adjustments. The results from the SEP and PEP assessments were graphically presented to clearly highlight which exchange of actions was possible and potentially beneficial, and which would put the schedule at risk of resource constraint breach or underperformance. These insights, which would be presented to end-users, can be supplemented with the details on reasoning, such as the memory value in present and future instances, or the overall gain in the objective function score.

Taking the argumentation concept further, in Chapter 6, a TAF was applied to an expanded satellite schedule, including additional constraints and a probability of execution. The schedule was based on capturing specific targeted areas across the time horizon, resulting in 31 unique actions that could attack each other throughout the schedule, leading to over 3500 calculations for determining the weighted value of every action against every other action at each time interval. Although this modified schedule resulted in more idle instances, applying TA produced a quantitative weighting calculation for every action, which could be mapped and presented as a node graph depicting all the attacks and supports that influence the weight for any action at any time instance. The high level of quantitative detail provided by the TA weighting calculations provided significant insight and information into which actions were possible and beneficial to achieving schedule optimality, as well as those which would cause inefficiency or failure. As each attacking and supporting condition was calculated individually,

an end-user would be able to know exactly what influenced a scheduling decision and use this information to validate the reasoning for the existing schedule and make more informed decisions for future schedules.

Objective 4: Create an original dataset of question-and-answer templates tailored to benchmark and the derived satellite schedules, enabling a system to respond to operator natural language queries with contextually relevant and explainable information.

Three datasets of queries were created to be used in different experiments and different schedules. The largest question dataset was created in Chapter 8 and included 750 manually created unique queries used for assessing the answering and comprehension capabilities of two LLMs (GPT-4 and LLaMA-3.1). Three query categories were used, with seven unique queries for the Swap category, and four each for the Increase and Decrease categories. Each query had 5 variations of the job, machine and any temporal conditions included in the query to make unique variations, which were then applied to 10 job shop schedule types to observe the LLMs performance in handling differing schedule data across the query categories. The results showed that GPT-4 had an overall average correctness score of 0.59, while LLaMA achieved 0.35, proving that LLMs do comprehend and respond to the same query and available data differently, which can greatly influence accuracy.

Next, in Chapter 6, a set of 50 queries was written manually by the author to be answered by various LLMs, utilising argumentation explainability on the CP-Solver derived satellite schedule. The LLMs were able to return factual accuracy of 96% in responses to the queries, demonstrating the success and capability of the approach and the LLMs used.

Finally, in Chapter 7, a set of 80 queries was created through an automated prompting configuration with GPT-4 through a few-shot learning approach, which included template examples of queries. This approach was combined with the extraction of data from a KG to provide the LLM with sufficient information to answer the query. The LLM was able to successfully create appropriate queries for four distinct categories (Swap Actions, Single Substitution, Verification, and Forward and Backward) with

minimal corrective action required by the author. In the results, the LLM achieved an average Final Score (FS) of 0.598 without including an explanation example within the query prompt, and 0.636 when including an example explanation, highlighting that generalist LLMs are proficient in generating and answering queries on specialist topics.

Objective 5: Propose a novel interactive methodology focused on explainability for both benchmark and derived satellite schedules to provide interpretable insight for scheduling decisions.

Three approaches for enabling explainability-focused interactivity (simulating explanation generated based on operator inputs) were established, each adapted to interfacing with specific data forms or models. Two LLMs (GPT-4 and LLaMA-3.1), in Chapter 8, were integrated into the querying of benchmark schedules through the use of API connectors. The LLMs were able to answer queries using natural language, with a high degree of quality and a reasonable level of accuracy - though it should be noted that the LLMs were not subject to prompt optimisation or pre-training, which could improve the correctness results (Overall correctness of GPT-4 with 0.59, and LLaMA3.1 with 0.35). Nonetheless, the LLMs demonstrated very consistent capability in interpretability with near-perfect completeness scores and encouraging Bert F1 scores, showing the human-like nature of the responses generated.

A SEP argumentation approach, in Chapter 6, was combined with LLMs (GPT-3, BLOOM, and PaLM) to evaluate a zero-shot and two few-shot approaches for answering queries on the derived satellite schedule. The zero-shot approach did not include any context examples to guide the response from the LLM, whereas the two few-shot approaches included 3 and 10 human-written examples, respectively. In the results, it was found that the few-shot approaches of both example sizes were more accurate than the zero-shot method, which achieved 96% and 90% accuracy, respectively. This affirmed the ability for generalist LLMs to understand and correctly answer queries for specialist and domain-specific topics, using argumentation conditions to supplement the prompting.

An LLM was also combined with KGs, in Chapter 7, as part of a near-complete autonomous structure to extract relevant data and answer queries on the derived satel-

lite schedule. A human operator is able to select queries from the dataset and generate a response through an API connector, with human-in-the-loop controls to ensure data extraction code is valid before proceeding. Two few-shot prompting approaches were implemented, with identical prompt structure except one prompt included a human-written example response from each query category, regardless of the query asked. The LLM generated responses that scored 0.575 and 0.663 in overall average correctness for the prompt without and with example explanation, respectively, showing a marked improvement in accuracy when providing examples. This experiment proved the capability of an LLM to extract data from an external source and combine that with a structured prompt and accurately answer domain-specific queries.

All approaches demonstrated the capability to answer user queries in an accurate and interpretable manner, establishing the baseline for operator interactivity of an automated decision-making or scheduling system.

The work documented in this thesis has met all the objectives of the research and answered the lead research question, through the introduction of multiple approaches to address the concerns and needs of XAI in satellite scheduling systems. This has furthered the research of XAI by focusing on natural, human-like language explanations that are easily interpretable and informative of the deep reasoning within a decision-making model. Furthermore, the research has demonstrated the capabilities of integrating specialist data sources (argumentation frameworks, tabular data, and knowledge graphs) and established that different data sources can be successfully implemented within a single model. While some experiments have exclusively investigated the performance on derived satellite schedules, the methodology has been designed in a way that is model and field-agnostic and can be applied to any field of application of scheduling, enabling further research in a broad range of areas. The results represent a baseline of performance in accuracy, interpretability, and model comprehension that can be utilised in future research to drive improvements to the concepts and implemented designs of XAI solutions.

9.3 Limitations

Generating scheduling systems to be executed onboard in real-world scenarios is more complex than the scheduling systems derived in Chapter 3 of this research. It is noted that this research primarily focused on the application of XAI techniques to explain decisions within schedules and scheduling systems, and not the derivation or optimisation of the schedules themselves. The applicability of XAI in hardware-constrained environments remains a challenge, with some examples of power limitations, communication latency, and computer processing efficiency being complexities that would need to be considered for real-time systems. Ensuring XAI capabilities are sophisticated yet lightweight to execute their procedures, while also being able to monitor the processes, depending on the model implemented, would require additional processing power. Therefore, the size of the model would need to be assessed and tested in advanced simulated conditions before use in real-world applications.

The scheduled data developed in Chapter 3 of this research was also aimed at offline scheduling, using predefined datasets, with the proposal for recovery from stochastic failures during a simulated event, therefore, limiting the ability to assess XAI methods in real-time operational contexts. In real space environments, scheduling must dynamically adapt to new constraints, conditions, and conflicts with other missions. Due to the limited access to real-time data, the XAI techniques proposed in this thesis are designed to scale for more dynamic and complex systems, but this would require coordination with specialist resources to facilitate access to real-world system configuration and designs.

One limitation of this study was also the involvement of real-world scheduling professionals and experts. While the developed XAI methods were designed with human operators in mind, working directly with professionals who interact with scheduling systems on a daily basis was limited, which would have been particularly insightful for the experiments conducted in Chapters 6, 7, and 8. Therefore, allowing for end users, with field specialist knowledge and experience, to test and assist in the development of a functional prototype XAI version should be considered to ensure their needs and

expectations are met.

This research used generalist LLMs, without fine-tuning or pre-training, rather than domain-specific models. This decision, which impacted the direction of Chapters 6, 7, and 8, was made to make the research as accessible to other researchers as possible so the approaches and principles here can be adapted to any environment and model. However, it should be noted that measured performance may have been improved if a specialist or suitably pre-trained model were used, though this would make the results less reproducible for future researchers.

The investigations during this research identified dozens of different XAI techniques and tools, which were not examined for their capabilities to meet the objectives of the research. Other techniques could have been explored, either in addition or as an alternative to the chosen techniques, which would have fundamentally changed the direction of Chapters 4, 5, 6, and 7. The decision to focus on a limited number of techniques was made to more deeply assess some of the more accessible and seemingly capable technologies, especially when adhering to a model-agnostic approach. Multiple generalist LLM models were assessed to compare the performance of different models, but alternative techniques, such as SHAP, LIME, decision rule extraction, or counterfactual explanations, could have been introduced instead; however, responses may not be in human-like language and therefore potentially less interpretable and valuable as explanations.

9.4 Future Research

As part of this research was to provide a benchmark level in approach and performance, there is an extensive opportunity to reproduce and further investigate the field of XAI.

More advanced scheduling techniques could be investigated, than those conducted in Chapter 3, to enhance the efficiency and optimality of offline and flexible schedules. This could include defining adaptive weighted objective functions based on onboard memory status and mission priorities. Furthermore, advanced recovery strategies incorporating ML and Predictive Solving (PS) techniques [429, 430] could be explored, leveraging historical data on weather patterns, component calibrations, and power con-

sumption. These enhancements to the schedule capabilities may also require a revised application of XAI to ensure the reasoning behind reactive decisions is made available to system operators. However, complex and detailed schedules could potentially reduce the quality of explanations and responses, which may be mitigated by pre-training or iterative prompt improvements. The combination of argumentation with ML techniques, creating a hybrid solution, may also improve the system's performance and adaptability, creating a more robust decision support across many applications. Further alternative XAI techniques could also be explored to identify performance gains or losses when compared with the approaches outlined here.

Additionally, introducing prompt optimisation techniques, such as CoT or conversational iteration [271, 431, 432], could be explored to assess the potential improvements in overall accuracy and performance. In some of the experiments, there were identifiable query and response pairs that performed exceptionally, either negatively or positively, when compared to the others in the same query category. This could be captured and introduced as a feedback loop to improve overall performance as part of a human-in-the-loop approach to response enhancement.

Where LLMs are implemented, altering the model hyperparameter settings, as shown in Chapters 6, 7, and 8, could also impact overall performance, while also considering expanding the number of models assessed for greater comparative analysis. Furthermore, in the event of a domain-specific scheduling management trained LLM is developed and made publicly available, the approaches established in this research should be investigated for performance differences and improvements.

Finally, testing the approaches developed in this research with real-world applications and data, integrated with an automated and/or real-time scheduling system, could directly prove the capability of XAI for scheduling decision explainability and greater trust in wider adoption. Systems to target could be robotic control systems, communication satellites or autonomous navigation guidance. These deeper investigations for real-world systems could add scope to develop a functional XUI prototype, which would measure the effectiveness of visual displays and prompts, how to provide XAI for different implementations of scheduling solvers, and the use of NLP for

facilitating interactions between the user and the system.

With the expanded knowledge and data available, KG frameworks can also be expanded to account for larger and more complex scenarios, enabling real-time adaptability for scheduling changes or handling unstructured or missing data. A more in-depth incorporation with predictive analysis could assist in task optimisation through enhanced decision-making processes and highlighting areas of conflicts within a schedule through multi-graph visualisation to enable comparative analysis before committing to execution, which would further aid users in making informed decisions.

Bibliography

- [1] H. Khayyam, B. Javadi, M. Jalili, and R. N. Jazar, *Artificial Intelligence and Internet of Things for Autonomous Vehicles*. Cham: Springer International Publishing, 2020, pp. 39–68. [Online]. Available: <https://doi.org/10.1007/978-3-030-18963-1>
- [2] J. Bajwa, U. Munir, A. Nori, and B. Williams, “Artificial intelligence in healthcare: transforming the practice of medicine,” *Future Healthcare Journal*, vol. 8, no. 2, pp. e188–e194, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2514664524005277>
- [3] M. Tavakoli, A. Faraji, J. Vrolijk, M. Molavi, S. T. Mol, and G. Kismihók, “An ai-based open recommender system for personalized labor market driven education,” *Advanced Engineering Informatics*, vol. 52, p. 101508, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034621002573>
- [4] J. S. Edu, J. M. Such, and G. Suarez-Tangil, “Smart home personal assistants: A security and privacy review,” *ACM Comput. Surv.*, vol. 53, no. 6, dec 2020. [Online]. Available: <https://doi.org/10.1145/3412383>
- [5] W. Ertel, *Introduction to artificial intelligence*. Springer, 2018. [Online]. Available: <https://doi.org/10.1007/978-3-658-43102-0>
- [6] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach, Global Edition*. Pearson Education, 2021. [Online]. Available: <https://books.google.co.uk/books?id=cb0qEAAAQBAJ>

Bibliography

- [7] J. McCarthy, M. L. Minsky, N. Rochester, and C. E. Shannon, “A proposal for the dartmouth summer research project on artificial intelligence, august 31, 1955,” *AI Magazine*, vol. 27, no. 4, p. 12, Dec. 2006. [Online]. Available: <https://ojs.aaai.org/aimagazine/index.php/aimagazine/article/view/1904>
- [8] E. Berkeley, *Giant Brains; Or, Machines that Think*. Wiley, 1949. [Online]. Available: <https://books.google.co.uk/books?id=EDcYAAAAIAAJ>
- [9] A. M. TURING, “I.—COMPUTING MACHINERY AND INTELLIGENCE,” *Mind*, vol. LIX, no. 236, pp. 433–460, 10 1950. [Online]. Available: <https://doi.org/10.1093/mind/LIX.236.433>
- [10] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959. [Online]. Available: <https://doi.org/10.1147/rd.33.0210>
- [11] J. Hernández-Orallo, “Twenty years beyond the turing test: moving beyond the human judges too,” *Minds and Machines*, vol. 30, no. 4, pp. 533–562, 2020. [Online]. Available: <https://doi.org/10.1007/s11023-020-09549-0>
- [12] G. Naidu, T. Zuva, and E. M. Sibanda, “A review of evaluation metrics in machine learning algorithms,” in *Computer Science On-line Conference*. Springer, 2023, pp. 15–25. [Online]. Available: https://doi.org/10.1007/978-3-031-35314-7_2
- [13] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of Research and Development*, vol. 3, no. 3, pp. 210–229, 1959. [Online]. Available: <https://doi.org/10.1147/rd.33.0210>
- [14] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018. [Online]. Available: <https://mitpress.mit.edu/9780262039246/reinforcement-learning/>
- [15] J. Andrews and A. Siddiqi, *Into the Cosmos: Space Exploration and Soviet Culture*, ser. Russian and East European Studies. University of

Bibliography

- Pittsburgh Press, 2011. [Online]. Available: <https://books.google.co.uk/books?id=W7oRuOZbb8IC>
- [16] B. Smith, W. Millar, J. Dunphy, Y.-W. Tung, P. Nayak, E. Gamble, and M. Clark, "Validation and verification of the remote agent for spacecraft autonomy," in *1999 IEEE Aerospace Conference. Proceedings (Cat. No. 99TH8403)*, vol. 1. IEEE, 1999, pp. 449–468. [Online]. Available: <https://doi.org/10.1109/AERO.1999.794352>
- [17] S. Chien, R. Doyle, A. G. Davies, A. Jonsson, and R. Lorenz, "The future of ai in space," *IEEE Intelligent Systems*, vol. 21, no. 4, pp. 64–69, 2006. [Online]. Available: <https://doi.org/10.1109/MIS.2006.79>
- [18] H. Wang, M. Wu, and J. Zhang, "Analysis of human factors in satellite control operation during equipment failure," in *HCI International 2020 – Late Breaking Posters*, C. Stephanidis, M. Antona, and S. Ntoa, Eds. Cham: Springer International Publishing, 2020, pp. 627–632. [Online]. Available: https://doi.org/10.1007/978-3-030-60703-6_80
- [19] B. A. Homssi, K. Dakic, K. Wang, T. Alpcan, B. Allen, S. Kandeepan, A. Al-Hourani, and W. Saad, "Artificial intelligence techniques for next-generation mega satellite networks," *arXiv preprint arXiv:2207.00414*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2207.00414>
- [20] M. Tipaldi, L. Feruglio, P. Denis, and G. D'Angelo, "On applying ai-driven flight data analysis for operational spacecraft model-based diagnostics," *Annual Reviews in Control*, vol. 49, pp. 197–211, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1367578820300213>
- [21] K. Thangavel, D. Spiller, R. Sabatini, S. Amici, N. Longepe, P. Servidia, P. Marzocca, H. Fayek, and L. Ansalone, "Trusted autonomous operations of distributed satellite systems using optical sensors," *Sensors*, vol. 23, no. 6, 2023. [Online]. Available: <https://www.mdpi.com/1424-8220/23/6/3344>

Bibliography

- [22] G. Furano, A. Tavoularis, and M. Rovatti, “Ai in space: Applications examples and challenges,” in *2020 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*. IEEE, 2020, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/DFT50435.2020.9250908>
- [23] K. Thangavel, R. Sabatini, A. Gardi, K. Ranasinghe, S. Hilton, P. Servidia, and D. Spiller, “Artificial intelligence for trusted autonomous satellite operations,” *Progress in Aerospace Sciences*, vol. 144, p. 100960, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0376042123000763>
- [24] V. Kothari, E. Liberis, and N. D. Lane, “The final frontier: Deep learning in space,” in *Proceedings of the 21st international workshop on mobile computing systems and applications*, 2020, pp. 45–49. [Online]. Available: <https://doi.org/10.1145/3376897.3377864>
- [25] B. Li, P. Qi, B. Liu, S. Di, J. Liu, J. Pei, J. Yi, and B. Zhou, “Trustworthy ai: From principles to practices,” *ACM Comput. Surv.*, vol. 55, no. 9, jan 2023. [Online]. Available: <https://doi.org/10.1145/3555803>
- [26] Q. V. Liao and K. R. Varshney, “Human-centered explainable ai (xai): From algorithms to user experiences,” 2022. [Online]. Available: <https://arxiv.org/abs/2110.10790>
- [27] W. Saeed and C. Omlin, “Explainable ai (xai): A systematic meta-survey of current challenges and future opportunities,” *Knowledge-Based Systems*, vol. 263, p. 110273, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950705123000230>
- [28] E. Baccour, N. Mhaisen, A. A. Abdellatif, A. Erbad, A. Mohamed, M. Hamdi, and M. Guizani, “Pervasive ai for iot applications: A survey on resource-efficient distributed artificial intelligence,” *IEEE Communications Surveys & Tutorials*, vol. 24, no. 4, pp. 2366–2418, 2022. [Online]. Available: <https://doi.org/10.1109/comst.2022.3200740>

Bibliography

- [29] J. Ji, T. Qiu, B. Chen, B. Zhang, H. Lou, K. Wang, Y. Duan, Z. He, J. Zhou, Z. Zhang, F. Zeng, K. Y. Ng, J. Dai, X. Pan, A. O’Gara, Y. Lei, H. Xu, B. Tse, J. Fu, S. McAleer, Y. Yang, Y. Wang, S.-C. Zhu, Y. Guo, and W. Gao, “Ai alignment: A comprehensive survey,” 2024. [Online]. Available: <https://arxiv.org/abs/2310.19852>
- [30] H. W. Loh, C. P. Ooi, S. Seoni, P. D. Barua, F. Molinari, and U. R. Acharya, “Application of explainable artificial intelligence for healthcare: A systematic review of the last decade (2011–2022),” *Computer Methods and Programs in Biomedicine*, vol. 226, p. 107161, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0169260722005429>
- [31] P. Weber, K. V. Carl, and O. Hinz, “Applications of explainable artificial intelligence in finance—a systematic review of finance, information systems, and computer science literature,” *Management Review Quarterly*, pp. 1–41, 2023. [Online]. Available: <https://doi.org/10.1007/s11301-023-00320-0>
- [32] H. Mankodiya, D. Jadav, R. Gupta, S. Tanwar, W.-C. Hong, and R. Sharma, “Od-xai: Explainable ai-based semantic object detection for autonomous vehicles,” *Applied Sciences*, vol. 12, no. 11, p. 5310, 2022. [Online]. Available: <https://doi.org/10.3390/app12115310>
- [33] M. Ashoori and J. D. Weisz, “In ai we trust? factors that influence trustworthiness of ai-infused decision-making processes,” *arXiv preprint arXiv:1912.02675*, 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1912.02675>
- [34] M. Kochupillai, M. Kahl, M. Schmitt, H. Taubenböck, and X. X. Zhu, “Earth observation and artificial intelligence: Understanding emerging ethical issues and opportunities,” *IEEE Geoscience and Remote Sensing Magazine*, vol. 10, no. 4, pp. 90–124, 2022. [Online]. Available: <https://doi.org/10.1109/MGRS.2022.3208357>
- [35] J. D. Lee and K. A. See, “Trust in automation: Designing for appropriate reliance,” *Human Factors*, vol. 46, no. 1, pp. 50–80, 2004, PMID: 15151155.

Bibliography

- [Online]. Available: https://journals.sagepub.com/doi/abs/10.1518/hfes.46.1.50_30392
- [36] S. C. Kohn, E. J. De Visser, E. Wiese, Y.-C. Lee, and T. H. Shaw, “Measurement of trust in automation: A narrative review and reference guide,” *Frontiers in psychology*, vol. 12, p. 604977, 2021. [Online]. Available: <https://doi.org/10.3389/fpsyg.2021.604977>
- [37] C. Powell and A. Riccardi, “Towards explainability of on-board satellite scheduling for end user interactions,” Oct. 2021, 72nd International Astronautical Congress, IAC 2021 ; Conference date: 25-10-2021 Through 29-10-2021. [Online]. Available: <https://strathprints.strath.ac.uk/id/eprint/79038>
- [38] C. Powell and A. Riccardi, “On-board re-planning of an earth observation satellite for maximisation of observation campaign goals,” Sep. 2022, 73rd International Astronautical Congress (IAC), IAC2022 ; Conference date: 18-09-2022 Through 22-10-2022. [Online]. Available: <https://www.iafastro.org/events/iac/iac-2022/>,<https://strathprints.strath.ac.uk/id/eprint/82803>
- [39] C. Powell and A. Riccardi, “Abstract argumentation for explainable satellite scheduling,” in *2022 IEEE 9th International Conference on Data Science and Advanced Analytics (DSAA)*, 2022, pp. 1–10. [Online]. Available: <https://doi.org/10.1109/dsaa54385.2022.10032348>
- [40] C. Powell, A. Berquand, and A. Riccardi, “Natural language processing for explainable satellite scheduling,” Apr. 2023, sSPACEOPS 2023 : The 17th International Conference on Space Operations, SPACEOPS 2023 ; Conference date: 06-03-2023 Through 10-03-2023. [Online]. Available: <https://spaceops2023.org/>,<https://strathprints.strath.ac.uk/id/eprint/85129>
- [41] C. Powell and A. Riccardi, “Question answering over knowledge graphs for explainable satellite scheduling,” Oct. 2023, pp. 1–16, international Astronautical Congress ; Conference date: 02-10-2023 Through 06-10-

Bibliography

2023. [Online]. Available: <https://www.iafastro.org/events/iac/iac-2023/>,<https://strathprints.strath.ac.uk/id/eprint/87271>
- [42] C. Powell and A. Riccardi, “Explaining ai decisions in autonomous satellite scheduling via computational argumentation,” Sept 2024, pp. 1–5, SPAICE; Conference date: 17-09-2024 Through 19-09-2024. [Online]. Available: <https://space.esa.int/>,<https://strathprints.strath.ac.uk/id/eprint/91041>
- [43] C. Powell and A. Riccardi, “Question answering over knowledge graphs for explainable satellite scheduling,” *Journal of Aerospace Information Systems*, vol. 22, no. 12, pp. 993–1012, 2025. [Online]. Available: <https://doi.org/10.2514/1.I011531>
- [44] C. Powell and A. Riccardi, “Generating textual explanations for scheduling systems leveraging the reasoning capabilities of large language models,” *Journal of Intelligent Information Systems*, Apr 2025. [Online]. Available: <https://doi.org/10.1007/s10844-025-00940-w>
- [45] R. Barták, “Mixing planning and scheduling to model complex process environments.” in *PACPL*. Manchester, UK, 2000, pp. 329–332. [Online]. Available: <https://kti.mff.cuni.cz/~bartak/clp/downloads/TR2000.pdf>
- [46] J. Kallrath, “Planning and scheduling in the process industry,” *OR spectrum*, vol. 24, pp. 219–250, 2002. [Online]. Available: <https://doi.org/10.1007/s00291-002-0101-7>
- [47] A. De Marco, “Planning and scheduling,” *Project Management for Facility Constructions: A Guide for Engineers and Architects*, pp. 93–122, 2018. [Online]. Available: https://doi.org/10.1007/978-3-319-75432-1_7
- [48] D. E. SMITH, J. FRANK, and A. K. JÓNSSON, “Bridging the gap between planning and scheduling,” *The Knowledge Engineering Review*, vol. 15, no. 1, p. 47–83, 2000. [Online]. Available: <https://doi.org/10.1017/S0269888900001089>

Bibliography

- [49] H. Mintzberg, “What is planning anyway?” *Strategic Management Journal*, vol. 2, no. 3, pp. 319–324, 1981. [Online]. Available: <http://www.jstor.org/stable/2486227>
- [50] N. H. Snyder, “What is planning anyway?: A rejoinder,” *Strategic Management Journal*, vol. 3, no. 3, pp. 265–267, 1982. [Online]. Available: <http://www.jstor.org/stable/2486128>
- [51] H. Kerzner, *Project Management: A Systems Approach to Planning, Scheduling, and Controlling*. John Wiley & Sons, 2007. [Online]. Available: <https://books.google.co.uk/books?id=B1u9e0Dgx80C>
- [52] W. A. Moylan, “Planning and scheduling: The yin and yang of managing a project,” Oct 2002. [Online]. Available: <https://www.pmi.org/learning/library/planning-scheduling-managing-project-8510>
- [53] G. Stecca, *Scheduling*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 1513–1516. [Online]. Available: https://doi.org/10.1007/978-3-662-53120-4_6571
- [54] J. C. Serrano-Ruiz, J. Mula, and R. Poler, “Smart manufacturing scheduling: A literature review,” *Journal of Manufacturing Systems*, vol. 61, pp. 265–287, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0278612521001977>
- [55] M. Geurtsen, J. B. Didden, J. Adan, Z. Atan, and I. Adan, “Production, maintenance and resource scheduling: A review,” *European Journal of Operational Research*, vol. 305, no. 2, pp. 501–529, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221722002673>
- [56] E. Yao, T. Liu, T. Lu, and Y. Yang, “Optimization of electric vehicle scheduling with multiple vehicle types in public transport,” *Sustainable Cities and Society*, vol. 52, p. 101862, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2210670719318098>

Bibliography

- [57] C. Fikar and P. Hirsch, “Home health care routing and scheduling: A review,” *Computers & Operations Research*, vol. 77, pp. 86–95, 2017. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054816301848>
- [58] S. A. Brah and J. L. Hunsucker, *A Study of the Space Shuttle Scheduling Problem*. Boston, MA: Springer US, 1998, pp. 301–315. [Online]. Available: https://doi.org/10.1007/978-1-4757-2876-7_14
- [59] M. Fromherz, “Constraint-based scheduling,” in *Proceedings of the 2001 American Control Conference. (Cat. No.01CH37148)*, vol. 4, 2001, pp. 3231–3244 vol.4. [Online]. Available: <https://doi.org/10.1109/acc.2001.946421>
- [60] R. Barták, “Constraint-based scheduling: An introduction for newcomers,” *IFAC Proceedings Volumes*, vol. 36, no. 3, pp. 75–80, 2003, 7th IFAC Workshop on Intelligent Manufacturing Systems - IMS 2003 [7th IFAC Workshop Preprints], Budapest, Hungary, 6-8 April 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017377388>
- [61] J. Thompson, *Heuristics: An Overview*. Singapore: Springer Nature Singapore, 2023, pp. 1–29. [Online]. Available: https://doi.org/10.1007/978-981-19-8851-6_32-1
- [62] S. Desale, A. Rasool, S. Andhale, and P. Rane, “Heuristic and meta-heuristic algorithms and their relevance to the real world: a survey,” vol. 351, no. 5, 2015, pp. 2349–7084. [Online]. Available: <https://www.ijcert.org/index.php/ijcert/article/view/174>
- [63] X. Luo, Q. Qian, and Y. F. Fu, “Improved genetic algorithm for solving flexible job shop scheduling problem,” *Procedia Computer Science*, vol. 166, pp. 480–485, 2020, proceedings of the 3rd International Conference on Mechatronics and Intelligent Robotics (ICMIR-2019). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050920301836>
- [64] N. Sannomiya, H. Iima, K. Suzuki, and Y. Kobayashi, “Genetic algorithm approach to a scheduling problem for a complex manufacturing system,”

Bibliography

- IFAC Proceedings Volumes*, vol. 31, no. 20, pp. 271–276, 1998, 8th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems: Theory and Applications 1998 (LSS'98), Rio Patras, Greece, 15-17 July 1998. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474667017418064>
- [65] S. Ólafsson, “Chapter 21 metaheuristics,” in *Simulation*, ser. Handbooks in Operations Research and Management Science, S. G. Henderson and B. L. Nelson, Eds. Elsevier, 2006, vol. 13, pp. 633–654. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0927050706130212>
- [66] F. Zhao, H. Bao, L. Wang, T. Xu, N. Zhu, and Jonrinaldi, “A heuristic and meta-heuristic based on problem-specific knowledge for distributed blocking flow-shop scheduling problem with sequence-dependent setup times,” *Engineering Applications of Artificial Intelligence*, vol. 116, p. 105443, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S095219762200433X>
- [67] M. E. Aydin and T. C. Fogarty, “A distributed evolutionary simulated annealing algorithm for combinatorial optimisation problems,” *Journal of heuristics*, vol. 10, pp. 269–292, 2004. [Online]. Available: <https://doi.org/10.1023/B:HEUR.0000026896.44360.f9>
- [68] D. Pham and D. Karaboga, *Intelligent optimisation techniques: genetic algorithms, tabu search, simulated annealing and neural networks*. Springer Science & Business Media, 2012. [Online]. Available: <https://doi.org/10.1007/978-1-4471-0721-7>
- [69] F. Khafa and A. W. Ip, “Optimisation problems and resolution methods in satellite scheduling and space-craft operation: a survey,” *Enterprise Information Systems*, vol. 15, no. 8, pp. 1022–1045, 2021. [Online]. Available: <https://doi.org/10.1080/17517575.2019.1593508>

Bibliography

- [70] P. Brucker, *Classification of Scheduling Problems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 1–10. [Online]. Available: https://doi.org/10.1007/978-3-662-03088-2_1
- [71] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, *Introduction*. Cham: Springer International Publishing, 2019, pp. 1–10. [Online]. Available: https://doi.org/10.1007/978-3-319-99849-7_1
- [72] P. Wojakowski and D. Warżolek, “The classification of scheduling problems under production uncertainty,” *Research in Logistics & Production*, vol. 4, no. 3, p. 245–256, 2014. [Online]. Available: <https://bibliotekanauki.pl/articles/409534>
- [73] A. Muluk, H. Akpolat, and J. Xu, “Scheduling problems—an overview,” *Journal of Systems Science and Systems Engineering*, vol. 12, no. 4, pp. 481–492, 2003. [Online]. Available: <https://doi.org/10.1007/s11518-006-0149-z>
- [74] M. L. Pinedo, *Scheduling: Theory, Algorithms, and Systems*, 6th ed. Springer Publishing Company, Incorporated, 2022. [Online]. Available: <https://doi.org/10.1007/978-3-031-05921-6>
- [75] R. V. Rasmussen and M. A. Trick, “Round robin scheduling – a survey,” *European Journal of Operational Research*, vol. 188, no. 3, pp. 617–636, 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221707005309>
- [76] S. Atin and R. Lubis, “Implementation of critical path method in project planning and scheduling,” *IOP Conference Series: Materials Science and Engineering*, vol. 662, no. 2, p. 022031, nov 2019. [Online]. Available: <https://dx.doi.org/10.1088/1757-899X/662/2/022031>
- [77] D. Müllerklein, P. Fontaine, and F. Ostermeier, “Integrated consideration of assembly line scheduling and feeding: A new model and case study from the automotive industry,” *Computers & Industrial Engineering*, vol. 170, p. 108288, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835222003527>

Bibliography

- [78] T. Aykin, “Optimal shift scheduling with multiple break windows,” *Management science*, vol. 42, no. 4, pp. 591–602, 1996. [Online]. Available: <https://doi.org/10.1287/mnsc.42.4.591>
- [79] A. Allahverdi, C. Ng, T. Cheng, and M. Y. Kovalyov, “A survey of scheduling problems with setup times or costs,” *European Journal of Operational Research*, vol. 187, no. 3, pp. 985–1032, 2008. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221706008174>
- [80] H. Xiong, S. Shi, D. Ren, and J. Hu, “A survey of job shop scheduling problem: The types and models,” *Computers and Operations Research*, vol. 142, p. 105731, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0305054822000338>
- [81] C.-Y. Lee, L. Lei, and M. Pinedo, “Current trends in deterministic scheduling,” *Annals of operations Research*, vol. 70, no. 0, pp. 1–41, 1997. [Online]. Available: <https://doi.org/10.1023/A:1018909801944>
- [82] A. Nagar, J. Haddock, and S. Heragu, “Multiple and bicriteria scheduling: A literature survey,” *European Journal of Operational Research*, vol. 81, no. 1, pp. 88–104, 1995. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0377221793E0140S>
- [83] J. Blazewicz, K. H. Ecker, E. Pesch, G. Schmidt, M. Sterna, and J. Weglarz, *Definition, Analysis and Classification of Scheduling Problems*. Cham: Springer International Publishing, 2019, pp. 61–76. [Online]. Available: https://doi.org/10.1007/978-3-319-99849-7_3
- [84] E. Taillard, “Benchmarks for basic scheduling problems,” *European Journal of Operational Research*, vol. 64, no. 2, pp. 278–285, 1993, project Management and Scheduling. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/037722179390182M>
- [85] L. Lin, M. Gen, Y. Liang, and K. Ohno, “A hybrid ea for reactive flexible job-shop scheduling,” *Procedia Computer Science*, vol. 12, pp.

Bibliography

- 110–115, 2012, complex Adaptive Systems 2012. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050912006308>
- [86] M. K. Amjad, S. I. Butt, R. Kousar, R. Ahmad, M. H. Agha, Z. Faping, N. Anjum, and U. Asgher, “Recent research trends in genetic algorithm based flexible job shop scheduling problems,” *Mathematical Problems in Engineering*, vol. 2018, no. 1, p. 9270802, 2018. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2018/9270802>
- [87] S. S. G. M. Komaki and B. Malakooti, “Flow shop scheduling problems with assembly operations: a review and new trends,” *International Journal of Production Research*, vol. 57, no. 10, pp. 2926–2955, 2019. [Online]. Available: <https://doi.org/10.1080/00207543.2018.1550269>
- [88] R. Graham, E. L. Lawler, J. K. Lenstra, and A. H. G. R. Kan, “Optimization and approximation in deterministic sequencing and scheduling: a survey,” *Annals of discrete mathematics*, vol. 5, pp. 287–326, 1977. [Online]. Available: [https://doi.org/10.1016/S0167-5060\(08\)70356-X](https://doi.org/10.1016/S0167-5060(08)70356-X)
- [89] J. K. Lenstra, “Job shop scheduling,” in *Combinatorial Optimization*, M. Akgül, H. W. Hamacher, and S. Tüfekçi, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1992, pp. 199–207. [Online]. Available: <https://doi.org/10.1007/978-3-642-77489-8>
- [90] S. Dauzère-Pérès, J. Ding, L. Shen, and K. Tamssaouet, “The flexible job shop scheduling problem: A review,” *European Journal of Operational Research*, vol. 314, no. 2, pp. 409–432, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S037722172300382X>
- [91] M. M. Ahmadian, M. Khatami, A. Salehipour, and T. Cheng, “Four decades of research on the open-shop scheduling problem to minimize the makespan,” *European Journal of Operational Research*, vol. 295, no. 2, pp. 399–426, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0377221721002526>

Bibliography

- [92] M. Durasevic and D. Jakobović, “Heuristic and metaheuristic methods for the unrelated machines scheduling problem: A survey,” *ArXiv*, vol. abs/2107.13106, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2107.13106>
- [93] C. Haddow, G. Whitehead, K. Adamson, and B. Sousa, “Mission planning—establishing a common concept for esoc’s missions.” in *SpaceOps 2010 Conference Delivering on the Dream Hosted by NASA Marshall Space Flight Center and Organized by AIAA*, 2010, p. 1969. [Online]. Available: <https://doi.org/10.2514/6.2010-1969>
- [94] P. M. Waswa and S. Redkar, “A survey of space mission architecture and system actualisation methodologies,” *International Journal of Space Science and Engineering*, vol. 4, no. 3, pp. 234–252, 2017. [Online]. Available: <https://doi.org/10.1504/IJSPACESE.2017.085674>
- [95] C. Lenzen, T. Göttfert, F. Mrowka, T. Uhlig, D. Herrmann, and J. Campan, “Mission planning,” in *Spacecraft Operations*. Springer, 2014, pp. 167–211. [Online]. Available: https://doi.org/10.1007/978-3-7091-1803-0_5
- [96] A. Popov, “Mission planning on the international space station program, concepts and systems,” in *IEEE Aerospace Conference, Big Sky, MT, USA*. Citeseer, 2003. [Online]. Available: <https://doi.org/10.1109/AERO.2003.1234186>
- [97] M. Wickler, “A mission planning concept and mission planning system for future manned space missions,” in *NASA. Goddard Space Flight Center, Third International Symposium on Space Mission Operations and Ground Data Systems, Part 1*, 1994. [Online]. Available: <https://ntrs.nasa.gov/citations/19950010820>
- [98] M. Cerf, “Multiple space debris collecting mission: optimal mission planning,” *Journal of Optimization Theory and Applications*, vol. 167, no. 1, pp. 195–218, 2015. [Online]. Available: <https://doi.org/10.1007/s10957-015-0705-0>
- [99] G. Zhang, X. Li, G. Hu, Z. Zhang, J. An, and W. Man, “Mission planning issues of imaging satellites: Summary, discussion, and prospects,” *International*

Bibliography

- Journal of Aerospace Engineering*, vol. 2021, no. 1, p. 7819105, 2021. [Online]. Available: <https://doi.org/10.1155/2021/7819105>
- [100] L. A. Davis and L. Filip, “How long does it take to develop and launch government satellite systems,” *International Cost Estimating and Analysis Association AEROSPACE REPORT*, 2015. [Online]. Available: <https://www.iceaaonline.com/wp-content/uploads/2014/03/Davis-Satellite-ICEAASoCal-090915.pdf>
- [101] B. Horais and J. Guerci, “Real-time mission planning for responsive space lift,” *DigitalCommons@USU*, 2004. [Online]. Available: <https://digitalcommons.usu.edu/smallsat/2004/All2004/13/>
- [102] C. C. for Space Data Systems (CCSDS). (2018) CCSDS mission planning and scheduling report concerning space data system standards. [Online]. Available: <https://public.ccsds.org/Pubs/529x0g1.pdf>
- [103] R. Levinson, S. Nag, and V. Ravindra, “Agile satellite planning for multi-payload observations for earth science,” *arXiv preprint arXiv:2111.07042*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2111.07042>
- [104] R. Tharmarasa, T. Kirubarajan, J. Berger, and M. C. Florea, “Mixed open-and-closed loop satellite task planning,” in *2019 22th International Conference on Information Fusion (FUSION)*, 2019, pp. 1–8. [Online]. Available: <https://doi.org/10.23919/FUSION43075.2019.9011405>
- [105] A. Fukunaga, G. Rabideau, S. Chien, and D. Yan, “Towards an application framework for automated planning and scheduling,” in *1997 IEEE Aerospace Conference*, vol. 1, 1997, pp. 375–386 vol.1. [Online]. Available: <https://doi.org/10.1109/aero.1997.574426>
- [106] S. A. Chien, M. D. Johnston, J. D. Frank, M. E. Giuliano, A. T. Kavelaars, C. Lenzen, and N. Policella, “A generalized timeline representation, services, and interface for automating space mission operations,” *SpaceOps 2012 Conference*, 2012. [Online]. Available: <http://ntrs.nasa.gov/citations/20130009364>

Bibliography

- [107] Y.-j. Song, Z.-y. Zhou, Z.-s. Zhang, F. Yao, and Y.-w. Chen, “A framework involving mec: imaging satellites mission planning,” *Neural Computing and Applications*, vol. 32, pp. 15 329–15 340, 2020. [Online]. Available: <https://doi.org/10.1007/s00521-019-04047-6>
- [108] Y. Chen, G. Tian, J. Guo, and J. Huang, “Task planning for multiple-satellite space-situational-awareness systems,” *Aerospace*, vol. 8, no. 3, p. 73, 2021. [Online]. Available: <https://doi.org/10.3390/aerospace8030073>
- [109] Y. Zhang, W. Wang, P. Zhang, and P. Huang, “Reinforcement-learning-based task planning for self-reconfiguration of cellular satellites,” *IEEE Aerospace and Electronic Systems Magazine*, vol. 37, no. 6, pp. 38–47, 2021. [Online]. Available: <https://doi.org/10.1109/MAES.2021.3089252>
- [110] H. Chen, S. Peng, C. Du, and J. Li, *Earth Observation Satellites book: Task Planning and Scheduling*. Springer Singapore, 2023. [Online]. Available: <https://doi.org/10.1007/978-981-99-3565-9>
- [111] M. Richharia, *Communication Satellites*. London: Macmillan Education UK, 1999, pp. 274–324. [Online]. Available: https://doi.org/10.1007/978-1-349-15131-8_9
- [112] J. H. Jiang, A. J. Zhai, J. Herman, C. Zhai, R. Hu, H. Su, V. Natraj, J. Li, F. Xu, and Y. L. Yung, “Using deep space climate observatory measurements to study the earth as an exoplanet,” *The Astronomical Journal*, vol. 156, no. 1, p. 26, jun 2018. [Online]. Available: <https://dx.doi.org/10.3847/1538-3881/aac6e2>
- [113] S. Li, Q. Yu, and H. Ding, “Reviews and prospects in satellite range scheduling problem,” *Autonomous Intelligent Systems*, vol. 3, no. 9, p. 15, oct 2023. [Online]. Available: <https://doi.org/10.1007/s43684-023-00054-6>
- [114] A. W. Ip, F. Khafa, J. Dong, and M. Gao, “Chapter 6 - an overview of optimization and resolution methods in satellite scheduling and spacecraft operation: description, modeling, and application,” in *IoT and Spacecraft*

Bibliography

- Informatics*, ser. Aerospace Engineering, K. Yung, A. W. Ip, F. Xhafa, and K. Tseng, Eds. Elsevier, 2022, pp. 157–217. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780128210512000027>
- [115] J. C. Pemberton and F. Galiber, “A constraint-based approach to satellite scheduling,” in *Constraint Programming and Large Scale Discrete Optimization*, 1998. [Online]. Available: <https://api.semanticscholar.org/CorpusID:46536625>
- [116] J. Wu, J. Zhang, J. Yang, and L. Xing, “Research on task priority model and algorithm for satellite scheduling problem,” *IEEE Access*, vol. 7, pp. 103 031–103 046, 2019. [Online]. Available: <https://doi.org/10.1109/access.2019.2928992>
- [117] J. N. Pelton, S. Madry, and S. Camacho Lara, *Satellite Applications Handbook: The Complete Guide to Satellite Communications, Remote Sensing, Navigation, and Meteorology*. New York, NY: Springer New York, 2013, pp. 3–17. [Online]. Available: https://doi.org/10.1007/978-1-4419-7671-0_91
- [118] EUMETSAT. (1999) EUMETSAT meteosat second generation. [Online]. Available: <https://www.eumetsat.int/meteosat-second-generation>
- [119] F. J. Batlles, J. Alonso, and G. López, “Cloud cover forecasting from meteosat data,” *Energy Procedia*, vol. 57, pp. 1317–1326, 2014, 2013 ISES Solar World Congress. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1876610214014891>
- [120] M. A. Wulder, D. P. Roy, V. C. Radeloff, T. R. Loveland, M. C. Anderson, D. M. Johnson, S. Healey, Z. Zhu, T. A. Scambos, N. Pahlevan, M. Hansen, N. Gorelick, C. J. Crawford, J. G. Masek, T. Hermosilla, J. C. White, A. S. Belward, C. Schaaf, C. E. Woodcock, J. L. Huntington, L. Lymburner, P. Hostert, F. Gao, A. Lyapustin, J.-F. Pekel, P. Strobl, and B. D. Cook, “Fifty years of landsat science and impacts,” *Remote Sensing of Environment*, vol. 280, p. 113195, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0034425722003054>

Bibliography

- [121] N. E. S. Data and I. Service. NOAA dscovr: Deep space climate observatory. [Online]. Available: <https://www.nesdis.noaa.gov/current-satellite-missions/currently-flying/dscovr-deep-space-climate-observatory>
- [122] T. IWATA, H. ISHIDA, Y. OSAWA, and K. TOMIOKA, “Advanced land observing satellite (alos): Development and on-orbit status,” *The Journal of Space Technology and Science*, vol. 23, no. 1, pp. 1_1–1_13, 2007. [Online]. Available: https://doi.org/10.11230/jsts.23.1_1
- [123] K. Sarabandi, J. Kellndorfer, and L. Pierce, “Gloria: Geostationary/low-earth orbiting radar image acquisition system: a multi-static geo/leo synthetic aperture radar satellite constellation for earth observation,” in *IGARSS 2003. 2003 IEEE International Geoscience and Remote Sensing Symposium. Proceedings (IEEE Cat. No.03CH37477)*, vol. 2, 2003, pp. 773–775 vol.2. [Online]. Available: <https://doi.org/10.1109/igarss.2003.1293913>
- [124] A. J. G. Jr., D. M. Tralli, and S. N. Bajpai, “Medium Earth Orbit (MEO) as an operational observation venue for NOAA’s post GOES-R environmental satellites,” in *Enabling Sensor and Platform Technologies for Spaceborne Remote Sensing*, G. J. Komar, J. Wang, and T. Kimura, Eds., vol. 5659, International Society for Optics and Photonics. SPIE, 2005, pp. 261 – 271. [Online]. Available: <https://doi.org/10.1117/12.577705>
- [125] B. Ashman, F. H. Bauer, J. Parker, and J. Donaldson, “Gps operations in high earth orbit: Recent experiences and future opportunities,” in *2018 SpaceOps Conference*, 2018, p. 2568. [Online]. Available: <https://doi.org/10.2514/6.2018-2568>
- [126] O. Olurope, U. Ukommi, and R. Udoh, “Comparative coverage and horizon plane analysis for leo, meo, geo and heo satellites,” *Journal of Multidisciplinary Engineering Science and Research (JMESR)*, vol. 1, no. 2, pp. 54–56, 2022. [Online]. Available: <http://www.jmesr.co.uk/wp-content/uploads/2022/11/JMESRN42350035..pdf>

Bibliography

- [127] H. Laur and V. Liebig, *Earth Observation Satellite Missions and Data Access*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2020, pp. 1–20. [Online]. Available: https://doi.org/10.1007/978-3-642-27793-1_3-4
- [128] A. Fukunaga, G. Rabideau, S. Chien, and D. Yan, “Aspen: A framework for automated planning and scheduling of spacecraft control and operations,” in *Proc. International Symposium on AI, Robotics and Automation in Space*. Citeseer, 1997, pp. 181–187. [Online]. Available: <https://ntrs.nasa.gov/citations/20060035306>
- [129] H. Chen, S. Peng, C. Du, and J. Li, *Satellite Task Scheduling System*. Singapore: Springer Nature Singapore, 2023, pp. 163–174. [Online]. Available: https://doi.org/10.1007/978-981-99-3565-9_7
- [130] Y. Xiao, S. Zhang, P. Yang, M. You, and J. Huang, “A two-stage flow-shop scheme for the multi-satellite observation and data-downlink scheduling problem considering weather uncertainties,” *Reliability Engineering & System Safety*, vol. 188, pp. 263–275, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0951832018311451>
- [131] J. Liang, Y. he Zhu, Y. zhong Luo, J. cheng Zhang, and H. Zhu, “A precedence-rule-based heuristic for satellite onboard activity planning,” *Acta Astronautica*, vol. 178, pp. 757–772, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576520306159>
- [132] E. Vassev and M. Hinchey, “On the autonomy requirements for space missions,” in *16th IEEE International Symposium on Object/component/service-oriented Real-time distributed Computing (ISORC 2013)*, 2013, pp. 1–10. [Online]. Available: <https://doi.org/10.1109/ISORC.2013.6913242>
- [133] P. Laborie, “Algorithms for propagating resource constraints in ai planning and scheduling: Existing approaches and new results,” *Artificial Intelligence*, vol. 143, no. 2, pp. 151–188, 2003. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370202003624>

Bibliography

- [134] R. Kowalski, *Logic for problem solving*. Department of Computational Logic, Edinburgh University, 1974, vol. 75. [Online]. Available: <https://doi.org/10.1145/1005937.1005947>
- [135] M. Dincbas, P. Van Hentenryck, H. Simonis, A. Aggoun, and A. Herold, “The chip system : Constraint handling in prolog,” in *9th International Conference on Automated Deduction*, E. Lusk and R. Overbeek, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988, pp. 774–775. [Online]. Available: <https://doi.org/10.1007/BFb0012892>
- [136] A. Aggoun and N. Beldiceanu, “Extending chip in order to solve complex scheduling and placement problems,” *Mathematical and computer modelling*, vol. 17, no. 7, pp. 57–73, 1993. [Online]. Available: [https://doi.org/10.1016/0895-7177\(93\)90068-A](https://doi.org/10.1016/0895-7177(93)90068-A)
- [137] P. Tassel, M. Gebser, and K. Schekotihin, “An end-to-end reinforcement learning approach for job-shop scheduling problems based on constraint programming,” *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 33, no. 1, pp. 614–622, Jul. 2023. [Online]. Available: <https://ojs.aaai.org/index.php/ICAPS/article/view/27243>
- [138] P. Laborie, J. Rogerie, P. Shaw, and P. Vilím, “Ibm ilog cp optimizer for scheduling,” *Constraints*, vol. 23, no. 2, pp. 210–250, 2018. [Online]. Available: <https://doi.org/10.1007/s10601-018-9281-x>
- [139] D. Khemani, *Search Methods in Artificial Intelligence*. Cambridge University Press, 2024. [Online]. Available: <https://doi.org/10.1017/9781009284349>
- [140] J. A. Starek, B. Açıkmese, I. A. Nesnas, and M. Pavone, *Spacecraft Autonomy Challenges for Next-Generation Space Missions*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, pp. 1–48. [Online]. Available: https://doi.org/10.1007/978-3-662-47694-9_1
- [141] Z. Zheng, J. Guo, and E. Gill, “Onboard autonomous mission re-planning for multi-satellite system,” *Acta Astronautica*, vol. 145, pp. 28–43,

Bibliography

2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576517312559>
- [142] P. Xie, H. Wang, Y. Chen, and P. Wang, “A heuristic algorithm based on temporal conflict network for agile earth observing satellite scheduling problem,” *IEEE Access*, vol. 7, pp. 61 024–61 033, 2019. [Online]. Available: <https://doi.org/10.1109/access.2019.2902669>
- [143] A. Adadi and M. Berrada, “Peeking inside the black-box: A survey on explainable artificial intelligence (xai),” *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018. [Online]. Available: <https://doi.org/10.1109/ACCESS.2018.2870052>
- [144] F. Doshi-Velez and B. Kim, “Towards a rigorous science of interpretable machine learning,” *arXiv: Machine Learning*, 2017. [Online]. Available: <https://doi.org/10.48550/arXiv.1702.08608>
- [145] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, “Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai,” *Information Fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253519308103>
- [146] V. Vishwarupe, P. M. Joshi, N. Mathias, S. Maheshwari, S. Mhaisalkar, and V. Pawar, “Explainable ai and interpretable machine learning: A case study in perspective,” *Procedia Computer Science*, vol. 204, pp. 869–876, 2022, international Conference on Industry Sciences and Computer Science Innovation. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050922008432>
- [147] V. Hassija, V. Chamola, A. Mahapatra, A. Singal, D. Goel, K. Huang, S. Scardapane, I. Spinelli, M. Mahmud, and A. Hussain, “Interpreting black-box models: A review on explainable artificial intelligence,” *Cognitive*

Bibliography

- Computation*, vol. 16, no. 1, 2024. [Online]. Available: <https://doi.org/10.1007/s12559-023-10179-8>
- [148] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, “Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence,” *Information Fusion*, vol. 99, p. 101805, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1566253523001148>
- [149] J. Allgaier, L. Mulansky, R. L. Draelos, and R. Pryss, “How does the model make predictions? a systematic literature review on the explainability power of machine learning in healthcare,” *Artificial Intelligence in Medicine*, vol. 143, p. 102616, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0933365723001306>
- [150] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, “Explainable ai: A review of machine learning interpretability methods,” *Entropy (Basel)*, 2020. [Online]. Available: <https://doi.org/10.3390/e23010018>
- [151] H. P. Grice, “Logic and conversation,” *Syntax and semantics*, vol. 3, pp. 43–58, 1975. [Online]. Available: https://doi.org/10.1163/9789004368811_003
- [152] R. Pugliese, S. Regondi, and R. Marini, “Machine learning-based approach: global trends, research directions, and regulatory standpoints,” *Data Science and Management*, vol. 4, pp. 19–29, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666764921000485>
- [153] S. V. Mahadevkar, B. Khemani, S. Patil, K. Kotecha, D. R. Vora, A. Abraham, and L. A. Gabralla, “A review on machine learning styles in computer vision—techniques and future directions,” *IEEE Access*, vol. 10, pp. 107 293–107 329, 2022. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3209825>
- [154] S. A. and S. R., “A systematic review of explainable artificial intelligence models and applications: Recent developments and future trends,” *Decision*

Bibliography

- Analytics Journal*, vol. 7, p. 100230, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S277266222300070X>
- [155] P. Cunningham, M. Cord, and S. J. Delany, *Supervised Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 21–49. [Online]. Available: https://doi.org/10.1007/978-3-540-75171-7_2
- [156] Z. Ghahramani, *Unsupervised Learning*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 72–112. [Online]. Available: https://doi.org/10.1007/978-3-540-28650-9_5
- [157] X. Zhu and A. B. Goldberg, *Overview of Semi-Supervised Learning*. Cham: Springer International Publishing, 2009, pp. 9–19. [Online]. Available: https://doi.org/10.1007/978-3-031-01548-9_2
- [158] L. P. Kaelbling, M. L. Littman, and A. W. Moore, “An introduction to reinforcement learning,” in *The Biology and Technology of Intelligent Autonomous Agents*, L. Steels, Ed. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 90–127. [Online]. Available: https://doi.org/10.1007/978-3-642-79629-6_5
- [159] J. Zerilli, U. Bhatt, and A. Weller, “How transparency modulates trust in artificial intelligence,” *Patterns*, vol. 3, no. 4, p. 100455, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666389922000289>
- [160] A. Jacovi, A. Marasović, T. Miller, and Y. Goldberg, “Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai,” in *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, ser. FAccT ’21. New York, NY, USA: Association for Computing Machinery, 2021, p. 624–635. [Online]. Available: <https://doi.org/10.1145/3442188.3445923>
- [161] S. Larsson and F. Heintz, “Transparency in artificial intelligence,” *Internet policy review*, vol. 9, no. 2, 2020. [Online]. Available: <http://dx.doi.org/10.14763/2020.2.1469>

Bibliography

- [162] A. Hanif, A. Beheshti, B. Benatallah, X. Zhang, Habiba, E. Foo, N. Shabani, and M. Shahabikargar, “A comprehensive survey of explainable artificial intelligence (xai) methods: Exploring transparency and interpretability,” in *International Conference on Web Information Systems Engineering*. Springer, 2023, pp. 915–925. [Online]. Available: https://doi.org/10.1007/978-981-99-7254-8_71
- [163] U. Czerwinska, *Interpretability of Machine Learning Models*. Cham: Springer International Publishing, 2022, pp. 275–303. [Online]. Available: https://doi.org/10.1007/978-3-030-88389-8_14
- [164] A. Vellido, “The importance of interpretability and visualization in machine learning for applications in medicine and health care,” *Neural computing and applications*, vol. 32, no. 24, pp. 18 069–18 083, 2020. [Online]. Available: <https://doi.org/10.1007/s00521-019-04051-w>
- [165] A. Corso and M. J. Kochenderfer, “Interpretable safety validation for autonomous vehicles,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2020, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/ITSC45102.2020.9294490>
- [166] D. Brigo, X. Huang, A. Pallavicini, and H. S. d. O. Borde, “Interpretability in deep learning for finance: a case study for the heston model,” *arXiv preprint arXiv:2104.09476*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2104.09476>
- [167] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nature Machine Intelligence*, vol. 1, pp. 206 – 215, 2018. [Online]. Available: <https://doi.org/10.1038/s42256-019-0048-x>
- [168] S. Chakraborty, R. Tomsett, R. Raghavendra, D. Harborne, M. Alzantot, F. Cerutti, M. Srivastava, A. Preece, S. Julier, R. M. Rao, T. D. Kelley, D. Braines, M. Sensoy, C. J. Willis, and P. Gurrarn, “Interpretability of deep learning models: A survey of results,” in *2017*

Bibliography

- IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (Smart-World/SCALCOM/UIC/ATC/CBDCOM/IOP/SCI)*, 2017, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/uic-atc.2017.8397411>
- [169] A. A. Ismail, S. Feizi, and H. C. Bravo, “Improving deep learning interpretability by saliency guided training,” in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS ’21. Red Hook, NY, USA: Curran Associates Inc., 2024. [Online]. Available: https://ui.adsabs.harvard.edu/link_gateway/2021arXiv211114338A/doi:10.48550/arXiv.2111.14338
- [170] C. Combi, B. Amico, R. Bellazzi, A. Holzinger, J. H. Moore, M. Zitnik, and J. H. Holmes, “A manifesto on explainability for artificial intelligence in medicine,” *Artificial Intelligence in Medicine*, vol. 133, p. 102423, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0933365722001750>
- [171] P. Banerjee and R. P. Barnwal, *Methods and Metrics for Explaining Artificial Intelligence Models: A Review*. Cham: Springer International Publishing, 2023, pp. 61–88. [Online]. Available: https://doi.org/10.1007/978-3-031-12807-3_4
- [172] A. Jacovi, A. Marasović, T. Miller, and Y. Goldberg, “Formalizing trust in artificial intelligence: Prerequisites, causes and goals of human trust in ai,” in *Proceedings of the 2021 ACM conference on fairness, accountability, and transparency*, 2021, pp. 624–635. [Online]. Available: <https://doi.org/10.1145/3442188.3445923>
- [173] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” *Artificial Intelligence*, vol. 267, pp. 1–38, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370218305988>
- [174] Z. C. Lipton, “The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery.”

Bibliography

- Queue*, vol. 16, no. 3, p. 31–57, Jun. 2018. [Online]. Available: <https://doi.org/10.1145/3236386.3241340>
- [175] Q. Li, R. Cummings, and Y. Mintz, “Optimal local explainer aggregation for interpretable prediction,” *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, pp. 12 000–12 007, Jun. 2022. [Online]. Available: <https://ojs.aaai.org/index.php/AAAI/article/view/21458>
- [176] R. Machlev, L. Heistrene, M. Perl, K. Levy, J. Belikov, S. Mannor, and Y. Levron, “Explainable artificial intelligence (xai) techniques for energy and power systems: Review, challenges and opportunities,” *Energy and AI*, vol. 9, p. 100169, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2666546822000246>
- [177] M. T. Ribeiro, S. Singh, and C. Guestrin, “”why should i trust you?”: Explaining the predictions of any classifier,” in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’16. New York, NY, USA: Association for Computing Machinery, 2016, p. 1135–1144. [Online]. Available: <https://doi.org/10.1145/2939672.2939778>
- [178] G. Visani, E. Bagli, and F. Chesani, “Optilime: Optimized lime explanations for diagnostic computer algorithms,” *arXiv preprint arXiv:2006.05714*, 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2006.05714>
- [179] S. M. Lundberg and S.-I. Lee, “A unified approach to interpreting model predictions,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf
- [180] X. Dastile and T. Celik, “Making deep learning-based predictions for credit scoring explainable,” *IEEE Access*, vol. 9, pp. 50 426–50 440, 2021. [Online]. Available: <https://doi.org/10.1109/ACCESS.2021.3068854>

Bibliography

- [181] S. Wachter, B. Mittelstadt, and C. Russell, “Counterfactual explanations without opening the black box: Automated decisions and the gdpr,” *Harv. JL & Tech.*, vol. 31, p. 841, 2017. [Online]. Available: https://heinonline.org/hol-cgi-bin/get_pdf.cgi?handle=hein.journals/hjlt31§ion=29
- [182] M.-S. Giannopoulou, “An application of latentcf++ on providing counterfactual explanations for fraud detection,” 2023. [Online]. Available: <https://urn.kb.se/resolve?urn=urn:nbn:se:su:diva-219606>
- [183] J. H. Friedman, “Greedy function approximation: a gradient boosting machine,” *Annals of statistics*, pp. 1189–1232, 2001. [Online]. Available: <https://www.jstor.org/stable/2699986>
- [184] R.-T. Mora-Garcia, M.-F. Cespedes-Lopez, and V. R. Perez-Sanchez, “Housing price prediction using machine learning algorithms in covid-19 times,” *Land*, vol. 11, no. 11, p. 2100, 2022. [Online]. Available: <https://doi.org/10.3390/land11112100>
- [185] C. Okoli, “Statistical inference using machine learning and classical techniques based on accumulated local effects (ale),” *arXiv preprint arXiv:2310.09877*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2310.09877>
- [186] J. A. Bastos and M. I. Bernardes, “Understanding online purchases with explainable machine learning,” *Information*, vol. 15, no. 10, p. 587, 2024. [Online]. Available: <https://doi.org/10.3390/info15100587>
- [187] A. Goldstein, A. Kapelner, J. Bleich, and E. Pitkin, “Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation,” *journal of Computational and Graphical Statistics*, vol. 24, no. 1, pp. 44–65, 2015. [Online]. Available: <https://doi.org/10.1080/10618600.2014.907095>
- [188] G. Casalicchio, C. Molnar, and B. Bischl, “Visualizing the feature importance for black box models,” in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2018, Dublin, Ireland, September 10–14,*

Bibliography

- 2018, *Proceedings, Part I 18*. Springer, 2019, pp. 655–670. [Online]. Available: https://doi.org/10.1007/978-3-030-10925-7_40
- [189] M. R. Smith, E. Acquesta, A. Ames, A. N. Carey, C. R. Cuellar, R. V. Field, T. Maxfield, S. Mitchell, B. Moss, E. Morris *et al.*, “Sage intrusion detection system: Sensitivity analysis guided explainability for machine learning.” Sandia National Lab.(SNL-NM), Albuquerque, NM (United States), Tech. Rep., 2021. [Online]. Available: <https://doi.org/10.2172/1820253>
- [190] B. Van Stein, E. Raponi, Z. Sadeghi, N. Bouman, R. C. Van Ham, and T. Bäck, “A comparison of global sensitivity analysis methods for explainable ai with an application in genomic prediction,” *IEEE Access*, vol. 10, pp. 103 364–103 381, 2022. [Online]. Available: <https://doi.org/10.1109/ACCESS.2022.3210175>
- [191] M. T. Ribeiro, S. Singh, and C. Guestrin, “Anchors: High-precision model-agnostic explanations,” in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018. [Online]. Available: <https://doi.org/10.1609/aaai.v32i1.11491>
- [192] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *CoRR*, vol. abs/1312.6034, 2013. [Online]. Available: <https://doi.org/10.48550/arXiv.1312.6034>
- [193] F. C. Morabito, M. Campolo, C. Ieracitano, and N. Mammone, “12 - explainable deep learning to information extraction in diagnostics and electrophysiological multivariate time series ,” in *Artificial Intelligence in the Age of Neural Networks and Brain Computing (Second Edition)*, second edition ed., R. Kozma, C. Alippi, Y. Choe, and F. C. Morabito, Eds. Academic Press, 2024, pp. 225–250. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/B9780323961042000117>
- [194] G. Montavon, A. Binder, S. Lapuschkin, W. Samek, and K.-R. Müller, *Layer-Wise Relevance Propagation: An Overview*. Cham: Springer International

Bibliography

- Publishing, 2019, pp. 193–209. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6_10
- [195] C. J. Anders, G. Montavon, W. Samek, and K.-R. Müller, *Understanding Patch-Based Learning of Video Data by Explaining Predictions*. Cham: Springer International Publishing, 2019, pp. 297–309. [Online]. Available: https://doi.org/10.1007/978-3-030-28954-6_16
- [196] L. Arras, F. Horn, G. Montavon, K.-R. Müller, and W. Samek, “Explaining predictions of non-linear classifiers in NLP,” in *Proceedings of the 1st Workshop on Representation Learning for NLP*, P. Blunsom, K. Cho, S. Cohen, E. Grefenstette, K. M. Hermann, L. Rimell, J. Weston, and S. W.-t. Yih, Eds. Berlin, Germany: Association for Computational Linguistics, Aug. 2016, pp. 1–7. [Online]. Available: <https://doi.org/10.18653/v1/w16-1601>
- [197] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” 2019. [Online]. Available: <https://arxiv.org/abs/1704.02685>
- [198] N. Y. Murad, M. H. Hasan, M. H. Azam, N. Yousuf, and J. S. Yalli, “Unraveling the black box: A review of explainable deep learning healthcare techniques,” *IEEE Access*, 2024. [Online]. Available: <https://doi.org/10.1109/access.2024.3398203>
- [199] Y. Izza, A. Ignatiev, and J. Marques-Silva, “On explaining decision trees,” *arXiv preprint arXiv:2010.11034*, 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2010.11034>
- [200] N. A. AHMED and A. Alpköçak, “A quantitative evaluation of explainable ai methods using the depth of decision tree,” *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 30, no. 6, pp. 2054–2072, 2022. [Online]. Available: <https://doi.org/10.55730/1300-0632.3924>
- [201] L. Breiman, “Random forests,” *Machine learning*, vol. 45, pp. 5–32, 2001. [Online]. Available: <https://doi.org/10.1023/A:1010933404324>

Bibliography

- [202] M. Saarela and S. Jauhiainen, “Comparison of feature importance measures as explanations for classification models,” *SN Applied Sciences*, vol. 3, no. 2, p. 272, 2021. [Online]. Available: <https://doi.org/10.1007/s42452-021-04148-9>
- [203] S. Wali and I. Khan, “Explainable ai and random forest based reliable intrusion detection system,” *Authorea Preprints*, 2021. [Online]. Available: <https://doi.org/10.36227/techrxiv.17169080.v1>
- [204] S. Kabir, M. S. Hossain, and K. Andersson, “An advanced explainable belief rule-based framework to predict the energy consumption of buildings,” *Energies*, vol. 17, no. 8, p. 1797, 2024. [Online]. Available: <https://doi.org/10.3390/en17081797>
- [205] G. Vilone, L. Rizzo, and L. Longo, “A comparative analysis of rule-based, model-agnostic methods for explainable artificial intelligence,” in *28th AIAI Irish Conference on Artificial Intelligence and Cognitive Science*. Technological University Dublin, 2020. [Online]. Available: <https://doi.org/10.21427/z4x3-3f86>
- [206] K. Yang and J. Deng, “Learning symbolic rules for reasoning in quasi-natural language,” *arXiv preprint arXiv:2111.12038*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2111.12038>
- [207] X. Zhu, D. Wang, W. Pedrycz, and Z. Li, “Fuzzy rule-based local surrogate models for black-box model explanation,” *IEEE Transactions on Fuzzy Systems*, vol. 31, no. 6, pp. 2056–2064, 2023. [Online]. Available: <https://doi.org/10.1109/TFUZZ.2022.3218426>
- [208] D. Macha, M. Kozielski, Łukasz Wróbel, and M. Sikora, “Rulexai—a package for rule-based explanations of machine learning model,” *SoftwareX*, vol. 20, p. 101209, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2352711022001273>
- [209] A. Rozinat and W. M. P. van der Aalst, *Decision Mining in ProM*, S. Dustdar, J. L. Fiadeiro, and A. P. Sheth, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006. [Online]. Available: https://doi.org/10.1007/11841760_33

Bibliography

- [210] R. Bemthuis, W. Wang, M.-E. Iacob, and P. Havinga, “Business rule extraction using decision tree machine learning techniques: A case study into smart returnable transport items,” *Procedia Computer Science*, vol. 220, pp. 446–455, 2023, the 14th International Conference on Ambient Systems, Networks and Technologies Networks (ANT) and The 6th International Conference on Emerging Data and Industry 4.0 (EDI40). [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877050923005926>
- [211] A. d. Garcez, M. Gori, L. C. Lamb, L. Serafini, M. Spranger, and S. N. Tran, “Neural-symbolic computing: An effective methodology for principled integration of machine learning and reasoning,” *arXiv preprint arXiv:1905.06088*, 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1905.06088>
- [212] G. Pisano, G. Ciatto, R. Calegari, A. Omicini *et al.*, “Neuro-symbolic computation for xai: Towards a unified model,” in *CEUR WORKSHOP PROCEEDINGS*, vol. 2706. Sun SITE Central Europe, RWTH Aachen University, 2020, pp. 101–117. [Online]. Available: <https://hdl.handle.net/11585/781387>
- [213] M. S. Munir, K. T. Kim, A. Adhikary, W. Saad, S. Shetty, S.-B. Park, and C. S. Hong, “Neuro-symbolic explainable artificial intelligence twin for zero-touch ioe in wireless network,” *IEEE Internet of Things Journal*, vol. 10, no. 24, pp. 22 451–22 468, 2023. [Online]. Available: <https://doi.org/10.1109/JIOT.2023.3303713>
- [214] B. M. Keneni, D. Kaur, A. Al Bataineh, V. K. Devabhaktuni, A. Y. Javaid, J. D. Zaiantz, and R. P. Marinier, “Evolving rule-based explainable artificial intelligence for unmanned aerial vehicles,” *IEEE Access*, vol. 7, pp. 17 001–17 016, 2019. [Online]. Available: <https://doi.org/10.1109/ACCESS.2019.2893141>
- [215] M. J. Rober, “Contrastive explanation for machine learning,” Master’s thesis, Utrecht University, 2018. [Online]. Available: <https://studenttheses.uu.nl/handle/20.500.12932/30669>

Bibliography

- [216] I. Stepin, J. M. Alonso, A. Catala, and M. Pereira-Fariña, “A survey of contrastive and counterfactual explanation generation methods for explainable artificial intelligence,” *IEEE Access*, vol. 9, pp. 11 974–12 001, 2021. [Online]. Available: <https://doi.org/10.1109/ACCESS.2021.3051315>
- [217] J. Hoffmann and D. Magazzeni, *Explainable AI Planning (XAIP): Overview and the Case of Contrastive Explanation (Extended Abstract)*. Cham: Springer International Publishing, 2019, pp. 277–282. [Online]. Available: https://doi.org/10.1007/978-3-030-31423-1_9
- [218] X. Dai, M. T. Keane, L. Shalloo, E. Ruelle, and R. M. Byrne, “Counterfactual explanations for prediction and diagnosis in xai,” in *Proceedings of the 2022 AAAI/ACM Conference on AI, Ethics, and Society*, ser. AIES ’22. New York, NY, USA: Association for Computing Machinery, 2022, p. 215–226. [Online]. Available: <https://doi.org/10.1145/3514094.3534144>
- [219] S. Nguyen and B. Tran, “Xmap: explainable mapping analytical process,” *Complex & Intelligent Systems*, vol. 8, no. 2, pp. 1187–1204, 2022. [Online]. Available: <https://doi.org/10.1007/s40747-021-00583-8>
- [220] G. Alicioglu and B. Sun, “A survey of visual analytics for explainable artificial intelligence methods,” *Computers & Graphics*, vol. 102, pp. 502–520, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0097849321001886>
- [221] L. van der Maaten and G. Hinton, “Visualizing data using t-sne,” *Journal of Machine Learning Research*, vol. 9, no. 86, pp. 2579–2605, 2008. [Online]. Available: <http://jmlr.org/papers/v9/vandermaaten08a.html>
- [222] A. Shabbir and M. Zubair, “Interpretable deep learning classifier using explainable ai for non-small cell lung cancer,” in *2024 Horizons of Information Technology and Engineering (HITE)*, 2024, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/HITE63532.2024.10777248>

Bibliography

- [223] L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction,” *arXiv preprint arXiv:1802.03426*, 2018. [Online]. Available: <https://doi.org/10.48550/arXiv.1802.03426>
- [224] Y. Sasaki, R. Watanabe, T. Shimizu, Y. Hasegawa, and F. Saitoh, “Visualization of evaluation viewpoints in similar customers by xai based on review evaluation scores,” in *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2023, pp. 1052–1056. [Online]. Available: <https://doi.org/10.1109/IEEM58616.2023.10406493>
- [225] H. T. T. Nguyen, H. Q. Cao, K. V. T. Nguyen, and N. D. K. Pham, “Evaluation of explainable artificial intelligence: Shap, lime, and cam,” in *Proceedings of the FPT AI Conference*, 2021, pp. 1–6. [Online]. Available: https://www.researchgate.net/publication/362165633_Evaluation_of_Explainable_Artificial_Intelligence_SHAP_LIME_and_CAM
- [226] B. Aldughayfiq, F. Ashfaq, N. Z. Jhanjhi, and M. Humayun, “Explainable ai for retinoblastoma diagnosis: Interpreting deep learning models with lime and shap,” *Diagnostics*, vol. 13, no. 11, 2023. [Online]. Available: <https://www.mdpi.com/2075-4418/13/11/1932>
- [227] E. I. Sklar and M. Q. Azhar, “Explanation through argumentation,” in *Proceedings of the 6th International Conference on Human-Agent Interaction*, 2018, pp. 277–285. [Online]. Available: <https://doi.org/10.1145/3284432.3284470>
- [228] K. Čyras, A. Rago, E. Albini, P. Baroni, and F. Toni, “Argumentative xai: A survey,” 2021. [Online]. Available: <https://arxiv.org/abs/2105.11266>
- [229] A. Borg and F. Bex, “A basic framework for explanations in argumentation,” *IEEE Intelligent Systems*, vol. 36, no. 2, pp. 25–35, 2021. [Online]. Available: <https://doi.org/10.1109/MIS.2021.3053102>
- [230] O. Cocarascu, A. Rago, and F. Toni, “Explanation via machine arguing,” in *Reasoning Web International Summer School*. Springer, 2020, pp. 53–84. [Online]. Available: https://doi.org/10.1007/978-3-030-60067-9_3

Bibliography

- [231] L. Caroprese, E. Vocaturo, and E. Zumpano, “Argumentation approaches for explainable ai in medical informatics,” *Intelligent Systems with Applications*, vol. 16, p. 200109, 2022. [Online]. Available: <https://doi.org/10.1016/j.iswa.2022.200109>
- [232] O. Cocarascu, A. Rago, and F. Toni, “Extracting dialogical explanations for review aggregations with argumentative dialogical agents,” in *Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems*. Association for Computing Machinery, 2019, pp. 1261–1269. [Online]. Available: <https://doi.org/10.5555/3306127.3331830>
- [233] D. C. Hernandez-Bocanegra and J. Ziegler, “Explaining recommendations through conversations: Dialog model and the effects of interface type and degree of interactivity,” *ACM Trans. Interact. Intell. Syst.*, vol. 13, no. 2, Apr. 2023. [Online]. Available: <https://doi.org/10.1145/3579541>
- [234] E. Cambria, L. Malandri, F. Mercurio, M. Mezzanzanica, and N. Nobani, “A survey on xai and natural language explanations,” *Information Processing & Management*, vol. 60, no. 1, p. 103111, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0306457322002126>
- [235] S. Luo, H. Ivison, S. C. Han, and J. Poon, “Local interpretations for explainable natural language processing: A survey,” *ACM Comput. Surv.*, vol. 56, no. 9, Apr. 2024. [Online]. Available: <https://doi.org/10.1145/3649450>
- [236] E. Cambria, L. Malandri, F. Mercurio, N. Nobani, and A. Seveso, “Xai meets llms: A survey of the relation between explainable ai and large language models,” *arXiv preprint arXiv:2407.15248*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2407.15248>
- [237] P. Mavrepis, G. Makridis, G. Fatouros, V. Koukos, M. M. Separdani, and D. Kyriazis, “Xai for all: Can large language models simplify explainable ai?” 2024. [Online]. Available: <https://arxiv.org/abs/2401.13110>

Bibliography

- [238] M. Dragoni and I. Donadello, “A knowledge-based strategy for xai: The explanation graph,” *Semantic Web Journal*, 2022. [Online]. Available: <https://semantic-web-journal.net/content/knowledge-based-strategy-xaithe-explanation-graph>
- [239] E. Rajabi and K. Etminani, “Knowledge-graph-based explainable ai: A systematic review,” *Journal of Information Science*, vol. 50, no. 4, pp. 1019–1029, 2024. [Online]. Available: <https://doi.org/10.1177/01655515221112844>
- [240] A. Füßl, V. Nissen, and S. H. Heringklee, “Knowledge graph-based explainable artificial intelligence for business process analysis,” *International Journal of Semantic Computing*, vol. 17, no. 02, pp. 173–197, 2023. [Online]. Available: <https://doi.org/10.1142/S1793351X23600024>
- [241] P. Perdomo-Quinteiro, K. Wolstencroft, M. Roos, and N. Queralt-Rosinach, “Knowledge graphs and explainable ai for drug repurposing on rare diseases,” *bioRxiv*, 2024. [Online]. Available: <https://www.biorxiv.org/content/early/2024/10/17/2024.10.17.618804>
- [242] N. Tiwary, S. A. Mohd Noah, F. Fauzi, and T. S. Yee, “A review of explainable recommender systems utilizing knowledge graphs and reinforcement learning,” *IEEE Access*, vol. 12, pp. 91 999–92 019, 2024. [Online]. Available: <https://doi.org/10.1109/ACCESS.2024.3422416>
- [243] P. Johri, S. K. Khatri, A. T. Al-Taani, M. Sabharwal, S. Suvanov, and A. Kumar, “Natural language processing: History, evolution, application, and future work,” in *Proceedings of 3rd International Conference on Computing Informatics and Networks: ICCIN 2020*. Springer, 2021, pp. 365–375. [Online]. Available: https://doi.org/10.1007/978-981-15-9712-1_31
- [244] S. C. Fanni, M. Febi, G. Aghakhanyan, and E. Neri, “Natural language processing,” in *Introduction to Artificial Intelligence*. Springer, 2023, pp. 87–99. [Online]. Available: https://doi.org/10.1007/978-3-031-25928-9_5

Bibliography

- [245] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here?” *Proceedings of the IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000. [Online]. Available: <https://doi.org/10.1109/5.880083>
- [246] A. Nadas, “Estimation of probabilities in the language model of the ibm speech recognition system,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 32, no. 4, pp. 859–861, 1984. [Online]. Available: <https://doi.org/10.1109/TASSP.1984.1164378>
- [247] H. Wang, J. Li, H. Wu, E. Hovy, and Y. Sun, “Pre-trained language models and their applications,” *Engineering*, vol. 25, pp. 51–65, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2095809922006324>
- [248] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of machine learning research*, vol. 3, no. Feb, pp. 1137–1155, 2003. [Online]. Available: https://www.jmlr.org/papers/v3/bengio03a.html?source=post_page-----&ref=https://githubhelp.com
- [249] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997. [Online]. Available: <https://doi.org/10.1162/neco.1997.9.8.1735>
- [250] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean, “Google’s neural machine translation system: Bridging the gap between human and machine translation,” *CoRR*, vol. abs/1609.08144, 2016. [Online]. Available: <http://arxiv.org/abs/1609.08144>
- [251] M. Yamada, “The impact of google neural machine translation on post-editing by student translators,” *The Journal of Specialised Translation*, vol. 31, no. 1, pp. 87–106, 2019. [Online]. Available: https://jostrans.soap2.ch/issue31/art_yamada.php

Bibliography

- [252] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, “Attention is all you need,” in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf
- [253] A. Radford and K. Narasimhan, “Improving language understanding by generative pre-training,” 2018. [Online]. Available: <https://api.semanticscholar.org/CorpusID:49313245>
- [254] M. U. Hadi, R. Qureshi, A. Shah, M. Irfan, A. Zafar, M. B. Shaikh, N. Akhtar, J. Wu, S. Mirjalili *et al.*, “A survey on large language models: Applications, challenges, limitations, and practical usage,” *Authorea Preprints*, 2023. [Online]. Available: <https://doi.org/10.36227/techrxiv.23589741.v3>
- [255] H. Naveed, A. U. Khan, S. Qiu, M. Saqib, S. Anwar, M. Usman, N. Akhtar, N. Barnes, and A. Mian, “A comprehensive overview of large language models,” *ACM Transactions on Intelligent Systems and Technology*, vol. 16, no. 5, pp. 1–72, 2025.
- [256] T. Hu and X.-H. Zhou, “Unveiling llm evaluation focused on metrics: Challenges and solutions,” *arXiv preprint arXiv:2404.09135*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2404.09135>
- [257] H. Abdel-Nabi, A. Awajan, and M. Z. Ali, “Deep learning-based question answering: a survey,” *Knowledge and Information Systems*, vol. 65, no. 4, pp. 1399–1485, 2023. [Online]. Available: <https://doi.org/10.1007/s10115-022-01783-5>
- [258] , L. Tong, and Q. Ge, “Achieving higher factual accuracy in llama llm with weighted distribution of retrieval-augmented generation,” May 2024. [Online]. Available: osf.io/ctw8v_v1

Bibliography

- [259] M. Capobianco, “Supervised machine generated text detection using llm encoders in various data resource scenarios,” Ph.D. dissertation, WORCESTER POLYTECHNIC INSTITUTE, 2023.
- [260] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” April 2019. [Online]. Available: <http://arxiv.org/abs/1904.09675>
- [261] D. Banerjee, P. Singh, A. Avadhanam, and S. Srivastava, “Benchmarking llm powered chatbots: Methods and metrics,” 2023. [Online]. Available: <https://arxiv.org/abs/2308.04624>
- [262] K. Singhal, S. Azizi, T. Tu, S. S. Mahdavi, J. Wei, H. W. Chung, N. Scales, A. Tanwani, H. Cole-Lewis, S. Pfohl *et al.*, “Large language models encode clinical knowledge,” *arXiv preprint arXiv:2212.13138*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2212.13138>
- [263] H. Song, Y. Xia, Z. Luo, H. Liu, Y. Song, X. Zeng, T. Li, G. Zhong, J. Li, M. Chen *et al.*, “Evaluating the performance of different large language models on health consultation and patient education in urolithiasis,” *Journal of Medical Systems*, vol. 47, no. 1, p. 125, 2023. [Online]. Available: <https://doi.org/10.1007/s10916-023-02021-3>
- [264] J. Harasta, T. Novotná, and J. Savelka, “It cannot be right if it was written by ai: On lawyers’ preferences of documents perceived as authored by an llm vs a human,” *Artificial Intelligence and Law*, 2024. [Online]. Available: <https://doi.org/10.1007/s10506-024-09422-w>
- [265] A. Liusie, V. Raina, Y. Fathullah, and M. Gales, “Efficient llm comparative assessment: a product of experts framework for pairwise comparisons,” *arXiv preprint arXiv:2405.05894*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2405.05894>
- [266] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, Y. Du, C. Yang, Y. Chen, Z. Chen, J. Jiang, R. Ren, Y. Li,

Bibliography

- X. Tang, Z. Liu, P. Liu, J.-Y. Nie, and J.-R. Wen, “A survey of large language models,” 2025. [Online]. Available: <https://arxiv.org/abs/2303.18223>
- [267] J. Kaplan, S. McCandlish, T. Henighan, T. B. Brown, B. Chess, R. Child, S. Gray, A. Radford, J. Wu, and D. Amodei, “Scaling laws for neural language models,” *arXiv preprint arXiv:2001.08361*, 2020. [Online]. Available: <https://doi.org/10.48550/arXiv.2001.08361>
- [268] J. Hoffmann, S. Borgeaud, A. Mensch, E. Buchatskaya, T. Cai, E. Rutherford, D. d. L. Casas, L. A. Hendricks, J. Welbl, A. Clark *et al.*, “Training compute-optimal large language models,” *arXiv preprint arXiv:2203.15556*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2203.15556>
- [269] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler *et al.*, “Emergent abilities of large language models,” *arXiv preprint arXiv:2206.07682*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2206.07682>
- [270] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, “Language models are few-shot learners,” in *Advances in Neural Information Processing Systems*, H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, Eds., vol. 33. Curran Associates, Inc., 2020, pp. 1877–1901. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfc4967418bfb8ac142f64a-Paper.pdf
- [271] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, “Finetuned language models are zero-shot learners,” *arXiv preprint arXiv:2109.01652*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2109.01652>

Bibliography

- [272] J. Wei, X. Wang, D. Schuurmans, M. Bosma, F. Xia, E. Chi, Q. V. Le, D. Zhou *et al.*, “Chain-of-thought prompting elicits reasoning in large language models,” *Advances in neural information processing systems*, vol. 35, pp. 24 824–24 837, 2022. [Online]. Available: <https://proceedings.neurips.cc/paper/2022/hash/9d5609613524ecf4f15af0f7b31abca4-Abstract-Conference.html>
- [273] T. Schick, J. Dwivedi-Yu, R. Dessì, R. Raileanu, M. Lomeli, E. Hambro, L. Zettlemoyer, N. Cancedda, and T. Scialom, “Toolformer: Language models can teach themselves to use tools,” *Advances in Neural Information Processing Systems*, vol. 36, 2024. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2023/hash/d842425e4bf79ba039352da0f658a906-Abstract-Conference.html
- [274] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, “Webgpt: Browser-assisted question-answering with human feedback,” *arXiv preprint arXiv:2112.09332*, 2021. [Online]. Available: <https://doi.org/10.48550/arXiv.2112.09332>
- [275] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, “Training language models to follow instructions with human feedback,” *Advances in neural information processing systems*, vol. 35, pp. 27 730–27 744, 2022. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract-Conference.html
- [276] P. K. Kanithi, C. Christophe, M. A. Pimentel, T. Raha, N. Saadi, H. Javed, S. Maslenskova, N. Hayat, R. Rajan, and S. Khan, “Medic: Towards a comprehensive framework for evaluating llms in clinical applications,” *arXiv preprint arXiv:2409.07314*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2409.07314>
- [277] C. Wu, Z. Lin, W. Fang, and Y. Huang, “A medical diagnostic assistant based on llm,” in *China Health Information Processing Conference*. Springer, 2023,

Bibliography

- pp. 135–147. [Online]. Available: https://doi.org/10.1007/978-981-97-1717-0_12
- [278] M. Grzesiak, K. Kluza, N. R. Potoczek, and L. Szała, “Beyond traditional learning: The llm revolution in bpm education at university,” in *International Conference on Business Process Management*. Springer, 2024, pp. 406–415. [Online]. Available: https://doi.org/10.1007/978-3-031-70445-1_29
- [279] Z. Zhang, D. Zhang-Li, J. Yu, L. Gong, J. Zhou, Z. Liu, L. Hou, and J. Li, “Simulating classroom education with llm-empowered agents,” *arXiv preprint arXiv:2406.19226*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2406.19226>
- [280] E. Johnson and N. Wilson, “Enhancing agricultural machinery management through advanced llm integration,” *arXiv preprint arXiv:2407.20588*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2407.20588>
- [281] A. Tzachor, M. Devare, C. Richards, P. Pypers, A. Ghosh, J. Koo, S. Johal, and B. King, “Large language models and agricultural extension services,” *Nature food*, vol. 4, no. 11, pp. 941–948, 2023. [Online]. Available: <https://doi.org/10.1038/s43016-023-00867-x>
- [282] H. Zhao, Z. Liu, Z. Wu, Y. Li, T. Yang, P. Shu, S. Xu, H. Dai, L. Zhao, G. Mai *et al.*, “Revolutionizing finance with llms: An overview of applications and insights,” *arXiv preprint arXiv:2401.11641*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2401.11641>
- [283] I. de Zarzà, J. de Curtò, G. Roig, and C. T. Calafate, “Optimized financial planning: integrating individual and cooperative budgeting models with llm recommendations,” *AI*, vol. 5, no. 1, pp. 91–114, 2023. [Online]. Available: <https://doi.org/10.3390/ai5010006>
- [284] H. Jin, L. Huang, H. Cai, J. Yan, B. Li, and H. Chen, “From llms to llm-based agents for software engineering: A survey of current, challenges and future,” *arXiv preprint arXiv:2408.02479*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.02479>

Bibliography

- [285] C. Gkournelos, C. Konstantinou, and S. Makris, “An llm-based approach for enabling seamless human-robot collaboration in assembly,” *CIRP Annals*, 2024. [Online]. Available: <https://doi.org/10.1016/j.cirp.2024.04.002>
- [286] S. R. Cox and W. T. Ooi, “Conversational interactions with npcs in llm-driven gaming: Guidelines from a content analysis of player feedback,” in *International Workshop on Chatbot Research and Design*. Springer, 2023, pp. 167–184. [Online]. Available: https://doi.org/10.1007/978-3-031-54975-5_10
- [287] Z. Song, C. Wang, J. Sheng, C. Zhang, G. Yu, J. Fan, and T. Chen, “Moviellm: Enhancing long video understanding with ai-generated movies,” *arXiv preprint arXiv:2403.01422*, 2024. [Online]. Available: https://ui.adsabs.harvard.edu/link_gateway/2024arXiv240301422S/doi:10.48550/arXiv.2403.01422
- [288] I. Cheong, K. Xia, K. K. Feng, Q. Z. Chen, and A. X. Zhang, “(a) i am not a lawyer, but...: Engaging legal experts towards responsible llm policies for legal advice,” in *The 2024 ACM Conference on Fairness, Accountability, and Transparency*, 2024, pp. 2454–2469. [Online]. Available: <https://doi.org/10.1145/3630106.3659048>
- [289] I. Cheong, A. Caliskan, and T. Kohno, “Envisioning legal mitigations for llm-based intentional and unintentional harms,” *Administrative Law Journal*, 2022. [Online]. Available: <https://blog.genlaw.org/CameraReady/32.pdf>
- [290] A. Kasuga and R. Yonetani, “Cxsimulator: A user behavior simulation using llm embeddings for web-marketing campaign assessment,” *arXiv preprint arXiv:2407.21553*, 2024. [Online]. Available: <https://doi.org/10.1145/3627673.3679894>
- [291] J. Brand, A. Israeli, and D. Ngwe, “Using llms for market research,” *Harvard business school marketing unit working paper*, no. 23-062, 2023. [Online]. Available: <https://dx.doi.org/10.2139/ssrn.4395751>
- [292] V. Pallagani, B. C. Muppasani, K. Roy, F. Fabiano, A. Loreggia, K. Murugesan, B. Srivastava, F. Rossi, L. Horesh, and A. Sheth, “On the prospects

Bibliography

- of incorporating large language models (llms) in automated planning and scheduling (aps),” in *Proceedings of the International Conference on Automated Planning and Scheduling*, vol. 34, 2024, pp. 432–444. [Online]. Available: <https://doi.org/10.1609/icaps.v34i1.31503>
- [293] D. Jobson and Y. Li, “Investigating the potential of using large language models for scheduling,” in *Proceedings of the 1st ACM International Conference on AI-Powered Software*, 2024, pp. 170–171. [Online]. Available: <https://doi.org/10.1145/3664646.3665084>
- [294] S. A. Prieto, E. T. Mengiste, and B. García de Soto, “Investigating the use of chatgpt for the scheduling of construction projects,” *Buildings*, vol. 13, no. 4, p. 857, 2023. [Online]. Available: <https://doi.org/10.3390/buildings13040857>
- [295] H. Oh, K. Kim, J. Kim, S. Kim, J. Lee, D.-s. Chang, and J. Seo, “Exegpt: Constraint-aware resource scheduling for llm inference,” in *Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, Volume 2*, 2024, pp. 369–384. [Online]. Available: <https://doi.org/10.1145/3620665.3640383>
- [296] H. Abgaryan, A. Harutyunyan, and T. Cazenave, “Llms can schedule,” 2024. [Online]. Available: <https://arxiv.org/abs/2408.06993>
- [297] X. Wang, G. Wu, L. Xing, and W. Pedrycz, “Agile earth observation satellite scheduling over 20 years: Formulations, methods, and future directions,” *IEEE Systems Journal*, vol. 15, no. 3, pp. 3881–3892, 2020. [Online]. Available: <https://doi.org/10.1109/JSYST.2020.2997050>
- [298] S. Augenstein, A. Estanislao, E. Guere, and S. Blaes, “Optimal scheduling of a constellation of Earth-imaging satellites, for maximal data throughput and efficient human management,” *Proceedings International Conference on Automated Planning and Scheduling, ICAPS*, vol. 2016-January, no. Icaps, pp. 345–352, 2016. [Online]. Available: <https://ojs.aaai.org/index.php/ICAPS/article/view/13784>

Bibliography

- [299] F. Khafa, J. Sun, A. Barolli, A. Biberaj, and L. Barolli, “Genetic algorithms for satellite scheduling problems,” *Mobile Information Systems*, vol. 8, no. 4, pp. 351–377, 2012. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.3233/MIS-2012-00153>
- [300] H. Fan, Z. Yang, S. Wu, X. Zhang, J. Long, and L. Liu, “An efficient satellite resource cooperative scheduling method on spatial information networks,” *Mathematics*, vol. 9, no. 24, pp. 1–23, 2021. [Online]. Available: <https://doi.org/10.3390/math9243293>
- [301] Z. Wang, Z. Zhang, and Y. Chen, “Multi-objective optimization of satellite-ground time synchronization scheduling problem,” in *2019 IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2019, pp. 530–537. [Online]. Available: <https://doi.org/10.1109/CEC.2019.8790100>
- [302] J. C. Pemberton and F. Galiber, “A constraint-based approach to satellite scheduling,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 57, pp. 101–114, 2001. [Online]. Available: <https://dl.acm.org/doi/10.5555/376243.376249>
- [303] X. Wang, G. Song, R. Leus, and C. Han, “Robust earth observation satellite scheduling with uncertainty of cloud coverage,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 3, pp. 2450–2461, 2020. [Online]. Available: <https://doi.org/10.1109/TAES.2019.2947978>
- [304] L. Lan and J. Berkhout, “Pyjobshop: Solving scheduling problems with constraint programming in python,” 2025. [Online]. Available: <https://arxiv.org/abs/2502.13483>
- [305] S. Krigman, T. Grinshpoun, and L. Dery, “Scheduling of earth observing satellites using distributed constraint optimization,” *Journal of Scheduling*, vol. 27, no. 5, pp. 507–524, 2024. [Online]. Available: <https://doi.org/10.1007/s10951-024-00816-x>

Bibliography

- [306] S. Squillaci, S. Roussel, and C. Pralet, “Parallel scheduling of complex requests for a constellation of earth observing satellites,” in *PAIS 2022*. IOS Press, 2022, pp. 100–113. [Online]. Available: <https://doi.org/10.3233/faia220068>
- [307] M. K. Ben-Larbi, K. Flores Pozo, T. Haylok, M. Choi, B. Grzesik, A. Haas, D. Krupke, H. Konstanski, V. Schaus, S. P. Fekete, C. Schurig, and E. Stoll, “Towards the automated operations of large distributed satellite systems. part 1: Review and paradigm shifts,” *Advances in Space Research*, vol. 67, no. 11, pp. 3598–3619, 2021, satellite Constellations and Formation Flying. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0273117720305676>
- [308] F. Affaitati, *Automation in Operations for Earth Bound Satellites – State of the Art and Prospective*. AIAA, 2024. [Online]. Available: <https://arc.aiaa.org/doi/abs/10.2514/6.2024-4806>
- [309] H. A. Tahir, F. Nabi, M. Z. Tariq, A. F. Khan, and A. Mahmud, “Insights into the future: Xai integration in o-ran and space communication systems,” in *2024 Multimedia University Engineering Conference (MECON)*, 2024, pp. 1–6. [Online]. Available: <https://doi.org/10.1109/mecon62796.2024.10776550>
- [310] N. van Omme, L. Perron, and V. Furnon, “or-tools user’s manual,” *Rapp. tech. Google*, 2014. [Online]. Available: https://acrogenesis.com/or-tools/documentation/user_manual/
- [311] D. Phiri, M. Simwanda, S. Salekin, V. R. Nyirenda, Y. Murayama, and M. Ranagalage, “Sentinel-2 data for land cover/use mapping: A review,” *Remote Sensing*, vol. 12, no. 14, 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/14/2291>
- [312] Ansys Government Initiatives (AGI), “Systems Tool Kit (STK): Digital Mission Engineering Software,” Current Year, accessed on: February 24, 2026. [Online]. Available: <https://www.ansys.com/products/missions/ansys-stk>

Bibliography

- [313] E. Rajabi and S. Kafaie, “Knowledge graphs and explainable ai in healthcare,” *Information*, vol. 13, no. 10, 2022. [Online]. Available: <https://www.mdpi.com/2078-2489/13/10/459>
- [314] A. Fensel, Z. Akbar, E. Kärle, C. Blank, P. Pixner, and A. Gruber, “Knowledge graphs for online marketing and sales of touristic services,” *Information*, vol. 11, no. 5, 2020. [Online]. Available: <https://www.mdpi.com/2078-2489/11/5/253>
- [315] H. Ko, P. Witherell, Y. Lu, S. Kim, and D. W. Rosen, “Machine learning and knowledge graph based design rule construction for additive manufacturing,” *Additive Manufacturing*, vol. 37, p. 101620, 2021. [Online]. Available: <https://doi.org/10.1016/j.addma.2020.101620>
- [316] Q. Yan, M. Ding, J. Lu, J. Lu, H. Yang, and F. Xie, “Construction of dynamic knowledge graph for grid day-ahead scheduling optimization decision,” in *2022 IEEE 10th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, vol. 10. IEEE, 2022, pp. 1821–1826. [Online]. Available: <https://doi.org/10.1109/itaic54216.2022.9836955>
- [317] A. Viros Martin, K. Cheng, A. Fang, Z. Zheng, H. Kress-Gazit, A. Mehta, D. Selva, and Y. Sun, “Decentralized context-based on-board planning for earth observation missions,” in *AIAA Scitech 2021 Forum*, 2021, p. 1469. [Online]. Available: <https://doi.org/10.2514/6.2021-1469>
- [318] E. Rajabi and K. Etminani, “Knowledge-graph-based explainable ai: A systematic review,” *Journal of Information Science*, p. 01655515221112844, 2022. [Online]. Available: <https://doi.org/10.1177/01655515221112844>
- [319] F. Lecue, “On the role of knowledge graphs in explainable ai,” *Semantic Web*, vol. 11, no. 1, pp. 41–51, 2020. [Online]. Available: <https://doi.org/10.3233/sw-190374>
- [320] Z. Chen, Y. Wang, B. Zhao, J. Cheng, X. Zhao, and Z. Duan, “Knowledge graph completion: A review,” *IEEE Access*, vol. 8, pp. 192 435–192 456, 2020. [Online]. Available: <https://doi.org/10.1109/ACCESS.2020.3030076>

Bibliography

- [321] A. Berquand, “Text mining and natural language processing for the early stages of space mission design,” Ph.D. dissertation, University of Strathclyde, 2022. [Online]. Available: <http://dx.doi.org/10.48730/95nx-rc75>
- [322] G. M. Valle and M. Cashmore, “Deep reinforcement learning for plan execution,” in *IntEx Workshop on Integrated Planning, Acting, and Execution*, 2022. [Online]. Available: <https://strathprints.strath.ac.uk/id/eprint/81518>
- [323] K. Sundar, J. Qin, S. Rathinam, L. Ntaimo, S. Darbha, and C. Valicka, “Algorithms for a Satellite Constellation scheduling Problem,” *IEEE International Conference on Automation Science and Engineering*, vol. 2016-November, pp. 373–378, 2016. [Online]. Available: <https://doi.org/10.1109/coase.2016.7743431>
- [324] M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi, and L. Vangelista, “A survey on technologies, standards and open challenges in satellite iot,” *IEEE Communications Surveys & Tutorials*, vol. 23, no. 3, pp. 1693–1720, 2021. [Online]. Available: <https://doi.org/10.1109/COMST.2021.3078433>
- [325] G. Stock, J. A. Fraire, H. Hermanns, E. Cruz, A. Isaacs, and Z. Imbrosh, “On the automation, optimization, and in-orbit validation of intelligent satellite constellation operations,” 2022. [Online]. Available: <https://arxiv.org/abs/2210.11171>
- [326] S. Cao, R. Chen, M. Pan, W. Yang, and X. Li, “Beyond manual modeling: Automating gui model generation using design documents,” in *Proceedings of the 39th IEEE/ACM International Conference on Automated Software Engineering*, 2024, pp. 91–103. [Online]. Available: <https://doi.org/10.1145/3691620.3695032>
- [327] K. Seßler, M. Fürstenberg, B. Bühler, and E. Kasneci, “Can ai grade your essays? a comparative analysis of large language models and teacher ratings in multidimensional essay scoring,” in *Proceedings of the 15th International Learning Analytics and Knowledge Conference*, ser. LAK '25. New York, NY, USA: Association for Computing Machinery, 2025, p. 462–472. [Online]. Available: <https://doi.org/10.1145/3706468.3706527>

Bibliography

- [328] K. Blair-Walcott, *Comparative Analysis*. Cham: Springer International Publishing, 2023, pp. 79–84. [Online]. Available: https://doi.org/10.1007/978-3-031-04394-9_13
- [329] S. Gupta and R. S. Singh, “User-defined weight based multi objective task scheduling in cloud using whale optimization algorithm,” *Simulation Modelling Practice and Theory*, vol. 133, p. 102915, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1569190X24000297>
- [330] R. Ghafari, F. H. Kabutarkhani, and N. Mansouri, “Task scheduling algorithms for energy optimization in cloud environment: a comprehensive review,” *Cluster Computing*, vol. 25, no. 2, pp. 1035–1093, 2022. [Online]. Available: <https://doi.org/10.1007/s10586-021-03512-z>
- [331] C. Guo, X. Abuduwayiti, Y. Shang, L. Huang, and C. Cui, “Optimized scheduling of integrated energy systems accounting for hydrogen energy multi-utilization models,” *Sustainability*, vol. 16, no. 3, 2024. [Online]. Available: <https://www.mdpi.com/2071-1050/16/3/1198>
- [332] A. Herrmann and H. Schaub, “A comparative analysis of reinforcement learning algorithms for earth-observing satellite scheduling,” *Frontiers in Space Technologies*, vol. 4, 2023. [Online]. Available: <https://www.frontiersin.org/journals/space-technologies/articles/10.3389/frspt.2023.1263489>
- [333] M. Chromik and A. Butz, “Human-xai interaction: A review and design principles for explanation user interfaces,” in *Human-Computer Interaction – INTERACT 2021*, C. Ardito, R. Lanzilotti, A. Malizia, H. Petrie, A. Piccinno, G. Desolda, and K. Inkpen, Eds. Cham: Springer International Publishing, 2021, pp. 619–640. [Online]. Available: https://doi.org/10.1007/978-3-030-85616-8_36
- [334] A. Vassiliades, N. Bassiliades, and T. Patkos, “Argumentation and explainable artificial intelligence: A survey,” *Knowledge Engineering Review*, vol. 36, no. 2019, 2021. [Online]. Available: <https://doi.org/10.1017/S0269888921000011>

Bibliography

- [335] T. J. Bench-Capon and P. E. Dunne, “Argumentation in artificial intelligence,” *Artificial Intelligence*, vol. 171, no. 10-15, pp. 619–641, 2007. [Online]. Available: <https://doi.org/10.1016/j.artint.2007.05.001>
- [336] P. M. Dung, “On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games,” *Artificial Intelligence*, vol. 77, no. 2, pp. 321–357, 1995. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/000437029400041X>
- [337] O. Arieli, A. Borg, M. Hesse, and C. Straßer, “Explainable logic-based argumentation,” in *Computational Models of Argument*. IOS Press, 2022, pp. 32–43. [Online]. Available: <https://doi.org/10.3233/faia220139>
- [338] P. Besnard, C. Cayrol, and M. C. Lagasque-Schiex, “Logical theories and abstract argumentation: A survey of existing works,” pp. 41–102, 2020. [Online]. Available: <https://doi.org/10.3233/AAC-190476>
- [339] M. O. Moguillansky, *A Study of Argument Acceptability Dynamics Through Core and Remainder Sets*, M. Gyssens and G. Simari, Eds. Cham: Springer International Publishing, 2016. [Online]. Available: https://doi.org/10.1007/978-3-319-30024-5_1
- [340] M. Ulbricht and J. P. Wallner, “Strong Explanations in Abstract Argumentation,” *35th Conference on Artificial Intelligence*, pp. 6496–6504, 2021. [Online]. Available: <https://doi.org/10.1609/aaai.v35i7.16805>
- [341] R. Baumann and G. Brewka, “Expanding argumentation frameworks: Enforcing and monotonicity results,” *Frontiers in Artificial Intelligence and Applications*, vol. 216, pp. 75–86, 2010. [Online]. Available: <https://doi.org/10.3233/978-1-60750-619-5-75>
- [342] Z. G. Saribatur and J. P. Wallner, “Existential Abstraction on Argumentation Frameworks via Clustering,” in *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, 11 2021, pp. 549–559. [Online]. Available: <https://doi.org/10.24963/kr.2021/52>

Bibliography

- [343] P. Baroni, D. Gabbay, M. Giacomin, and L. van der Torre, *Handbook of Formal Argumentation*. London, England: College Publications, 2018. [Online]. Available: <https://philpapers.org/rec/BARHOF>
- [344] A. Rago, O. Cocarascu, C. Bechlivanidis, D. Lagnado, and F. Toni, “Argumentative explanations for interactive recommendations,” *Artificial Intelligence*, vol. 296, p. 103506, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0004370221000576>
- [345] S. Modgil and H. Prakken, “The ASPIC + framework for structured argumentation: A tutorial,” *Argument and Computation*, vol. 5, no. 1, pp. 31–62, 2014. [Online]. Available: <https://doi.org/10.1080/19462166.2013.869766>
- [346] G. Brewka and S. Woltran, “Abstract dialectical frameworks,” *KR’10: Proceedings of the Twelfth International Conference on Principles of Knowledge Representation and Reasoning*, p. 102–111, 2010. [Online]. Available: <https://dl.acm.org/doi/abs/10.5555/3031748.3031765>
- [347] G. Alfano, S. Greco, F. Parisi, and I. Trubitsyna, “Incomplete argumentation frameworks: Properties and complexity,” 2022. [Online]. Available: <https://doi.org/10.1609/aaai.v36i5.20483>
- [348] D. M. Gabbay, “Logical foundations for bipolar and tripolar argumentation networks: preliminary results,” *Journal of Logic and Computation*, vol. 26, no. 1, pp. 247–292, 07 2013. [Online]. Available: <https://doi.org/10.1093/logcom/ext027>
- [349] T. Duchatelle, “Argumentation, Logic and Explainability,” Theses, Université Paul Sabatier - Toulouse III, Dec. 2023. [Online]. Available: <https://theses.hal.science/tel-04597625>
- [350] H. Trenquier, “Analyzing and explaining data-driven artificial intelligence models by argumentation,” Ph.D. dissertation, Université Paul Sabatier-Toulouse III, 2023. [Online]. Available: <https://theses.hal.science/tel-04612661/>

Bibliography

- [351] G. Vilone, “A fully automated global post-hoc method based on abstract argumentation for explainable artificial intelligence and its application on fully connected dense deep neural networks,” Ph.D. dissertation, Technological University Dublin, 2024.
- [352] K. Čyras, D. Letsios, R. Misener, and F. Toni, “Argumentation for explainable scheduling (Full Paper with Proofs),” *arXiv*, 2018. [Online]. Available: <https://doi.org/10.48550/arXiv.1811.05437>
- [353] W. Zhang, W. Hou, C. Li, W. Yang, and M. Gen, “Multidirection update-based multiobjective particle swarm optimization for mixed no-idle flow-shop scheduling problem,” *Complex System Modeling and Simulation*, vol. 1, no. 3, pp. 176–197, 2021. [Online]. Available: <https://doi.org/10.23919/CSMS.2021.0017>
- [354] S. Wang, X. Li, L. Gao, and J. Li, “A multi-disjunctive-graph model-based memetic algorithm for the distributed job shop scheduling problem,” *Advanced Engineering Informatics*, vol. 60, p. 102401, 2024. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1474034624000491>
- [355] Y. Li, X. Li, L. Gao, C. Wang, and Y. Teng, “Distributed permutation flow shop scheduling method based on efficient job allocation strategy,” in *2023 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2023, pp. 1421–1425. [Online]. Available: <https://doi.org/10.1109/IEEM58616.2023.10406534>
- [356] A. Borg and F. Bex, “Enforcing Sets of Formulas in Structured Argumentation,” in *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning*, 11 2021, pp. 130–140. [Online]. Available: <https://doi.org/10.24963/kr.2021/13>
- [357] T. Datta and J. P. Dickerson, “Who’s thinking? a push for human-centered evaluation of llms using the xai playbook,” 2023. [Online]. Available: <https://arxiv.org/abs/2303.06223>

Bibliography

- [358] S. Narteni, V. Orani, E. Ferrari, D. Verda, E. Cambiaso, and M. Mongelli, “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)*, 2022, pp. 167–172. [Online]. Available: <https://ieeexplore.ieee.org/document/9982780>
- [359] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, “Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai,” *Information fusion*, vol. 58, pp. 82–115, 2020. [Online]. Available: <https://doi.org/10.1016/j.inffus.2019.12.012>
- [360] S. Ali, T. Abuhmed, S. El-Sappagh, K. Muhammad, J. M. Alonso-Moral, R. Confalonieri, R. Guidotti, J. Del Ser, N. Díaz-Rodríguez, and F. Herrera, “Explainable artificial intelligence (xai): What we know and what is left to attain trustworthy artificial intelligence,” *Information fusion*, vol. 99, p. 101805, 2023. [Online]. Available: <https://doi.org/10.1016/j.inffus.2023.101805>
- [361] J. Wei, X. Wang, D. Schuurmans, M. Bosma, b. ichter, F. Xia, E. Chi, Q. V. Le, and D. Zhou, “Chain-of-thought prompting elicits reasoning in large language models,” in *Advances in Neural Information Processing Systems*, S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, Eds., vol. 35. Curran Associates, Inc., 2022, pp. 24 824–24 837. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
- [362] Z. Shao, Y. Gong, Y. Shen, M. Huang, N. Duan, and W. Chen, “Synthetic prompting: Generating chain-of-thought demonstrations for large language models,” in *Proceedings of the 40th International Conference on Machine Learning*, ser. Proceedings of Machine Learning Research, A. Krause, E. Brunskill, K. Cho, B. Engelhardt, S. Sabato, and J. Scarlett, Eds., vol. 202. PMLR, 23–29 Jul 2023, pp. 30 706–30 775. [Online]. Available: <https://proceedings.mlr.press/v202/shao23a.html>

Bibliography

- [363] M. Turpin, J. Michael, E. Perez, and S. R. Bowman, “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*, ser. NIPS ’23. Red Hook, NY, USA: Curran Associates Inc., 2023. [Online]. Available: <https://dl.acm.org/doi/10.5555/3666122.3669397>
- [364] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, “On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation,” *PLoS one*, vol. 10, no. 7, p. e0130140, 2015. [Online]. Available: <https://doi.org/10.1371/journal.pone.0130140>
- [365] C. Singh, A. Askari, R. Caruana, and J. Gao, “Augmenting interpretable models with large language models during training,” *Nature Communications*, 2023. [Online]. Available: <https://doi.org/10.1038/s41467-023-43713-1>
- [366] F. Friedrich, P. Schramowski, C. Tauchmann, and K. Kersting, “Interactively providing explanations for transformer language models,” in *HHAI*, 2021. [Online]. Available: <https://doi.org/10.3233/faia220218>
- [367] S. Jain and B. C. Wallace, “Attention is not explanation,” in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 3543–3556. [Online]. Available: <https://aclanthology.org/N19-1357>
- [368] S. Wiegrefe and Y. Pinter, “Attention is not not explanation,” in *Conference on Empirical Methods in Natural Language Processing*, 2019. [Online]. Available: <https://doi.org/10.48550/arXiv.1908.04626>
- [369] D. Kim, Y. Song, S. Kim, S. Lee, Y. Wu, J. Shin, and D. Lee, “How should the results of artificial intelligence be explained to users? - research on consumer preferences in user-centered explainable artificial intelligence,” *Technological*

Bibliography

- Forecasting and Social Change*, vol. 188, p. 122343, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0040162523000288>
- [370] M. Foutter, P. Bhoj, R. Sinha, A. Elhafsi, S. Banerjee, C. Agia, J. Kruger, T. Guffanti, D. Gammelli, S. D’Amico *et al.*, “Adapting a foundation model for space-based tasks,” *arXiv preprint arXiv:2408.05924*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2408.05924>
- [371] V. Rodriguez-Fernandez, A. Carrasco, J. Cheng, E. Scharf, P. M. Siew, and R. Linares, “Language models are spacecraft operators,” *arXiv preprint arXiv:2404.00413*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2404.00413>
- [372] P. Darm, A. V. Miceli-Barone, S. B. Cohen, and A. Riccardi, “Knowledge base question answering for space debris queries,” *arXiv preprint arXiv:2305.19734*, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2305.19734>
- [373] M. Doctor, M. López-Caniego, F. Marinic, R. Bhatawdekar, and J. Reerink, “Enhancing object-type searches in esa astronomy science archives extending esasky ai capabilities with llm and retrieval augmented generation,” in *Proceedings of SPAICE2024: The First Joint European Space Agency/IAA Conference on AI in and for Space*, 2024, pp. 134–139. [Online]. Available: <https://doi.org/10.5281/zenodo.13885536>
- [374] E. Waisberg, J. Ong, M. Masalkhi, N. Zaman, S. A. Kamran, P. Sarker, A. G. Lee, and A. Tavakkoli, “Generative pre-trained transformers (gpt) and space health: A potential frontier in astronaut health during exploration missions,” *Prehospital and Disaster Medicine*, pp. 1–5, 2023. [Online]. Available: <https://doi.org/10.1017/s1049023x23005848>
- [375] E. Lagona, S. Hilton, A. Afful, A. Gardi, and R. Sabatini, “Autonomous trajectory optimisation for intelligent satellite systems and space traffic management,” *Acta Astronautica*, vol. 194, pp. 185–201, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576522000364>

Bibliography

- [376] J. A. Ruiz-De-Azua, V. Ramírez, H. Park, A. C. AUGé, and A. Camps, “Assessment of satellite contacts using predictive algorithms for autonomous satellite networks,” *IEEE Access*, vol. 8, pp. 100 732–100 748, 2020. [Online]. Available: <https://doi.org/10.1109/access.2020.2998049>
- [377] X. Wang, J. Wu, Z. Shi, F. Zhao, and Z. Jin, “Deep reinforcement learning-based autonomous mission planning method for high and low orbit multiple agile earth observing satellites,” *Advances in Space Research*, vol. 70, no. 11, pp. 3478–3493, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0273117722007335>
- [378] S. Nag, D. D. Murakami, N. A. Marker, M. T. Lifson, and P. H. Kopardekar, “Prototyping operational autonomy for space traffic management,” *Acta Astronautica*, vol. 180, pp. 489–506, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0094576520307323>
- [379] S. Nag, A. S. Li, V. Ravindra, M. S. Net, K.-M. Cheung, R. Lammers, and B. Bledsoe, “Autonomous scheduling of agile spacecraft constellations with delay tolerant networking for reactive imaging,” 2020. [Online]. Available: <https://arxiv.org/abs/2010.09940>
- [380] K. Čyras, A. Rago, E. Albini, P. Baroni, and F. Toni, “Argumentative xai: a survey,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 4392–4399, survey Track. [Online]. Available: <https://doi.org/10.24963/ijcai.2021/600>
- [381] S. Pandzić, “A logic of defeasible argumentation: Constructing arguments in justification logic,” *Argument and Computation*, vol. 13, pp. 3–47, 2022. [Online]. Available: <https://doi.org/10.3233/aac-200536>
- [382] E. Balkir, S. Kiritchenko, I. Nejadgholi, and K. Fraser, “Challenges in applying explainability methods to improve the fairness of NLP models,” in *Proceedings of the 2nd Workshop on Trustworthy Natural Language Processing (TrustNLP 2022)*,

Bibliography

- A. Verma, Y. Pruksachatkun, K.-W. Chang, A. Galstyan, J. Dhamala, and Y. T. Cao, Eds. Seattle, U.S.A.: Association for Computational Linguistics, Jul. 2022, pp. 80–92. [Online]. Available: <https://aclanthology.org/2022.trustnlp-1.8/>
- [383] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation.” Association for Computational Linguistics, July 2002, pp. 311–318. [Online]. Available: <https://doi.org/10.3115/1073083.1073135>
- [384] C.-Y. Lin, “Rouge: A package for automatic evaluation of summaries.” Association for Computational Linguistics, July 2004, pp. 74–81. [Online]. Available: <https://aclanthology.org/W04-1013/>
- [385] N. Reimers and I. Gurevych, “Sentence-bert: Sentence embeddings using siamese bert-networks,” August 2019.
- [386] C. van der Lee, A. Gatt, E. van Miltenburg, and E. Krahmer, “Human evaluation of automatically generated text: Current trends and best practice guidelines,” *Computer Speech and Language*, vol. 67, 5 2021.
- [387] B. Goodrich, V. Rao, P. J. Liu, and M. Saleh, “Assessing the factual accuracy of generated text.” Association for Computing Machinery, July 2019, pp. 166–175.
- [388] G. Liu, Y. Zhang, Y. Li, and Q. Yao, “Dual reasoning: A gnn-llm collaborative framework for knowledge graph question answering,” 2025. [Online]. Available: <https://arxiv.org/abs/2406.01145>
- [389] D. Shu, T. Chen, M. Jin, C. Zhang, M. Du, and Y. Zhang, “Knowledge graph large language model (kg-llm) for link prediction,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.07311>
- [390] H. Abu-Rasheed, C. Weber, and M. Fathi, “Knowledge graphs as context sources for llm-based explanations of learning recommendations,” 2024. [Online]. Available: <https://arxiv.org/abs/2403.03008>

Bibliography

- [391] Y. Li, R. Zhang, and J. Liu, “An enhanced prompt-based llm reasoning scheme via knowledge graph-integrated collaboration,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.04978>
- [392] J. Wang, K. Sun, L. Luo, W. Wei, Y. Hu, A. W.-C. Liew, S. Pan, and B. Yin, “Large language models-guided dynamic adaptation for temporal knowledge graph reasoning,” 2024. [Online]. Available: <https://arxiv.org/abs/2405.14170>
- [393] C. Cao, Z. Du, and X. Li, “Deep reasoning of large models based on knowledge graph,” in *Sixteenth International Conference on Graphics and Image Processing (ICGIP 2024)*, vol. 13539. SPIE, 2025, pp. 775–782.
- [394] M. Danilevsky, K. Qian, R. Aharonov, Y. Katsis, B. Kawas, and P. Sen, “A survey of the state of explainable AI for natural language processing,” *CoRR*, vol. abs/2010.00711, 2020. [Online]. Available: <https://arxiv.org/abs/2010.00711>
- [395] V. Lai, C. Chen, Q. V. Liao, A. Smith-Renner, and C. Tan, “Towards a science of human-ai decision making: A survey of empirical studies,” *CoRR*, vol. abs/2112.11471, 2021. [Online]. Available: <https://arxiv.org/abs/2112.11471>
- [396] O. Biran and K. McKeown, “Human-centric justification of machine learning predictions,” in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 1461–1467. [Online]. Available: <https://doi.org/10.24963/ijcai.2017/202>
- [397] A. Jarabo Peñas, “Digital twin knowledge graphs for iot platforms: Towards a virtual model for real-time knowledge representation in iot platforms,” 2023. [Online]. Available: <https://urn.kb.se/resolve?urn=urn%3Anbn%3Ase%3Aakth%3Adiva-329198>
- [398] A. M. Hasani, S. Singh, A. Zahergivar, B. Ryan, D. Nethala, G. Bravomontenegro, N. Mendhiratta, M. Ball, F. Farhadi, and A. Malayeri, “Evaluating the performance of generative pre-trained transformer-4 (gpt-4) in standardizing radiology reports,” *European Radiology*, pp. 1–9, 2023. [Online]. Available: <https://doi.org/10.1007/s00330-023-10384-x>

Bibliography

- [399] C. Feng, X. Zhang, and Z. Fei, “Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs,” 2023. [Online]. Available: <https://arxiv.org/abs/2309.03118>
- [400] Y. Wu, N. Hu, S. Bi, G. Qi, J. Ren, A. Xie, and W. Song, “Retrieve-rewrite-answer: A kg-to-text enhanced llms framework for knowledge graph question answering,” 2023. [Online]. Available: <https://arxiv.org/abs/2309.11206>
- [401] X. Wu, H. Zhao, Y. Zhu, Y. Shi, F. Yang, T. Liu, X. Zhai, W. Yao, J. Li, M. Du *et al.*, “Usable xai: 10 strategies towards exploiting explainability in the llm era,” *arXiv preprint arXiv:2403.08946*, 2024. [Online]. Available: <https://doi.org/10.48550/arXiv.2403.08946>
- [402] E. Ben Abdallah, R. Grati, and K. Boukadi, “Towards an explainable irrigation scheduling approach by predicting soil moisture and evapotranspiration via multi-target regression,” *Journal of Ambient Intelligence and Smart Environments*, pp. 1–22, 2023. [Online]. Available: <https://doi.org/10.3233/AIS-220477>
- [403] M. Gashi, B. Mutlu, and S. Thalmann, “Impact of interdependencies: Multi-component system perspective toward predictive maintenance based on machine learning and xai,” *Applied Sciences*, vol. 13, no. 5, p. 3088, 2023. [Online]. Available: <https://doi.org/10.3390/app13053088>
- [404] B. Mullins, “The shape of explanations: A topological account of rule-based explanations in machine learning,” *arXiv e-prints*, pp. arXiv–2301, 2023. [Online]. Available: <https://doi.org/10.48550/arXiv.2301.09042>
- [405] T. Chakraborti, S. Sreedharan, and S. Kambhampati, “The emerging landscape of explainable automated planning & decision making,” in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, C. Bessiere, Ed. International Joint Conferences on Artificial Intelligence Organization, 7 2020, pp. 4803–4811, survey track. [Online]. Available: <https://doi.org/10.24963/ijcai.2020/669>

Bibliography

- [406] P. Gajane, A. Saxena, M. Tavakol, G. Fletcher, and M. Pechenizkiy, “Survey on fair reinforcement learning: Theory and practice,” *arXiv preprint arXiv:2205.10032*, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2205.10032>
- [407] E. Puiutta and E. M. Veith, “Explainable reinforcement learning: A survey,” in *International cross-domain conference for machine learning and knowledge extraction*. Springer, 2020, pp. 77–95. [Online]. Available: https://doi.org/10.1007/978-3-030-57321-8_5
- [408] M. T. Keane, E. M. Kenny, E. Delaney, and B. Smyth, “If only we had better counterfactual explanations: Five key deficits to rectify in the evaluation of counterfactual xai techniques,” in *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*, Z.-H. Zhou, Ed. International Joint Conferences on Artificial Intelligence Organization, 8 2021, pp. 4466–4474, survey Track. [Online]. Available: <https://doi.org/10.24963/ijcai.2021/609>
- [409] T.-C. T. Chen, *Explainable Artificial Intelligence (XAI) in Manufacturing*. Cham: Springer International Publishing, 2023, pp. 1–11. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-031-27961-4_1
- [410] A. Francis, “Graphical modelling classification for construction project scheduling,” *Procedia Engineering*, vol. 123, pp. 162–168, 2015, selected papers from Creative Construction Conference 2015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1877705815031744>
- [411] Y. Sui, M. Zhou, M. Zhou, S. Han, and D. Zhang, “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*, ser. WSDM ’24. New York, NY, USA: Association for Computing Machinery, 2024, p. 645–654. [Online]. Available: <https://dl.acm.org/doi/10.1145/3616855.3635752>

Bibliography

- [412] C. Powell, A. Berquand, and A. Riccardi, “Natural language processing for explainable satellite scheduling,” in *SPACEOPS 2023*, ARE, April 2023, p. #349. [Online]. Available: <https://strathprints.strath.ac.uk/85129/>
- [413] A. S. Jain and S. Meeran, “Deterministic job-shop scheduling: Past, present and future,” *European journal of operational research*, vol. 113, no. 2, pp. 390–434, 1999. [Online]. Available: [https://doi.org/10.1016/S0377-2217\(98\)00113-1](https://doi.org/10.1016/S0377-2217(98)00113-1)
- [414] O. V. Shylo and H. Shams, “Boosting binary optimization via binary classification: A case study of job shop scheduling,” 2018. [Online]. Available: <https://arxiv.org/abs/1808.10813>,<https://optimizer.com/TA.php>
- [415] P. Brucker and S. Knust, *Complex Scheduling*. Springer, 01 2006. [Online]. Available: <https://doi.org/10.1007/3-540-29546-1>
- [416] W. Brinkkötter and P. Brucker, “Solving open benchmark instances for the job-shop problem by parallel head–tail adjustments,” *Journal of Scheduling*, vol. 4, no. 1, pp. 53–64, 2001. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/1099-1425%28200101/02%294%3A1%3C53%3A%3AAID-JOS59%3E3.0.CO%3B2-Y>
- [417] A. Henning Dr. rer. nat., “Practical job shop scheduling issues,” Ph.D. dissertation, Friedrich Schiller University, Jena, Aug 2002, dissertation, Friedrich Schiller Universityät Jena, 2003. [Online]. Available: <https://nbn-resolving.org/urn:nbn:de:gbv:27-20060809-115700-4>
- [418] M. A. Cruz-Chávez, J. d. C. Peralta-Abarca, and M. H. Cruz-Rosales, “Cooperative threads with effective-address in simulated annealing algorithm to job shop scheduling problems,” *Applied Sciences*, vol. 9, no. 16, p. 3360, 2019. [Online]. Available: <https://doi.org/10.3390/app9163360>
- [419] D. Sha and C.-Y. Hsu, “A hybrid particle swarm optimization for job shop scheduling problem,” *Computers & Industrial Engineering*, vol. 51, no. 4, pp. 791–808, 2006. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0360835206001471>

Bibliography

- [420] E. Yuan, S. Cheng, L. Wang, S. Song, and F. Wu, “Solving job shop scheduling problems via deep reinforcement learning,” *Applied Soft Computing*, vol. 143, p. 110436, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1568494623004544>
- [421] S. Matsui and S. Yamada, “An empirical performance evaluation of a parameter-free genetic algorithm for job-shop scheduling problem,” in *2007 IEEE Congress on Evolutionary Computation*. IEEE, 2007, pp. 3796–3803. [Online]. Available: <https://doi.org/10.1109/CEC.2007.4424965>
- [422] E. Taillard, “Best lower and upper bounds known, from or-lib.” 1997. [Online]. Available: http://mistic.heig-vd.ch/taillard/problemes.dir/ordonnancement.dir/jobshop.dir/best_lb_up.txt
- [423] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, “React: Synergizing reasoning and acting in language models,” *arXiv preprint arXiv:2210.03629*, 2022. [Online]. Available: <https://par.nsf.gov/biblio/10451467>
- [424] H. Face, “all-MiniLM-L6-v2 sentence transformer,” 2024. [Online]. Available: <https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2>
- [425] H. Face., “Metric: bert_score,” 2024. [Online]. Available: <https://huggingface.co/spaces/evaluate-metric/bertscore>
- [426] M. Zaki, Jayadeva, Mausam, and N. M. A. Krishnan, “Mascqa: investigating materials science knowledge of large language models,” *Digital Discovery*, vol. 3, pp. 313–327, 2024. [Online]. Available: <http://dx.doi.org/10.1039/D3DD00188A>
- [427] F. Yang, P. Zhao, Z. Wang, L. Wang, B. Qiao, J. Zhang, M. Garg, Q. Lin, S. Rajmohan, and D. Zhang, “Empower large language model to perform better on industrial domain-specific question answering,” in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, M. Wang and I. Zitouni, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 294–312. [Online]. Available: <https://aclanthology.org/2023.emnlp-industry.29/>